

**Міністерство освіти і науки України  
Національний університет водного господарства та  
природокористування**



Національний університет  
водного господарства  
та природокористування

*Власюк Анатолій Павлович  
Мартинюк Петро Миколайович  
Прищеп Оксана Володимирівна  
Філатова Інна Анатоліївна  
Філатов Михайло Сергійович  
Роценюк Алла Михайлівна  
Демчук Олена Станіславівна  
Демчук Микола Борисович  
Мічута Ольга Романівна  
Цветкова Тетяна Павлівна  
Федорчук Наталія Анатоліївна*



**ЛАБОРАТОРНИЙ  
ПРАКТИКУМ  
З ПРОГРАМУВАННЯ**

**Рівне-2011**

УДК 681.3.06(075)

ББК 22.18

В 58

*Затверджено вченою радою Національного університету водного господарства та природокористування.  
(Протокол № 11 від 30 листопада 2009)*

**Рецензенти:**

**Савула Я.Г.**, доктор фізико-математичних наук, професор, декан факультету прикладної математики та інформатики, завідувач кафедри прикладної математики Львівського національного університету ім. І.Франка;

**Марценюк В.П.**, доктор технічних наук, професор, завідувач кафедри медичної інформатики з біофізикою Тернопільського державного медичного університету імені І.Я. Горбачевського;

**Шахрайчук М.Й.**, кандидат фізико-математичних наук, доцент, декан факультету математики та інформатики Рівненського державного гуманітарного університету.

Власюк А.П., Мартинюк П.М., Прищеп О.В., Філатова І.А., Філатов М.С., Рощенюк А.М., Демчук О.С., Демчук М.Б., Мічута О.Р., Цветкова Т.П., Федорчук Н.А.

**В 58** Лабораторний практикум з програмування. Навч. посібник / За загальною редакцією проф. Власюка А.П. – Рівне: НУВГП, 2011. – 495 с.

У посібнику розглянуто типові способи розробки алгоритмів та їх програмування на мовах Pascal, Object Pascal, C, C++, C#. Лабораторний практикум також містить елементи програмування інтелектувальних систем. Завдання до лабораторних робіт супроводжуються достатньою кількістю демонстраційних програм, що суттєво полегшує їх виконання.

Для студентів вузів, коледжів, учнів ліцеїв та шкіл.


**УДК 681.3.06(075)**

**ББК 22.18**

© Власюк А.П., Мартинюк П.М., Прищеп О.В., Філатова І.А.,  
Філатов М.С., Рощенюк А.М., Демчук О.С., Демчук М.Б.,  
Мічута О.Р., Цветкова Т.П., Федорчук Н.А.

© НУВГП, 2011

## Зміст

Вступ .....	8
Література.....	9
Основні скорочення .....	13
Глосарій.....	14
 РОЗДІЛ I. Основи алгоритмізації.....	20
РОЗДІЛ II. Програмування на мові Pascal.....	25
<b>Лабораторна робота №1.</b> .....	25
Математичні та логічні вирази. Оператори присвоєння. Стандартні математичні функції. ....	25
<b>Лабораторна робота №2.</b> .....	32
Лінійні алгоритми та програми.....	32
<b>Лабораторна робота №3.</b> .....	38
Алгоритми та програми розгалуженої структури.....	38
<b>Лабораторна робота № 4.</b> .....	56
Циклічні алгоритми та програми.....	56
<b>Лабораторна робота №5.</b> .....	70
Типові прийоми програмування.....	70
<b>Лабораторна робота №6.</b> .....	79
Програмування ітераційних обчислювальних процесів.....	79
<b>Лабораторна робота №7.</b> .....	89
Одновимірні масиви.....	89
<b>Лабораторна робота №8.</b> .....	100
Багатовимірні масиви .....	100
<b>Лабораторна робота №9.</b> .....	117
Програми з використанням підпрограм.....	117
<b>Лабораторна робота №10.</b> .....	128
Робота з текстовими змінними.....	128
<b>Лабораторна робота №11.</b> .....	135
Використання записів .....	135
<b>Лабораторна робота №12.</b> .....	142
Файли даних. Робота з текстовими та типізованими файлами... ..	142

<b>Лабораторна робота №13.</b>	168
Множини	168
<b>Лабораторна робота №15.</b>	191
Використання модулів	191
<b>Лабораторна робота №16.</b>	197
Використання вказівників та динамічних структур даних	197
<b>16.1.</b> Робота з динамічними масивами	198
<b>16.2.</b> Робота зі списками, стеками, чергами	203
<b>Лабораторна робота №17.</b>	216
Використання об'єктно-орієнтованого програмування (ООП)	216
Повідомлення компілятора про помилки	227



Національний університет  
водного господарства  
та природокористування

### **РОЗДІЛ III. Програмування на мовах C, C++, C#.....239**

<b>Лабораторна робота №1.</b>	239
Вирази. Оператори присвоєння. Стандартні функції	239
<b>Лабораторна робота №2.</b>	241
Лінійні алгоритми та програми	241
<b>Лабораторна робота №3.</b>	244
Алгоритми та програми розгалуженої структури	244
<b>Лабораторна робота № 4.</b>	250
Циклічні алгоритми та програми	250
<b>Лабораторна робота №5.</b>	256
Типові прийоми програмування	256
<b>Лабораторна робота №6.</b>	260
Програмування ітераційних циклічних обчислювальних процесів	260
<b>Лабораторна робота №7.</b>	264
Одновимірні масиви	264
<b>Лабораторна робота №8.</b>	268
Багатовимірні масиви	268
<b>Лабораторна робота №9.</b>	273
Програми з використанням підпрограм	273
<b>Лабораторна робота №10.</b>	276
Робота з текстовими змінними	276

<b>Лабораторна робота №11.</b> .....	279
Використання вказівників та динамічних структур даних .....	279

## **РОЗДІЛ IV. Програмування на мовах C++, C# з використанням ООП** .....283



<b>Лабораторна робота №1.</b> .....	283
Використання класів в C++ .....	283
<b>Лабораторна робота №2.</b> .....	286
Наслідування в C++ .....	286
<b>Лабораторна робота №3.</b> .....	297
Масиви в C# .....	297
<b>Лабораторна робота №4.</b> .....	302
Перевантаження функцій .....	302
<b>Лабораторна робота №5.</b> .....	305
Використання зсилок в C-подібних мовах програмування.....	305
<b>Лабораторна робота №6.</b> .....	312
Створення найпростіших Windows- додатків в середовищі. Visual Studio .NET.....	312
<b>Лабораторна робота №7.</b> .....	315
Робота з компонентами Button, EditBox, Label, CheckButton, RadioButton, GroupBox.....	315
<b>Лабораторна робота №8.</b> .....	320
Циклічні алгоритми та програми .....	320
<b>Лабораторна робота №9.</b> .....	323
Робота з одновимірними масивами. Використання меню при створенні проектів.....	323
<b>Лабораторна робота №10.</b> .....	325
Робота з багатовимірними масивами. Створення багатовіконних додатків .....	325

## **РОЗДІЛ V. Візуально-подійне програмування в середовищі Delphi...329**

Загальні вимоги до виконання лабораторних робіт .....	329
<b>Лабораторна робота №1.</b> .....	330
Ознайомлення з середовищем візуального об'єктно-орієнтованого	



Національний університет  
водного господарства  
та природокористування

програмування Delphi. Створення найпростіших програм.....	330
<b>Лабораторна робота №2.</b> .....	341
Розробка лінійних програм. ....	341
<b>Лабораторна робота №3.</b> .....	359
Розробка циклічних програм.....	359
<b>Лабораторна робота №4.</b> .....	373
Розробка програм з використанням одновимірних масивів. Використання меню при створенні проектів.....	373
<b>Лабораторна робота №5.</b> .....	392
Розробка програм з обробки багатовимірних масивів. Створення багатовіконних додатків. ....	392
<b>Лабораторна робота №6.</b> .....	407
Розробка додатків з використанням графічних можливостей Delphi. ....	407
<b>Лабораторна робота №7.</b> .....	425
Робота з файлами в Delphi.....	425
<b>Лабораторна робота №8.</b> .....	437
Створення баз даних засобами Delphi.....	437
Контрольні завдання по курсу .....	444
<b>Курсова робота</b> .....	
Тематика курсових робіт.....	446
Вимоги до курсової роботи.....	448
Приклад типової курсової роботи.....	450

## **РОЗДІЛ VI. Елементи програмування інтелектуальних систем.....473**

Загальні вимоги до виконання лабораторних робіт.....	473
<b>Лабораторна робота №1.</b> .....	473
Розробка найпростішої інтелектуальної системи для здійснення діалогу... ..	473
<b>Лабораторна робота №2.</b> .....	485
Розробка інтелектуальної системи для можливостями діалогу з використанням асоціативних мереж. ....	485
<b>Лабораторна робота №3.</b> .....	487
Створення автоматизованої системи для розв'язку задач. ....	487

<b>Лабораторна робота №4.</b> .....	<b>491</b>
Розробка інтелектуальних систем для розв'язку задач з елементарної геометрії на основі загальновідомих правил та відомих фактів. ....	<b>491</b>



Національний університет  
водного господарства  
та природокористування



Національний університет  
водного господарства  
та природокористування

## Вступ

Даний посібник призначено для проведення лабораторних занять для вивчення програмування в середовищах *Turbo (Borland) Pascal*, *Delphi*, *C*, *C++*, *C#*, а також для програмування інтелектуальних систем.

Лабораторний практикум містить: необхідний теоретичний матеріал з основ алгоритмізації; практикуми розв'язування задач на мовах *Pascal*, *Object Pascal*, *C*, *C++*, *C#*; короткі відомості про середовище візуально-подійного програмування *Delphi*, особливості програмування в *Object Pascal*, методику створення найпростіших *Delphi*-додатків; приклади програм з детальними коментарями до них та завдання для самостійної роботи. Лабораторний практикум також містить елементи програмування інтелектуальних систем.

Позитивною стороною посібника є детальний опис алгоритмів сортування масивів з детальною оцінкою їх складності та ефективності, тому його можна розглядати, як посібник по алгоритмізації та програмуванню. Матеріал в посібнику подається в доступній для студентів формі, що дозволяє ефективно засвоювати програмування в описаних вище середовищах програмування.

Лабораторний практикум включає лабораторні роботи з найважливіших тем. Кожна лабораторна робота супроводжується необхідним для її виконання теоретичним матеріалом та типовими прикладами з відповідної теми та переліком завдань і контрольних запитань. Крім того, в посібнику подано виконання типової курсової роботи з програмування, примірну тематику курсових робіт та вимоги до її виконання.

Посібник побудовано на матеріалах лекцій та лабораторних занять, які проводяться на кафедрі прикладної математики Національного університету водного господарства та природокористування (м. Рівне).

Автори вдячні магістру прикладної математики Бодову О.С. за художнє оформлення обкладинки.



## Література

1. Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык Паскаль: Учебное пособие. – М.: Наука, 1988. – 320с.
2. Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Семон М.И. Задачи по программированию. – М.: Наука, 1998.
3. Абрамов С.А., Зима Е.В. Начала программирования на языке Паскаль. – М.: Наука, 1987. – 112с.
4. Архангельський А.Я. Программирование в Delphi 7. – М.: ООО „Бином - Пресс”, 2005. – 1152 с.
5. Ахо А.В., Хопкрофт Д.Э., Джеффри Д.У. Структуры данных и алгоритмы: Уч. Пос. – М.:Издательский дом «Вильямс», 2000 – 384с.
6. Бартків А. Б. Гринчишин Я. Т. Ломакович А. М. Turbo Pascal: алгоритми та програми. – К.: Вища школа, 1992.
7. Бобровский С. Delphi 5: учебный курс. – СПб: Питер, 2001.– 540 с.
8. Бородич Ю.С. Разработка программных систем на языке Паскаль: Справочное пособие. – Мн.:Высшая школа, 1992. – 143с.
9. Бородич Ю.С., Вальвачев А.Н., Кузьмич А.М. Паскаль для персональных компьютеров. – Мн.: Высшая школа, 1991. – 365с.
10. Буч Г. ООП с примерами применения. – Киев: Диалектика, 1992.
11. Вальвачев А.Н. Графическое программирование на языке Паскаль: Справ. пособие.- М.: Высшая школа, 1992. – 143 с.
12. Вальвачев А.Н., Крисевич В.С. Программирование на Языке Паскаль для персональных ЭВМ. М.: Высшая школа, 1989.
13. Вирт Н. Алгоритмы и структуры данных. – М.: Мир, 1989.
14. Власюк А.П. Практикум з програмування в середовищі Turbo Pascal. – Рівне: НУВГП, 2005. – Ч.1. – 179 с.
15. Власюк А.П., Рощенко А.М. Збірник задач до виконання лабораторних робіт з дисципліни “Програмування”. – Рівне. – 2001. – Ч.1,2.
16. Власюк А.П., Рощенко А.М., Демчук О.С. Збірник задач для виконання лабораторних робіт з дисципліни “Програмування”. – Рівне: НУВГП, 2003. – Ч.3. – 20 с.
17. Галисеев Г.В. Программирование на языке C#: Самоучитель. – Москва: Вильямс, 2006. – 368 с.
18. Глушаков С.В., Клевцов А.Л., Геребилов С.А. Программирование на Delphi 5.0. – Харьков: Фолио, 2002. – 518 с.
19. Давыдов В.Г. Технологии программирования C++. – СПб.: БХВ, 2005. – 672 с.
20. Джонс Ж., Харроу К. Решение задач в системе Турбо Паскаль – М.: Финансы и статистика, 1991.

21. Дідик І.А. Збірник задач з програмування з прикладами їх розв'язування. – Рівне, 1998. – 79 с.
22. Довгаль С.И., Сбитнев А.И. Персональные ЭВМ: Турбо Паскаль V7.0, Объектное программирование. – Киев: Информ системи сервис, 1993. – 320с.
23. Довгаль С.Н., Литвинов Б.Ю., Сбитнев А.Н. Прикладное программирование в системе Турбо Паскаль. – К.: Довгань, 1996.
24. Дудзяний І.М. Програмування мовою Object Pascal. Навчальний посібник. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2004. – 328 с.
25. Емелина Е.И. Основы программирования на языке Паскаль. – М.: Финансы и статистика. – 1997. – 208с.
26. Епанешников А., Епанешников В. Программирование в среде Turbo Pascal 7.0. – М.: Диалог. МИФИ, 1993.
27. Жуков А. Изучаем Delphi. СПб: Питер, 2000. – 352 с.
28. Зубов В. С. Программирование на Языке TURBO PASCAL.– М.: Филинь, 1997.
29. Ивашкова Э.А. С++. Начала программирования. - Москва: БИНОМ, 2001. – 352 с.
30. Ишкова Э.А. С++. начала программирования. – Москва: БИНОМ, 2004. – 368 с.
31. Йесен К., Вирт Н. Паскаль Руководство для пользователя. – М.: Финансы и статистика, 1989.
32. Климов Ю.С., Касаткин А.И., Мороз С.М. Программирование в среде Turbo Pascal 6.0. – Мн.: Высшая школа, 1992. – 158с.
33. Ковалюк Т.В. Основи програмування. – К.: Видавнична група ВНУ, 2005. – 384 с.: іл..
34. Кораблёв В. С и С++. – СПб.: БХВ, 2002. – 432 с.
35. Крэйг Арнуш. Borland С++: Освой самостоятельно. – Москва: Восточная Книжная Компания, 1997. - 720 с.
36. Культин Н. Turbo Pascal 7.0 – СПб:БХВ. – Санкт-Петербург, 1998. – 336с.
37. Культин Н. Самоучитель: Программирование на Object Pascal в Delphi. Дюссельдорф-Киев-Москва-Санкт-Петербург, 1999. – 464 с.
38. Культин Н.Б. Delphi в задачах и примерах. СПб: БХВ – Петербург, 2004. – 288 с.
39. Культин Н.Б. Основы программирования в Delphi 7. – СПб: БХВ – Петербург, 2003.
40. Лафоре Р. Объектно-ориентированное программирование в С++. – СПб.: Питер, 2003. – 928 с.
41. Либерти Дж., Джонс Д. Освой самостоятельно С++ за 21 день. – Москва: Вильямс, 2006. – 784 с.

42. Либерти Дж., Хорват Д. Освой самостоятельно С++ за 24 часа. – Москва: Вильямс, 2006. – 448 с.
43. Марченко А.И., Марченко Л.М. Программирование в среде Turbo Pascal 7.0. / Под ред. Тарасенко В.П. – Киев: ВЕК, М.: Бином Универсал, 1998. – 496 с.
44. Мизрохи С.В. Turbo Pascal и объектно-ориентированное программирование. – М.: Финансы и статистика, 1992. – 192с.
45. Немнюгин С.А. Turbo Pascal: практикум. – СПб.: Питер, 2000.
46. Немнюгин С.А. Turbo Pascal: учебник. – СПб.: Питер, 2000.
47. Окулов С.М. Программирование в алгоритмах – Москва: БИНОМ, 2004. – 341 с.
48. Офицеров Д.В., Долгий А.В., Старых В.А. Программирование на персональных ЭВМ. Практикум.– Минск, “Высшая школа”, 1993. – 255с.
49. Павловская Т.А. С/С++. Программирование на языке высокого уровня С/С++. – СПб.: Питер, 2002. – 464 с.
50. Павловская Т.А., Щупак Ю.А. С/С++. Объектно-ориентированное программирование: Практикум. – СПб.: Питер, 2004. – 265 с.
51. Павловская Т.А., Щупак Ю.А. С/С++. Структурное программирование: Практикум. – СПб.: Питер, 2003. – 240 с.
52. Паппас Крис, Мюррей Уильям. Программирование на С и С++. - Киев: ВНУ, 2000. – 320 с.
53. Пильщиков В.Н. Сборник упражнений по языку Паскаль. – М.: Наука, 1989. – 160с.
54. Поляков Д.Б., Круглов Н.Ю. Программирование в среде Turbo Pascal (версия 5.5). – М.: Изд-во МАИ, 1992.
55. Пономарёв В. Программирование на С++/С# в Visual Studio .NET 2003. – СПб.: БХВ, 2004. – 352 с.
56. Прата С. Язык программирования С++: Лекции и упражнения. – Москва: ДиаСофтЮП, 2005. – 1104 с.
57. Программирование на С++: Учебное пособие/ Под. ред. А.Д.Хомоненко - СПб.: КОРОНА-ПРИНТ, 2003. – 512 с.
58. Рютген Т., Франкен Г. Turbo Pascal 6.0. – Киев: Торгово-издательское бюро ВНИ, 1992.
59. Савенко Р.Г., Лисенко М.В., ПК: Работа з поширеними програмними продуктами. Полтава, 1995.
60. Сергеев А.П., Терен А.Н. Программирование в Microsoft Visual С++ 2005: Самоучитель. – Москва: Вильямс, 2006. – 352 с.
61. Сердюченко В.Я. Розробка алгоритмів та програмування мовою Turbo Pascal: Навчальний посібник для технічних вузів. – Харків: ВКП “Паритет” ЛТД, 1995. – 352 с.
62. Складаров В.А. Язык С++ и ООП. - Минск: Выш. шк., 1997. – 478 с.

63. Страуструп Б. Дизайн и эволюция языка С++. – СПб.: Питер, 2006. – 448 с.
64. Turbo Pascal – Издательская группа К.: ВНУ, 2000.
65. Учимся программировать: Pascal 7.0. Задачи и методы их решения – 2-е издание, переработанное и дополненное. – М: Диалог и Мифи, 1999.
66. Учимся программировать: Pascal 7.0. Задачи и методы их решения – 2-е издание, переработанное и дополненное. – М: Диалог и Мифи, 1999.
67. Фаронов В.В. Программирование на языке высокого уровня Delphi. – СПб: Питер, 2003. – 640 с.
68. Фаронов В.В. Программирование на языке высокого уровня Delphi. – СПб: Питер, 2003. – 640 с.
69. Фаронов В.В. Турбо Паскаль 7.0. Навчальний курс. Учебное пособие – М.: “Нолидж”, 1999 – 616с.
70. Фаронов В.В. Турбо Паскаль. – М.: МВТУ, 1992 – 304с.
71. Федоров А. Особенности программирования на Borland Pascal. – Киев, Диалектика
72. Фигурнов В.Е. IBM PC для пользователя (кр.курс) 7-е изд. – 1997.
73. Черняхівський В.В. Delphi 4: сучасні технології візуального програмування. – Львів: БаК, 1999. – 196 с.
74. Шеферд Дж. Программирование на Microsoft Visual С++ .NET. – Москва: Издательско-торговый дом «Русская редакция», 2003. – 928 с.
75. Шилдт Г. Самоучитель С++. – СПб.: БХВ, 2006. – 688 с.
76. Юркин А. Задачник по программированию. – СПб.: Питер, 2002.–192 с.



Національний університет  
водного господарства  
та природокористування

## Основні скорочення

БД – база даних

ООП – об'єктно-орієнтоване програмування

ПК – персональний комп'ютер

ПЗ – програмне забезпечення

СКБД – система керування базами даних

BDE – Borland Database Engine

BLOB – Binary Large Object

BP – середовище програмування Borland Pascal

CLX – Cross-Platform Component Library

CORBA – Common Object Request Broker Architecture

DCOM – Distributed Component Object Model

HTML – Hypertext Markup Language

MIDAS – Multi-tier Distributed Application Services Suite

OLE – Object Linking and Embedding

TP – Turbo Pascal

SQL – Stardart Query Language

VCL – Visual Component Library

C – мова програмування C

C++ – мова програмування C ++

C# – мова програмування C#

OP – мова програмування Object Pascal

DLL – динамічна бібліотека

## Глосарій

**BDE** – це набір драйверів, що забезпечують доступ до даних.

**Delphi** – одна з найбільш потужних візуально об'єктно-орієнтованих систем програмування, що дозволяє на найсучаснішому рівні створювати як окремі прикладні програми, орієнтовані на роботу в **Windows**, так і розгалужені комплекси, що призначені для роботи в корпоративних мережах та в **Internet**.

**Object Pascal** – мова програмування, в основі якої лежить **Pascal**, реалізована за допомогою принципів **ООП**.

**Абстракція даних** – це можливість визначати нові типи даних, з якими можна працювати майже так само, як із основними.

**Алгоритм** – система правил, яка описує скінченну послідовність дій, необхідних для розв'язання задачі.

**Алгоритмізація задачі** – розробка алгоритму її розв'язання.

**Арифметичний вираз** являє собою сукупність констант, змінних, звертань до стандартних функцій і функцій користувача, з'єднаних знаками арифметичних операцій та круглих дужок.

**База даних** – це спеціальне електронне сховище інформації, доступ до якого здійснюється за допомогою одного або декількох комп'ютерів.

**Блок-схема алгоритму** – це сукупність, з'єднаних лініями, стандартних блоків, кожен з яких позначає певне правило (дію) алгоритму.

**Виконуваний файл** – це виконуваний файл додатку;

**Вираз** – це послідовність знаків операцій, операндів та круглих дужок, які в сукупності задають обчислювальний процес для отримання результату.

**Віддалені бази даних** – це бази даних, які розміщуються на сервері мережі, який також називають віддаленим сервером, а додатки, які працюють з цією базою даних, розміщуються на комп'ютері користувача, що відповідає архітектурі **клієнт-сервер**.

**Відкомпільований файл пакету часу виконання** – це пакет часу виконання – файл бібліотеки **DLL** із певною специфікою **Delphi**.

**Візуальні компоненти** можуть відобразитись на формі під час виконання додатку.

**Вказівник** – це змінна, призначена для збереження адреси в пам'яті.

**Впорядкування масиву** – послідовне розміщення елементів масиву в комірках пам'яті ПК.

**Дескриптор** – це є число цілого типу, яке встановлюється для кожного файлу операційною системою.

**Динамічна бібліотека** – створюється при проектуванні DLL.

**Динамічний масив** – це масив, пам'ять для якого визначається під час виконання програми.

**Елементи масиву** – це дані, які складають масив.

**Змінні** – це елементи програми, що можуть змінювати своє значення у процесі її виконання.

**Зовнішня пам'ять** – це пам'ять на магнітних дисках.

**Ідентифікатор** – це послідовність літер і цифр, яка починається з літери або знаку підкреслювання.

**Індекси** – це є комбінація полів таблиці, по яких має проводитись сортування записів таблиці.

**Інкапсуляція** – це механізм, який об'єднує дані і код, що маніпулює цими даними, а також захищає їх (і дані, і код) від зовнішнього втручання.

**Інспектор об'єктів** є інструментом середовища розробки **Delphi**. Він дозволяє редагувати власний компонент і визначені для нього обробники подій (реакцію на стандартні події).

**Канва (Canvas)** – це область компоненти, на якій можна малювати або відображати готові зображення.

**Клас** – тип даних, який містить поля даних і методи їх обробки (аналог обробки об'єкта в TP/VP).

**Ключ** – комбінація полів набору даних, які однозначно визначають кожен запис в таблиці.

**Код** – текст програми.

**Коментар** – це є досить зручний засіб, який використовується для пояснення коду програми або при налагоджуванні програми для тимчасового виключення деяких блоків.

**Компонента** – клас, ініціалізацію якого можна провести під час розробки форми, який має зручний інтерфейс для зміни своїх властивостей.

**Константи** – це елементи програми, значення яких не змінюється в процесі її виконання.



**Логічний вираз** являє собою сукупність констант, змінних, звертань до стандартних функцій і функцій користувача, з'єднаних знаками арифметичних та логічних операцій і круглих дужок.

**Локальні бази даних** – це бази даних, які розміщуються разом з додатком, що працює з ними, на одному комп'ютері.

**Масив** – це структура даних, під якою розуміється впорядкована сукупність скінченного числа однорідних даних простої або складної структури, об'єднаних одним іменем.

**Методи** – це функції та процедури, що забезпечують всі необхідні операції над даними.

**Наслідкування** – це принцип ООП, завдяки якому один об'єкт може набувати властивостей іншого.

**Невізуальні компоненти** відображаються на формі тільки на етапі проектування у вигляді піктограм.

**Об'єкти проекту** – за термінологією візуального програмування – це діалогові вікна і елементи управління, що є в діалогових вікнах (командні кнопки, поля введення текстової інформації, перемикачі та інше).

**Об'єктний файл модуля** – це відкомпільований файл модуля, який компонується у результуючий виконуваний файл.

**Об'єктно-орієнтований підхід** – метод, який дає можливість при розв'язанні складних задач в певній предметній області виділити та описати конкретні об'єкти, їх характеристики, зв'язки та взаємодію між ними.

**Оперативна пам'ять** зберігає поточну інформацію (виконувані програми і дані) в даний момент часу, а також ряд допоміжних програм для роботи комп'ютера.

**Палітра компонент** – це набір (контейнер) компонент *Delphi*, який являє собою каталог візуальних і прикладних об'єктів.

**Панель інструментів** – панель, яка містить набір піктограм. Використовується для розташування кнопок, які відповідають деяким командам меню, що часто використовуються.

**Підпрограма** – це автономна частина програми, яка реалізує певну відносно самостійну частину загального алгоритму розв'язання задачі і допускає звертання до неї з різних частин програми.



**Персональний комп'ютер** – невелика за розмірами і вартістю універсальна мікро ЕОМ, призначена для індивідуального користування.

**Подія** – це те, що відбувається під час роботи **Delphi**-дodatка.

**Поліморфізм** – це властивість коду вести себе по-різному залежно від ситуації, що виникла в момент виконання.

**Постійна пам'ять** призначена для збереження допоміжної інформації, програм, таблиць тощо.

**Предметна область** – це частина реального світу, що підлягає вивченню з метою організації керування і у кінцевому рахунку – автоматизації.

**Програмне забезпечення** – це сукупність програм для управління роботою комп'ютера та обробки інформації, що надходить в комп'ютер.

**Проект в Delphi** – це розроблюваний **Delphi**-дodatок, що включає сукупність файлів-модулів, файлів-форм та інші файли, які зв'язані власне файлом-проекту.

**Реляційна база даних** – це є набір пов'язаних одна з одною таблиць.

**Розгалуження** – алгоритмічна структура, в якій залежно від виконання умови потрібно виконати ту чи іншу дію.

**Сервер баз даних** – це є програма, яка забезпечує керування віддаленою базою даних та видачу клієнту результатів виконання на надісланий запит.

**Середовище візуального об'єктно-орієнтованого програмування Delphi** – це графічна автоматизована оболонка над об'єктно-орієнтованою версією мови Pascal – **Object Pascal**.

**Слідування** – алгоритмічна структура, в якій кожна наступна дія виконується після завершення попередньої.

**Сортування масиву** – це процес перегрупування елементів масиву у конкретному порядку.

**Список формальних параметрів** – це сукупність імен змінних з вказанням їх типу, які служать для передачі даних з програми, яка звертається у функцію.

**Статичний масив** – це масив, для якого пам'ять виділяється під час його створення.

**Текстові файли** – це файли, які складаються із символів, що об'єднані в рядки.

**Типізований файл** – це структурований тип даних, який складається з компонент одного типу і однакової довжини.

**Трансляція** – переведення програми з мови високого рівня на мову машини.

**Універсальні мови програмування** – це мови програмування високого рівня.



**Файл** – це сукупність послідовно організованих однотипних даних, які зберігаються на зовнішньому носії інформації.

**Файл групи проектів** – це файл, що створюється при роботі з групою проектів.

**Файл діаграм** – це файл діаграми, що створюється на сторінці діаграм Редактора коду.

**Файл змісту пакета** – це двійковий файл, що містить заголовок пакета та список пакетів, що йому належать.

**Файл інформації про пакети** – це бінарний файл, що використовується *Delphi* при роботі із пакетами.

**Файл конфігурації** – це файл, що зберігає конфігурацію всіх вікон.

**Файл модуля** – кожній створеній формі відповідає файл модуля, що використовується для збереження коду (можна створювати модулі, які не пов'язані із формами).

**Файл опису форми** – це файл, який містить інформацію про форму.

**Файл пакета** – це двійковий файл пакета (package).

**Файл параметрів проекту** – це файл, в якому зберігаються установки параметрів проекту.

**Файл послідовного доступу** – це набір даних, в якому до певної інформації можна отримати доступ, послідовно переглянувши дані з його початку до потрібного елемента.

**Файл проекту** – це текстовий файл, що використовується для збереження інформації про форми та модулі; містить оператори ініціалізації та запуску програми на виконання.

**Файл прямого доступу** – це набір даних, в якому до потрібної інформації можна отримати доступ безпосередньо, знаючи тільки її розміщення в файлі.

**Файл ресурсів** – це файл, який містить ресурси, що використовуються проектом.

**Файли без типу** – це двійкові файли, які можуть містити найрізноманітніші дані у вигляді послідовності байтів.

**Файли допомоги** – це стандартні файли допомоги *Windows*, які можуть використовуватися додатком *Delphi*.

**Файли зображень** – це файли зазвичай використовуються в додатках.

**Файли резервних копій** – це відповідно файли резервних копій для файлів проекту, форми та модуля.

**Фактичні параметри** – параметри, які вказуються при виклику підпрограм-функції.

**Форма** – це діалогове Windows-вікно програми, яке відкривається під час роботи *Delphi*-додатку на етапі розробки програми, на яке розміщуються об'єкти проекту.

**Фрейм** – це панель, тобто деякий фрагмент вікна додатку, який можна переносити на різні форми, в різні додатки, при цьому допускається використання переваг наслідування.

**Цикл** – багаторазове виконання дії або групи дій.



Національний університет  
водного господарства  
та природокористування




Національний університет  
водного господарства  
та природокористування

## РОЗДІЛ I. Основи алгоритмізації

### 1.1. Основні етапи розв'язання задач з використанням комп'ютерів

Розв'язання задач з використанням сучасних комп'ютерів передбачає такі етапи:

- 
- 1. Вивчення предметної області.**
  - 2. Змістова постановка задачі.** На цьому етапі в результаті вивчення предметної області чітко формулюється зміст, мета задачі, описуються вхідні дані та форма представлення результату її розв'язання, вказуються обмеження задачі.
  - 3. Побудова моделі задачі (математичної, імітаційної, комп'ютерної і т.д.).**
  - 4. Алгоритмізація задачі** – розробка алгоритму її розв'язання.
  - 5. Програмування алгоритму** – реалізація алгоритму на одній з алгоритмічних мов програмування.
  - 6. Реалізація програми на комп'ютері** – відлагодження програми (виявлення та виправлення помилок).
  - 7. Тестування програми на контрольному(их) прикладі(ах).**
  - 8. Проведення чисельних експериментів.**
  - 9. Аналіз отриманих результатів та прогнозування.**

В кожній конкретній задачі не обов'язково всі перелічені пункти можуть бути присутні.



### 1.2. Поняття про алгоритмізацію та алгоритм

Під **алгоритмізацією** розуміють процес зведення задачі до чіткої послідовності етапів, які виконуються один за одним. Результатом алгоритмізації є алгоритм. **Алгоритмом** називається система правил, яка описує скінченну послідовність дій, необхідних для розв'язання задачі. Сам термін “алгоритм” походить від імені древньосхідного математика Аль-Хорезмі.

Алгоритм характеризується наступними **властивостями**:

- **дискретністю** (означає, що процес перетворення вихідних даних у кінцевий результат розв'язання задачі відбувається не неперервно, а дискретно, окремими кроками);
- **визначеністю** або **детермінованістю** (означає, що кожне правило алгоритму повинно бути чітким і однозначним);
- **результативністю** або **скінченністю** (алгоритм повинен приводити до розв'язку задачі за скінченне число кроків);

- **масовістю** (означає, що алгоритм розв'язання задачі розробляється у загальному вигляді так, щоб його можна було застосувати для цілого класу задач, які відрізняються лише початковими даними).

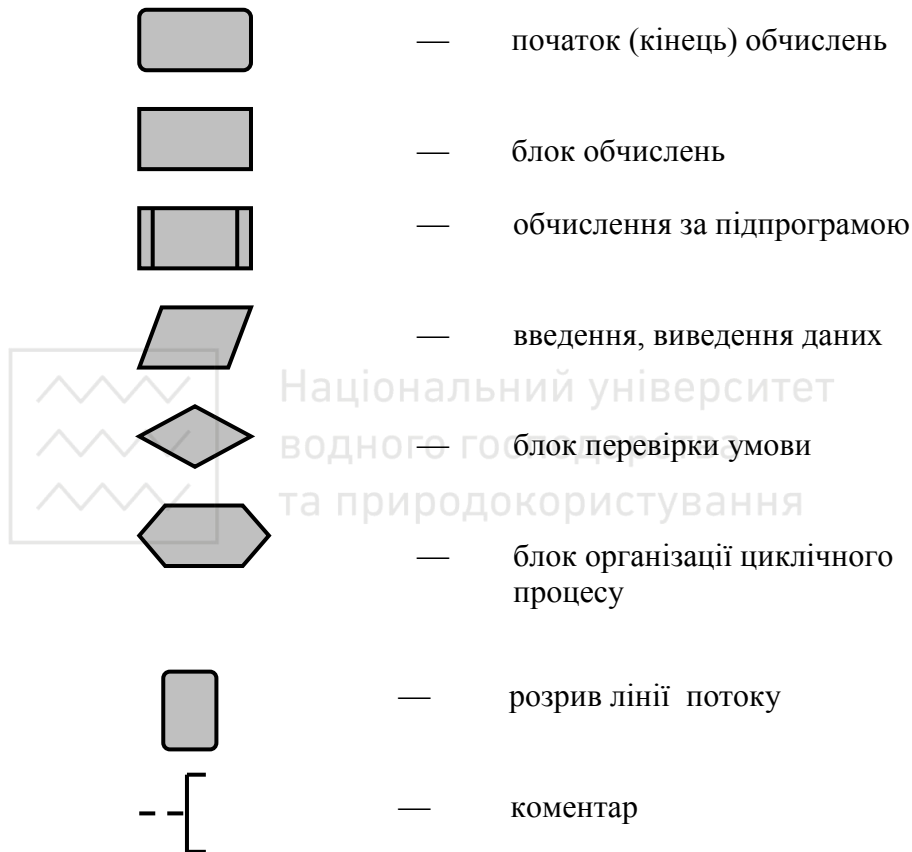
Алгоритм можна описати одним із таких способів:

- **словесним** (звичайною розмовною мовою);
- **графічним** (у вигляді блок-схеми);
- **операторним** (в спеціальних кодах за О.О. Ляпуновим);
- запис алгоритму **на одній з мов програмування**.



Найбільш наочним є опис алгоритму у вигляді блок-схеми. **Блок-схема алгоритму** – це сукупність, з'єднаних лініями, стандартних блоків, кожен з яких позначає певне правило (дію) алгоритму.

Основними блоками блок-схем алгоритмів є такі:



Кожен блок в алгоритмі нумерується, як правило, зверху блоку.

### 1.3. Основні алгоритмічні структури

Послідовності дій, описувані будь-яким алгоритмом, можна поділити на такі **алгоритмічні структури**:

- **слідування** – кожна наступна дія виконується після завершення попередньої (рис.1.1);
- **розгалуження** – залежно від виконання умови потрібно виконати ту чи іншу дію. Розрізняють *повне розгалуження* (рис.1.2) або *обхід* (рис.1.3);



Національний університет  
водного господарства  
та природокористування

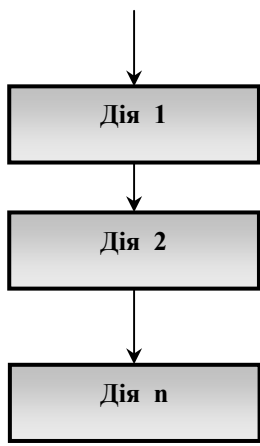


Рис. 1.1. Алгоритмічна структура слідування (лінійна структура)

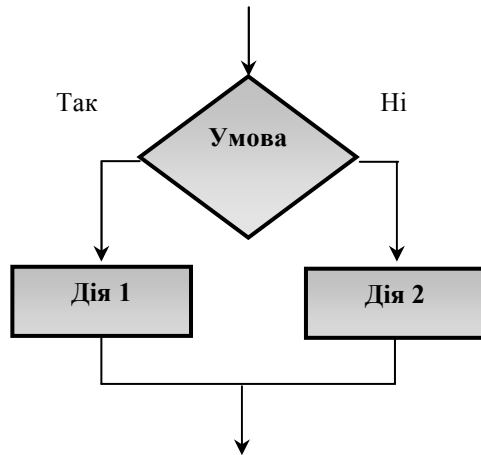


Рис. 1.2. Алгоритмічна структура повного розгалуження

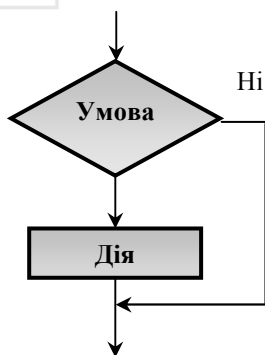


Рис. 1.3. Алгоритмічна структура неповного розгалуження – “обхід”



Рис. 1.4. Цикл з передумовою

- **цикли** – багаторазове виконання дії або групи дій. Повторювані дії називаються **тілом циклу**.

Для організації циклу необхідно задати: початкові присвоєння до входження в цикл, організувати тіло циклу та задати умову завершення циклу. Крім того, в циклі необхідно передбачити **крок зміни параметра циклу**, інакше це буде нескінченний цикл або "зациклювання".

Спосіб задання умови завершення циклу визначає **такі види циклів**:



- **цикл з передумовою** (умова повторення перевіряється щоразу перед виконанням тіла циклу – рис.1.4);

- **цикл з післяумовою** (умова завершення циклу перевіряється щоразу після виконання тіла циклу – рис.1.5);

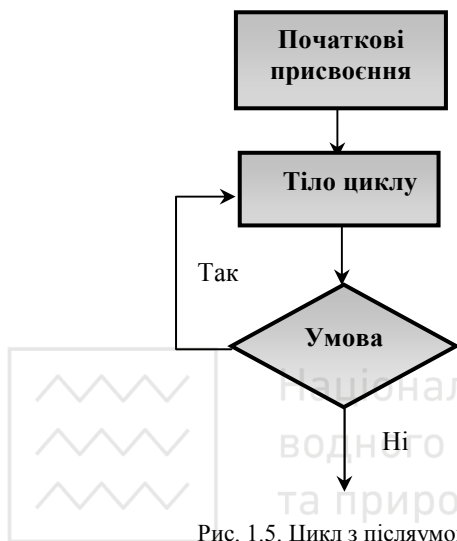


Рис. 1.5. Цикл з післяумовою

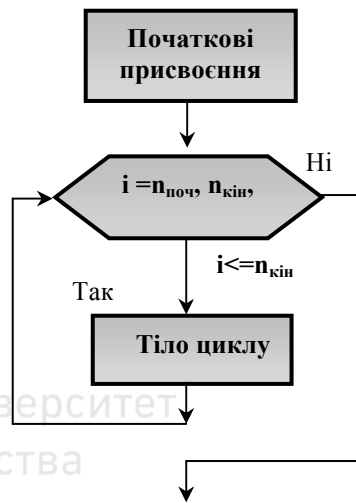


Рис. 1.6. Цикл з параметром

- **цикл з параметром** (повторення послідовності дій відбувається при різному значенні деякої змінної, яка називається **лічильником циклу**; значення параметру циклу змінюється від деякого початкового значення ( $n_{поч}$ ) із заданим кроком до деякого кінцевого значення ( $n_{кін}$ ); досягнення параметром його кінцевого значення є умовою завершення циклу – рис. 1.6. Тут  $i$  – лічильник або параметр циклу).

Залежно від використання вказаних алгоритмічних структур розрізняють відповідно **лінійні**, **розгалужені** та **циклічні обчислювальні процеси**. Алгоритми розв'язання практичних задач здебільшого поєднують у собі всі перелічені вище алгоритмічні структури.

**Контрольні запитання:**

1. Сформулюйте основні етапи розв'язання прикладної задачі на комп'ютері.
2. Дайте означення алгоритму.
3. Які є способи опису алгоритму?
4. Назвіть основні алгоритмічні структури.
5. Перерахуйте основні властивості алгоритму та розкрийте їх зміст.
6. Яка відмінність роботи алгоритму "циклу з передумовою" в порівнянні з "циклом з післяумовою"?
7. Що таке алгоритмізація?



Національний університет  
водного господарства  
та природокористування



Національний університет  
водного господарства  
та природокористування



## РОЗДІЛ II. Програмування на мові Pascal

### Лабораторна робота №1

**Тема: Математичні та логічні вирази. Оператори присвоєння. Стандартні математичні функції**



Національний університет  
водного господарства  
та природокористування

**Мета роботи:** Ознайомлення студентів з операторами присвоєння, стандартними функціями, правилами запису математичних та логічних виразів.

#### Хід роботи:

1. Записати на одній з мов програмування математичний та логічний вирази.
2. Відтворити за заданими формулами запис математичного виразу.
3. Відтворити значення числа за його представленням у пам'яті ПК.
4. Виправити синтаксичні помилки у заданому записі математичних виразів.
5. Захистити лабораторну роботу.



#### Завдання 1.1

Записати мовою програмування математичний вираз, вибираючи ідентифікатори змінних відповідного типу по замовчуванню (без явного опису типу). Звернути увагу на тип аргументів стандартних функцій.

#### Приклад 1.1

$$z = \left( \frac{|x-1| + e^{-x} - 12,34}{\lg \sqrt{|x|} - \cos x^3} \right)^{-0,4};$$

#### Розв'язання:

Запишемо приклад розв'язання даного завдання на мові програмування Pascal.

```
z:=exp((-0.4)*ln(((abs(x-1)+exp(-x)-12.34)/(ln(sqrt(abs(x)))/ln(10)-cos(x*x*x))));
```

**Завдання для самостійної роботи:**

$$1. S = \log_c \frac{c \cdot e^{-2,5c+x} + \operatorname{arctg}^2 |c-x|}{(-1)^{-2,5c} + \sqrt{|\ln|x| + \lg|c||}};$$

$$2. k = \sqrt{\frac{\sin^2 m \cdot y + \cos^2 \frac{y}{m} + 0,64}{\lg|m \cdot y| + \ln|m^2 - y^3| + e^{-(m-y)}}};$$

$$3. i = \arcsin^3 \left( \frac{2,5x^{-2} + \sqrt{|x-3|^3}}{\operatorname{tg} \frac{5,4}{x} + \ln(x+0,3)^5} + 1 \right);$$

$$4. j = e^{\lg m} \cdot \frac{\sqrt{\arcsin m^{-3} + \operatorname{arctg} \frac{\pi}{3}}}{\log_x (13,4 \cdot x^{-m} + |\frac{\pi}{5} - x|)};$$

$$5. l = \frac{1}{e^{-kx+0,5}} \cdot \left( \frac{\lg|k+x| - \sqrt{\sin^4 x}}{\operatorname{arctg} \frac{x+1}{x-k} + \frac{\pi}{10} \ln \pi| + 1} + 2 \right);$$

$$6. t = (m-y)^{-|x+1|} \cdot \frac{\ln|m-y| + \cos^3 m \cdot y + 0,01}{\sqrt{|m+y|^3} + 17,14 \cdot \lg \frac{\pi}{3}};$$

$$7. q = \frac{e^{-3,5|z| + \sqrt{14}}}{\operatorname{tg} z + \operatorname{ctg} \frac{\pi}{14} + z} - \frac{\operatorname{arctg}^3(z-1)}{\ln z^4 + \ln z^6};$$

$$8. \varphi = \arccos \frac{x}{0,13} + \frac{e^{0,6x-1} - \sqrt{(x+6,1)^3}}{\ln \left| \frac{\pi}{16} - x \right| + \operatorname{tg}^2 x^3};$$

$$9. \varepsilon = e^2 \cdot \log_2 x^4 \cdot \frac{21,4(x-0,5)^2 - \cos \frac{\pi}{x}}{\sqrt[3]{|\ln x^{-1}|} - \sqrt{|x+1| + |x|}};$$

$$10. \gamma = \omega \cdot x^{-5,3} \cdot \frac{e^{\lg|x|-1} + \sqrt{(\omega \cdot x - 3) - e}}{\arccos^2 \omega^{-1} + \operatorname{arccctg} \frac{x}{\pi}};$$

$$11. \beta = \frac{e^{-x^2} - e^{\sqrt{|x-0,5|}} + 12,47x}{\lg|x+1| - \ln^3|2^x - 1| + \operatorname{ctg} \frac{x^2 - 1}{e}};$$

$$12. \alpha = (-2)^{\frac{k+1}{2}} \cdot \operatorname{arctg} \frac{e^{kx-5,1} + \cos^2 kx^3}{\ln|kx+1| - \lg|x|};$$

$$13. a = \gamma \cdot \sqrt[3]{y+0,01} + \frac{\arccos \frac{\pi+y}{3} - \ln \left| \frac{\pi}{\gamma} \right|}{\sqrt{|\pi-y|} + \sin^2 \pi y + 1};$$

$$14. b = (\beta + z)^{-e} - \frac{\cos^3 z^4 + \operatorname{tg}^4(\beta - 1)^{-2} - 0,03}{6,51 + \sqrt{|\beta - z|} + \operatorname{lg}|z - 1|};$$

$$15. d = \frac{\operatorname{arctg} x^{-0,5e}}{\omega} - \left| \frac{\sqrt{\sin^2 \cdot x^3} - \sqrt[3]{\cos^5 x^2}}{\operatorname{lg} x^2 + \ln \left| \frac{2}{x} \right|} \right|;$$

$$16. f = 217,5e^{-x+0,77} + \frac{\gamma \cdot x^e - e^{-x} + 0,1}{\sqrt{|\sin^3(x-1) + \cos \gamma|}};$$

$$17. h = \frac{\pi}{8} \sin^2 \left( \frac{x - \pi}{8} \right) - \frac{e^{-\pi} + \pi^{-e} + 0,15}{\log_x \left| \frac{\pi}{e} + 1 \right| - \operatorname{tg}^2 x};$$

$$18. n = \frac{\sin^2(\alpha + x)}{0,5 + e^{-\alpha x}} \cdot \sqrt{\frac{\ln|\alpha + 2| - \operatorname{lg}|x - 2|}{\operatorname{arctg}(\sin^2 x + \operatorname{tg}^3 \alpha)}};$$

$$19. m = \frac{(\gamma - y)^{-0,4}}{e^\gamma + e^{-y}} + \frac{\operatorname{lg}^2|y - 5,5| + \sin^2 \frac{\gamma}{4}}{\sqrt{|\gamma + y|} + \sqrt[3]{\operatorname{arctg} y}};$$

$$20. u = 0,3 \cdot \log_5 e^{-|x|+2} + \frac{\operatorname{arctg} \frac{\pi}{x} - \sin^2 \frac{x}{\pi}}{\gamma \cdot \sqrt{\ln|\pi - x|} + \operatorname{lg} \left| \frac{\pi}{x} \right|}};$$

$$21. y = \frac{\omega \cdot x^{-3,4}}{e^{\omega \cdot x}} - \frac{\operatorname{tg} \frac{x}{\omega} + \operatorname{ctg} \frac{x^2 - 1,4}{\omega \cdot x}}{\operatorname{arctg}(\sqrt{6,6 + x} - |x^4 - \omega|)};$$

$$22. x = \gamma \cdot \left( \frac{15,6 \cdot e^{-\gamma \cdot z}}{\sqrt{|\gamma + \sin^2 z|}} - \frac{\arccos \frac{\pi - z}{3} + e}{\sqrt[3]{|z + 0,5|} + \sqrt[5]{\gamma + z}} \right);$$

$$23. p = (k + 1)^{-4,3} \cdot \gamma + \frac{\operatorname{arctg}(k \cdot x) - 0,006 \cdot x}{\sqrt{|x| + 3,41} + \sqrt[3]{\sin^4 \gamma \cdot x}};$$

$$24. t = \frac{x^2 - y^2}{e^{-(x+y)}} - \sqrt{\frac{8,67 + \cos^3(x - y) + e^y}{\operatorname{arcsin}^3|y - x - 1|}} + \alpha;$$

$$25. c = (3,14)^{m-\gamma} + \arctg \gamma \cdot x - \frac{\sqrt{|\ln \gamma + \ln x|}}{\sqrt[3]{\lg \gamma} + \sqrt[5]{\cos x^{-1}}}.$$

## Завдання 1.2

Записати мовою програмування даний логічний вираз і визначити значення результату логічних операцій TRUE або FALSE при вказаних значеннях змінних.

### Приклад 1.2

або  $0,4 < \sqrt{y} \leq x$  при  $y = 24$ ;

### Розв'язання:

Запишемо приклад розв'язання даного завдання на мові програмування Pascal.

$(X \leq 16.5) \text{ or } ((\text{Sqr}(Y) > 0.4) \text{ and } (\text{Sqr}(Y) \leq X))$

При  $X=0,2$ ;  $Y=24$  набуває значення TRUE.

### Завдання для самостійної роботи:

1.  $6,4 < \sqrt{a}$  та  $b < 2a \leq 8$  при  $a = 3$ ;  $b = 0$ ;
2.  $1 \leq x^2 \leq 2$  або  $y \leq x$  при  $x = -1$ ;  $y = 0$ ;
3.  $\cos x < 1$  та  $x + y \geq 5$  при  $x = -1$ ;  $y = 2$ ;
4.  $\text{tgy} \leq 3$  або  $\frac{y}{4} > \sqrt{x}$  при  $x = 2$ ;  $y = 1$ ;
5.  $x - y \leq z + \sqrt{x} \leq 2y$  при  $x = 1$ ;  $y = 2$ ;  $z = 3$ ;
6.  $xy \leq 0$  та  $y \geq 4$  при  $x = -2$ ;  $y = 5$ ;
7.  $-0,7 \leq k < 1,5$  та  $z \neq 5$  при  $z = 6$ ;  $k = 1$ ;
8.  $\text{tg} x < x \leq \frac{\pi}{4}$  при  $x = 2$ ;
9.  $ab < a \leq \frac{a}{b}$  при  $a = 4$ ;  $b = 0,2$ ;
10.  $\sqrt{a} \leq x < 2,5$  при  $x = a + 2,5$ ;
11.  $y \neq x$  та  $|xy| > 1$  при  $x = -1$ ;  $y = 2$ ;
12.  $z \neq 6,5$  або  $zq \geq 2$  при  $q = -1$ ;  $z = 6,5$ ;

13.  $x \leq 15 \leq y < z$  при  $x = 17; y = 22; z = 20;$
14.  $|x| \leq 1$  або  $\left| \frac{x}{y} \right| > 3$  при  $x = 0,5; y = 0,2;$
15.  $x - y \leq z + 2$  та  $x < |z|$  при  $x = y = 2; z = 1;$
16.  $0,5 < y \leq x + \sqrt{x}$  при  $x = 2; y = 0,2;$
17.  $x^2 + y^2 \leq 4 < xy$  при  $x = -1; y = -5;$
18.  $xyz < x + y + z < 1$  при  $x = -1; y = -2; z = 8;$
19.  $0,51 \neq xy$  або  $x - y > 0$  при  $x = -1; y = -2;$
20.  $\sqrt{x - y} > xy$  та  $x \geq 3$  при  $x = 4; y = -3;$
21.  $ab \leq a + b < \frac{a}{b}$  при  $a = 3; b = 0,1;$
22.  $ma < \frac{m}{a} < m + a$  при  $a = 0,2; m = 4;$
23.  $|x - a| \leq |x| - |a| < |x|a$  при  $a = -4; x = 1;$
24.  $0,15y + x < xy \leq \left| \frac{x}{y} \right|$  при  $y = 0,5; x = 5;$
25.  $a \neq y < |2a|$  при  $a = -4; y = 1.$



### Національний університет водного господарства та природокористування

#### Завдання 1.3

У поданих нижче варіантах відтворити математичний вираз за його записом на мові програмування Pascal.

#### Приклад 1.3

Q:=ln(abs(x+1))/ln(10)-sqrt(ln(abs(exp(x\*ln(2)))))\*ln(abs(sqrt(x)));

#### Розв'язання:

$$Q = \lg|x+1| - \sqrt{\ln|2^x|} * \ln|x^2|$$

#### Завдання для самостійної роботи:

1.  $q := \sin(y/w) / \cos(y/w) + \cos(\sqrt{y}) / \sin(\sqrt{y});$
2.  $q := \cos((a-1) / \exp(1)) / \sin((a-1) / \exp(1)) + \exp((a+1) / 2 * \ln(2));$
3.  $q := \sqrt{\ln(\text{abs}(\exp(a * \ln(2)) - 1))} * \ln(\text{abs}(\exp(a * \ln(2)) - 1)) - 12.47;$
4.  $q := \exp(k - 5.1) + \ln(\text{abs}(k + a)) / \ln(10);$

5.  $q := \sqrt{\text{abs}(\sqrt{(m+n)} * (m+n))} + 17.14 * m * n;$
6.  $q := \arctan((x+1)/(y-2)) + \ln(\text{abs}(k+x)) / \ln(10);$
7.  $q := \ln(\text{abs}(\pi/a-b)) + \sqrt{\sin(\sqrt{b} * b) / \cos(\sqrt{b} * b)};$
8.  $q := \exp(1/3 * \ln(\text{abs}(\ln(1/\text{abs}(x)) - \sqrt{\text{abs}(x+1)})));$
9.  $q := \exp(-\exp(1) * \ln(2 * x)) + \pi/2 - \arctan(\sqrt{\exp(1)});$
10.  $q := (\sqrt{\sin(b * b * b/2)} - \cos(c/4) / \sin(c/4)) / (\ln(\text{abs}(b)) + \ln(c * c));$
11.  $q := \ln(\text{abs}(c+b)) / \ln(10) / (\pi/2 - \arctan(\pi/b)) + 0.17;$
12.  $q := c * \exp(-2.5 * x + \sqrt{y}) - \exp(1/3 * \ln(c * x));$
13.  $q := (\sqrt{\arctan(b-a)} * \arctan(b-a) + \exp(1/3 * \ln(a * b))) / (1 + \ln(a) / \ln(b));$
14.  $q := (\exp(-2.5 * a) + \sqrt{\sin(a * a * a)}) / (2 * \ln(\text{abs}(b * a)) / \ln(10));$
15.  $q := \arctan(\sqrt{\sin(x)} + \sqrt{\sin(\sqrt{y})} / \cos(\sqrt{y})) * \sin(\sqrt{y}) / \cos(\sqrt{y});$
16.  $q := \ln(\text{abs}(\pi / \exp(1) + 1)) / \ln(m) + \sin(m) / \cos(m);$
17.  $q := (\ln(\text{abs}(x - y)) + \sqrt{\cos(x * y)} * \cos(x * y)) / (\sqrt{\text{abs}(\sqrt{(x+y)} * (x+y))} + 17.14);$
18.  $q := \exp(1/3 * \ln(\ln(\exp(1)) / \ln(10))) + \exp(1/5 * \ln(\text{abs}(\cos(\exp(1)) - 2)));$
19.  $q := \sqrt{8.67 + \exp(y) + \text{abs}(y)};$
20.  $q := \arctan(\sqrt{\sin(x)} + \sqrt{\sin(y) / \cos(y)} * \sin(y) / \cos(y));$
21.  $q := \exp(2) * 4 * \ln(x) / \ln(2) - \sqrt{\text{abs}(y+1)};$
22.  $q := a * \exp(1/3 * \ln(\sqrt{\sqrt{\sin(b * b * b)}})) + 12.47;$
23.  $q := \sqrt{\text{abs}(b+a)} + 17.14 * \ln(\pi/3) / \ln(10);$
24.  $q := (\sqrt{x+2} + \text{abs}(t + \sin(\text{alfa}))) / (\sqrt{\cos(a)} + \sin(\sqrt{b}));$
25.  $q := (\exp(-x * y) + 17.4) / \exp(1/3 * \ln(\sqrt{\sin(x * y)}));$

#### Завдання 1.4

У поданих нижче варіантах відтворити число, представлене в експоненціальній формі.

#### Приклад 1.4

5.18E+02                      6510E-1                      4E-4

#### Розв'язання:

518                                      651                                      0,0004

#### Завдання для самостійної роботи:

- |             |                |                  |
|-------------|----------------|------------------|
| 1. 7.0E01   | 10. 3E1        | 19. -6.41E+2     |
| 2. 1.793E2  | 11. 0.435E12   | 20. 2.98E-10     |
| 3. -0.39E-2 | 12. -16.3E-03  | 21. 0.0183907E+5 |
| 4. 0.3E2    | 13. 87.143E+02 | 22. -6.65E+0     |
| 5. 57700E-2 | 14. 3.45E+00   | 23. 0.25E+15     |
| 6. 6510E-1  | 15. 5470E-1    | 24. -897E-2      |

7. 31.279E4

16. 4E-4

25. 3.789E+5

8. 12.5E2

17. 1.55E-2

9. 17.93E1

18. 27.870E0

### Завдання 1.5

Виправити синтаксичні помилки в записі наведених нижче математичних виразів, записаних на мові Pascal.

#### Приклад 1.5

$X:=2A/B \operatorname{tg}(x) \sqrt[3]{(X+Y)^{4-4}}$

#### Розв'язання:

$X:=2*A/B*(\sin(x)/\cos(x))*3*(x+y)/4-4$

#### Завдання для самостійної роботи:

1.  $y:=\operatorname{srt}(\operatorname{abc}(m^3))+2\ln(n);$
2.  $x:=\tan(5,4/m)+mn;$
3.  $z:=-m*\log(\operatorname{abc}(-x))/\ln(\pi)+\operatorname{abs}(\pi^{\sqrt{5-y}});$
4.  $y:=\exp(1/3*\ln(\operatorname{abs}(k*m-3)))+\pi/6;$
5.  $x:=\operatorname{srt}(\operatorname{abs}(\operatorname{sqrt}(m+n)(m+n)))+17,14m*n;$
6.  $z:=\operatorname{arctan}((x+1)/(y-2))+\ln(\operatorname{abc}(k+x))\ln 10;$
7.  $x:=\ln(\operatorname{abs}(\pi/\exp+1))\ln(m)+\sin/\cos(m);$
8.  $y:=\log(\operatorname{abs}(\pi/a-b))+\operatorname{sqr}(\sin(\sqrt[3]{b})/\cos(\operatorname{sqr}(b\ b)));$
9.  $x:=21,4\operatorname{sqr}(a0,5)+\cos \pi/b;$
10.  $z:=e^{2*4\ln x / \ln 2-\operatorname{sqr}(y+1)};$
11.  $z:=\log(\operatorname{abs}(x-y))+\operatorname{sqr}(\cos(x\ y))*\cos x*y;$
12.  $z:=\exp(-3,5\operatorname{abs}(x)+\operatorname{sqr}(x))\operatorname{sqr}(\operatorname{arctn}(y-1));$
13.  $x:=\operatorname{srt}(\operatorname{abs}(b+a))+17,14 \ln(\pi/3)\ln 10;$
14.  $y:=(\operatorname{sr}(\sin(b^3))-\cos(c/4)/\sin(c^4))/(\ln(\operatorname{mod}(b))+\ln(c\ c));$
15.  $z:=\exp(\log(m)/\ln 10*\ln(2))*\operatorname{srt}(m\ m\ m+2,5\ x)/\exp(-m);$
16.  $x:=\operatorname{sqr}(\operatorname{abc}(\sin())(\exp 2)+3,41));$
17.  $y:=a \exp(1/3*\ln(\operatorname{sqr}(\operatorname{sqr}(\sin())(b*b*b))))+12,47;$
18.  $y:=4 \ln(k)-6*\log(m)\ln 10;$
19.  $z:=\operatorname{arct}(\operatorname{sqr}(1-\operatorname{sqr}(0,13\ \operatorname{sqr}(x)))/(0,13/\operatorname{sqr}(x)))+\ln(\operatorname{abs}(1/y));$
20.  $x:=\cos((a-1)/\exp 1)/\sin((a-1)/\exp())+\exp((a+1)^2 \ln(2));$
21.  $y:=\exp(-\exp(1)*\ln(2\ x))+\pi^2-\operatorname{arctn}(\operatorname{sqr}(\exp 1));$
22.  $x:=\operatorname{expon}((k*y-5,1)+(\operatorname{sqr}(\cos k*y)));$

23.  $y := 0,3 \ln(\exp(-2,3)) \log(5);$

24.  $x := \exp(-\Pi i) + \exp(-\exp 1 * \ln y + 0,15);$

25.  $z := \cos(x+y) \sqrt{x-y} / \sin x + y / \sqrt{x-y} + 1,3.$

### **Контрольні запитання:**

1. Що складає алфавіт мови ТР?
2. Наведіть класифікацію лексем в ТР.
3. Як позначаються операції відношення в ТР?
4. Який пріоритет операцій в ТР?
5. Наведіть класифікацію типів даних в ТР?
6. Назвіть типи даних, які належать до дійсних, цілих?
7. Що таке змінні, де і як здійснюється їх опис в ТР програмі?
8. Дайте означення типізованої константи та наведіть її приклад.
9. Як реалізується операція піднесення до степеня в ТР?

## **Лабораторна робота №2**

### **Тема: Лінійні алгоритми та програми**

**Мета роботи:** Розробка алгоритмів обчислення значень математичних виразів, написання відповідних програм на мові програмування, реалізація завдань на ПК.

#### **Хід роботи:**

1. Розробити алгоритм обчислення значення математичного виразу.
2. Написати відповідну програму на мові програмування.
3. Виконати дане завдання на ПК.
4. Захистити лабораторну роботу.

#### **Завдання 2.1**

Розробити алгоритм та записати відповідну програму обчислення значення математичного виразу.

#### **Вимоги до програми:**

- вхідні дані ввести за допомогою стандартних процедур введення;
- на друк вивести значення вхідної змінної та результати обчислень.



## Приклад 2.1

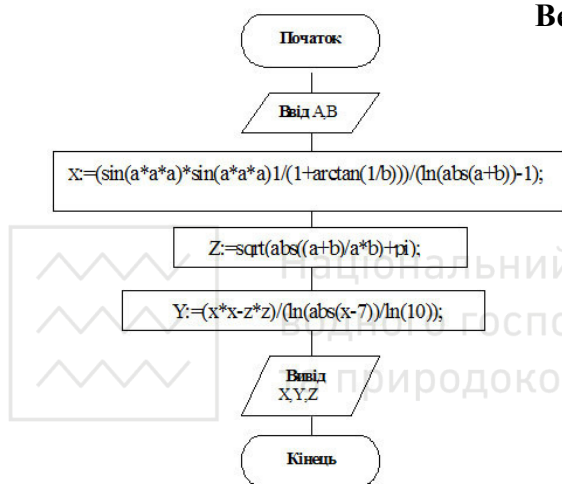
$$y = \frac{x^2 - z^2}{\lg|x-7|}, \quad x = \frac{\sin^2 a^3 - \arcsin \frac{1}{b}}{\ln|a+b|-1}, \quad z = \sqrt{\left| \frac{a+b}{ab} \right| + \pi},$$

$$a = 3,5, \quad b = -2.16;$$



**Розв'язання:** ситет водного господарства та природокористування

Запишемо приклад розв'язання даного завдання на мові програмування Pascal.



```

Program l2_z1;
Uses crt; {підключення CRT модуля}
Var {розділ опису змінних}
  A,b,x,y,z:real;
Begin {початок виконавчої частини}
  Clrscr; {очистка екрану}
  Write(' Введіть A:');
  Readln(a); {ввід значення змінної A з
  клавіатури}
  Write(' Введіть B:');
  Readln(b);
  X:=(sin(a*a*a)*sin(a*a*a)-
  1/(1+arctan(1/b)))/(ln(abs(a+b))-1);
  Z:=sqrt(abs((a+b)/a*b)+pi);
  Y:=(x*x-z*z)/(ln(abs(x-7))/ln(10));
  Writeln('При a=',a:6:2,' і
  b=',b:6:2,' x=',x:6:2,' y=',y:6:2,'
  z=',z:6:2); {вивід результатів на
  екран}
End. {кінець виконавчої частини
  програми}
  
```

### Результати:

При a=10 і b=5, x=-0.09; y=0.00; z=3.26.

### Завдання для самотійної роботи:

$$1. \quad p = \frac{e^{-xy} + 17,4}{\sqrt[3]{\sin^2 xy}}, \quad x = (a^2 + b^2)^{-4,1}, \quad y = \arctg^3 \frac{1}{b},$$

$$a = -2,004, \quad b = 0,87;$$

$$2. r = \operatorname{ctg} \frac{x+y}{(x-y)^2} + 1,3, x = \sin^4 e^{-b} + |ab|, y = \ln|a-b| + \lg \frac{\pi}{a},$$

$$a = 1,77, b = -0,62;$$

$$3. \varphi = \arccos \left( \frac{x^2}{0,13} \right)^{-1} + \ln|y^{-1}|, x = \sqrt{(k+6,1)^3}, y = \ln k^4 + \lg m^{-6},$$

$$k = 14, m = 0,42;$$

$$4. \alpha = \frac{e^{-3,5|x| + \sqrt{\pi}}}{\operatorname{arctg}^3(y-1)}, x = a + \cos \frac{\pi}{b}, y = \ln \left| \frac{\pi}{16} - b \right|, a = \frac{1}{2}, b = 1,4 \cdot 10^3;$$

$$5. t = \ln|m-y| + \cos^3 my, m = \sqrt{|x+a|} + 17,14 \cdot \lg \frac{\pi}{3}, y = a \cdot \sqrt[3]{\sin^4 x^3} + 12,47,$$

$$x = 3,4; a = -1,17;$$

$$6. \varepsilon = e^2 \cdot \log_2 x^4 - \sqrt{|y+1|}, x = 21,4(a-0,5)^2 + \cos \frac{\pi}{b}, y = \ln \left| \frac{\pi}{a} - b \right| + \operatorname{tg}^2 b^3,$$

$$a = 0,7, b = -4;$$

$$7. \gamma = \operatorname{arctg} \frac{x+1}{y-2} + \lg|k+x|, x = \sqrt{|m+n|^3} + 17,14mn, y = \sqrt[3]{|km-3|} + \frac{\pi}{6},$$

$$m = 3, n = -2,2, k = 0,801;$$

$$8. j = \log_{\pi} |x|^{-m} + \left| \frac{\pi}{5} - y \right|, x = \operatorname{arctg} \frac{5,4}{m} + mn, y = \sqrt{|m-3|} + \ln n^2,$$

$$m = -2, n = 3,87;$$

$$9. f = \frac{x^e - e^{-x} + 0,12}{\sqrt{|\sin(y-1)|}}, x = e^{-\pi} + \pi^{-e}, y = \lg a^3 - \operatorname{arctg} a, a = 6,45;$$

$$10. n = \operatorname{arctg}(\sin^2 x + \operatorname{tg}^3 y), x = \ln|\alpha + 2,3| - \lg|\beta - 3,2|, y = \sin^2(\alpha - \beta)^3,$$

$$\alpha = 15,3, \beta = -0,012;$$

$$11. b = (\beta + |z|)^{-e} + \sqrt[3]{|z| + 0,1}, \beta = e^{k-5,1} + \lg|k+x|, z = \ln^3 |2^x - 1| - 12,47,$$

$$x = 0,03, k = 4;$$

$$12. a = \sqrt{|\pi - y|} + \sin^2 \pi x + 1,67, y = \operatorname{tg}^4(\beta - 1)^2 - 0,035, x = \operatorname{ctg} \frac{\alpha - 1}{e} + 2^{\frac{\alpha+1}{2}},$$

$$\alpha = 4,4, \beta = 1,87;$$

$$13. y = ax^{-3,1} + e^{\omega z}, x = \operatorname{tg} \frac{z}{\omega} + \operatorname{ctg} \sqrt{z}, z = \sqrt[3]{\ln \omega + \ln \omega^2},$$

$$\omega = 2,77;$$

$$14. t = \frac{x^2 - y^3}{e^{-(x+y)}}, x = \sqrt{8,67 + e^y + |y|}, y = \sqrt[3]{lge} + \sqrt[5]{|\cos e - 2|};$$

$$15. x = \arccos \frac{\pi - z}{3} + e, z = \sqrt{|y + \sin^2 y|}, y = 0,3 \log_5 e^{-2,3};$$

$$16. m = \lg^2 |y - 5,5| + \sin^2 \frac{y}{4}, y = \ln |\pi - x| + \lg \left| \frac{\pi}{x} \right|, x = \sqrt{|\sin e^2 + 3,41|};$$

$$17. g = e^{-3,5 \frac{1}{|z|}} + \ln z^4, z = \sqrt[5]{(x + 6,1)^3}, x = 21,4(\alpha - 0,5)^2 - \cos \frac{\pi}{\alpha}, \alpha = 6,42;$$

$$18. t = \frac{\ln |m - y| + \cos^3 my}{\sqrt{|m + y|^3 + 17,14}}, y = (2m)^{-e} + \operatorname{arctg} \sqrt{e}, m = 2,7 \cdot 10^{-3};$$

$$19. \gamma = \sqrt[3]{|\ln |x|^{-1} - \sqrt{|x + 1|}|}, x = \frac{\arcsin \omega^{-1} + \ln |\omega|}{(-2) \cdot e^{-\omega}}, \omega = 3,47;$$

$$20. i = 2^{\lg m} \frac{\sqrt{m^3 + 2,5x}}{e^{-m}}, x = \cos^2 \frac{\pi}{y} - 29,45, y = (3m)^e, \\ m = 13,44;$$

$$21. a = \gamma \cdot \sqrt[3]{y + 0,01} + \sin^2 \pi x, y = \operatorname{tg}^4(x - 1), x = \lg |\gamma + 6,6| + 0,77, \gamma = -3,41;$$

$$22. d = \sqrt{|\sin^3(x - 1) + \cos \gamma|}, x = \log_\gamma \left| \frac{\pi}{e} + 1 \right| + \operatorname{tg} \gamma, \gamma = 23,41;$$

$$23. z = \operatorname{arctg} (\sin^2 x + \operatorname{tg}^3 y^2), x = \ln |\alpha - 2| - \lg |y + 2|, y = e^{-\alpha} + \frac{\pi}{8}, \\ \alpha = 4,45;$$

$$24. h = \frac{\pi}{8} \sin^2 \left( \frac{x - y}{8\pi} \right), x = e^{-\pi} + y^{-e} + 0,15, y = \arccos(\pi e)^{-1};$$

$$25. z = \lg |x + 1| - \ln^3 |2^x - 1|, x = e^{ky - 5,1} + \cos^2 ky, y = \sqrt{|\pi - e|}, k = 2,26.$$

## Завдання 2.2

Для даного завдання розробити алгоритм та написати відповідну програму з виведенням результатів на екран.

### Вимоги до програми:

- вхідні дані ввести за допомогою стандартних процедур введення;
- на друк вивести значення вхідної змінної та результати обчислень;

- вхідні дані взяти довільними.

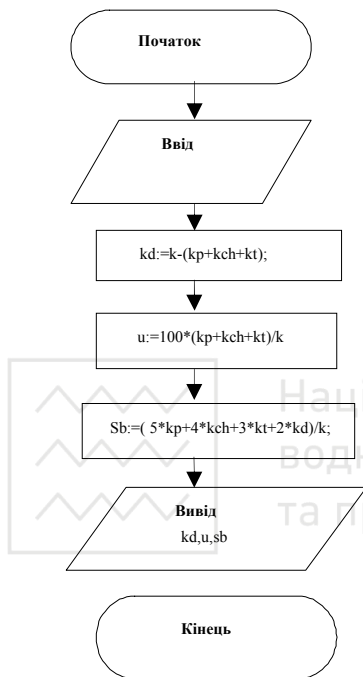
## Приклад 2.2

Скласти програму для обчислення кількості студентів-двієчників, відсотка успішності студентів групи та середнього балу, якщо задано кількість студентів в групі, які отримали “5”, “4”, “3” на іспиті з інформатики.



**Розв’язання:** ситет  
водного господарства  
та природокористування

Запишемо приклад розв’язання даного завдання на мові програмування Pascal.



```

Program l2_z3;
Uses crt;
Var
  k,kp,kch,kt,kd:integer;
  u,sb:real;
Begin
  Clrscr;
  Writeln('Введіть к-ть студентів групи');
  Readln(k);
  Writeln('Введіть к-ть студентів, які
отримали 5,4,3 відповідно');
  Readln(kp,kch,kt);
  kd:=k-(kp+kch+kt);
  Writeln('В групі ',kd:5,' двієчників');
  u:=100*(kp+kch+kt)/k;{Обчислення
відсотка успішності}
  Sb:=(
          5*kp+4*kch+3*kt+2*kd)/k;
{Обчислення середнього балу групи}
  Writeln('Відсоток успішності становить
',u:4:1,'%');
  Writeln('Середній бал групи: ',sb:4:1);
End.
  
```

### Результати:

k=25; kp=5; kch=10; kt=5; kd=5; u=80%; sb=3,6.

### Завдання для самостійної роботи:

1. Дано три числа  $a$ ,  $b$ ,  $c$ , які задають довжини сторін деякого трикутника. Обчислити висоту трикутника, опущену на сторону  $c$ .
2. Дано три числа  $a$ ,  $b$ ,  $c$ , які задають довжини сторін трикутника. Визначити медіану трикутника, проведenu до сторони  $b$ .

3. Дано три числа  $a, b, c$ , які задають довжини сторін трикутника. Визначити бісектрису трикутника, проведену до сторони  $b$ .
4. Дано три числа  $a, b, c$ , які задають довжини сторін трикутника. Визначити периметр трикутника.
5. Дано три числа  $a, b, c$ , які задають довжини сторін трикутника. Визначити площу трикутника.
6. Дано кут  $\alpha$  в радіанах. Визначити його градусну міру.
7. Скласти програму знаходження площі прямокутного трикутника, якщо відомі довжини двох його катетів  $a$  і  $b$ .
8. Знаючи площу прямокутного трикутника і довжину одного катета  $a$ , знайти периметр даного трикутника.
9. Знаючи площу прямокутного трикутника і кут при основі, знайти всі сторони трикутника.
10. Відомо периметр рівностороннього трикутника. Знайти його площу.
11. Знайти площу круга діаметром  $d$ .
12. Відома довжина кола  $L$ . Знайти площу круга, обмеженого цим колом.
13. Огорожа має форму кола і обмежує ділянку площею  $S$ . Якою буде сторона квадрата, якщо цією огорожею обмежити квадратну ділянку?
14. Дано координати точок  $A, B, C$  і  $D$ , які є вершинами трапеції. Визначте площу даної трапеції.
15. Яка довжина математичного маятника, якщо за час  $t$  він зробить  $n$  коливань?
16. Обчислити період коливання математичного маятника довжиною  $L$ .
17. Визначити опір кола, створеного з 4 - резисторів, опори яких відповідно  $R_1, R_2, R_3, R_4$ , з'єднаних: а) послідовно, б) паралельно.
18. Довжина одного маятника  $L_1$ , другого –  $L_2$ . На скільки відрізняються періоди їх коливань?
19. Вантажний автомобіль масою  $m$  рухається по інерції з силою  $F$ . Визначити його прискорення.
20. Дано коло з центром в т. $O(x_0, y_0)$  і координати точок на колі  $A(x_a, y_a), B(x_b, y_b), C(x_c, y_c), D(x_d, y_d)$ . Знайти різницю периметрів трикутників  $AOC$  і  $BOD$ .
21. Обчислити об'єм кулі і площу сфери  $V$  радіусом  $R$ .
22. Дано квадрат  $ABCD$ . Точки  $A(1;2), B(5; -1)$  – вершини квадрата. Знайти координати вершини  $D$ .
23. Дано трикутник, де  $A(x_a, y_a), B(x_b, y_b)$  і  $C(x_c, y_c)$ . Знайти його площу.
24. Дано три сторони трикутника. Визначити його кути. Вказівка: Згідно теореми косинусів кут між сторонами  $a$  і  $b$  дорівнює  $\arccos \frac{a^2 + b^2 - c^2}{2ab}$ . Для обчислення арккосинуса використати співвідношення  $\arccos(x) = \arctg \sqrt{\frac{1-x^2}{x}}$ .

25. Яка висота телевізійної башти в Останкіно, якщо кулька, падаючи з башти без початкової швидкості, останні  $k$  м шляху пролетить за  $t$  хв.? Опір повітря при цьому не враховувати ( $k=185$  м,  $t= 2$  хв.)

### **Контрольні запитання:**

1. На які групи поділяються оператори мови TP?
2. Які оператори називаються простими?
3. Як виконується оператор присвоєння?
4. Якими операторами здійснюється введення та виведення даних в TP?
5. Як здійснюється «затримка» вікна результатів на екран монітора в TP?
6. Які є способи інтерпретацій даних в TP?
7. Яка особливість при введенні даних: на принтер; у файл даних?
8. Яку алгоритмічну структуру реалізує лінійний обчислювальний процес?
9. Як вивести число в заданому форматі?
10. Чим відрізняються оператори read(), readln(), write(), writeln()?



Національний  
університет  
водного господарства  
та природокористування

### **Лабораторна робота №3**

#### **Тема: Алгоритми та програми розгалуженої структури**

**Мета роботи:** Освоєння алгоритмів та програм розгалуженої структури, складання програм з використанням операторів умовного та безумовного переходу

#### **Хід роботи:**

1. Розробити алгоритми розгалуженої структури для поданих нижче завдань.
2. Написати відповідні програми з використанням умовного оператора чи оператора вибору case.
3. Визначені викладачем завдання виконати на ПК.
4. Захистити лабораторну роботу.

#### **Завдання 3.1**

Розробити алгоритм та записати відповідну програму знаходження значення функції, яка обчислюється залежно від значення аргументу.

### Вимоги до програми:

- вхідні дані ввести за допомогою стандартних процедур введення;
- на друк вивести значення вхідної змінної та результати обчислень;
- вхідні дані взяти довільними.

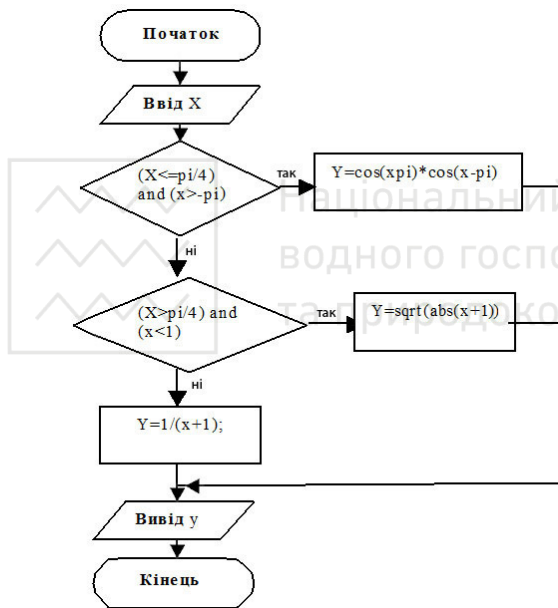
### Приклад 3.1



$$y = \begin{cases} -\cos^2(x - \pi), & -\pi < x < \frac{\pi}{4}; \\ \sqrt{|x+1|}, & \frac{\pi}{4} \leq x \leq 1; \\ \frac{1}{x-1}, & 1 < x. \end{cases} \quad x \in \mathbb{R}$$

### Розв'язання:

Запишемо приклад розв'язання даного завдання на мові програмування Pascal.



### Program l3z1;

```

Uses Crt;
Var X,y:real;
Begin
  Clrscr;
  Writeln('Введіть X:');
  readln(x);
  If (x>-pi) and (x<pi/4) then y:=
    -(cos(x-pi)*cos(x-pi)); {Перевірка
    належності X до 1-го інтервалу}
  If (x>=pi/4) and (x<=1) then
    y:=sqrt(abs(x+1));
  If x>1 then y:=1/(x+1);
  Writeln('При X=',x:6:2,'
  y=',y:6:2);
End.
  
```

Результат: x=0,56, Y=1.25.

**Завдання для самостійної роботи:**

$$1. \quad y = \begin{cases} e^{-x} + |x^2 - 1|, & x > 1; \\ \lg \sqrt{|1-x|}, & -\pi < x \leq 1. \end{cases}$$

$$2. \quad y = \begin{cases} 2x\sqrt{x^2 + 30}, & 1 < x < 20,4; \\ \operatorname{arctg}(x-7), & 0 < x \leq 1; \\ e^{x+4}, & x \leq 0. \end{cases}$$

$$3. \quad y = \begin{cases} -\sqrt[3]{cx}, & c > 9, x < -1; \\ \operatorname{ctg} \frac{c}{x}, & c < 0, -1 < x \leq 1; \\ \ln|c^2 - x^2|, & c > 0, x < c. \end{cases}$$

$$4. \quad y = \begin{cases} \lg \left| \frac{\pi}{16} - x \right|, & 0 < x < \frac{1}{4}; \\ (x^2 - 2,04)^{-3,14}, & \frac{1}{4} \leq x < 1; \\ \arccos \frac{x}{4}, & x \geq 1. \end{cases}$$

$$5. \quad z = \begin{cases} \frac{x+y}{e^{xy}}, & |xy| < 1, x < 0; \\ -\ln^2 x, & 2 < x, y \leq 0; \\ \lg \sqrt{y}, & 0 < y, 0 \leq x \leq 2. \end{cases}$$

$$6. \quad y = \begin{cases} e^{-|x|}, & 1 \leq x; \\ \lg \sqrt{1-x^2}, & |x| < 1; \\ \operatorname{arctg} x, & x \leq -1. \end{cases}$$

$$7. \quad y = \begin{cases} x^e - e^{-x}, & |x| < 2; \\ \lg x^2, & x \leq -2; \\ \sin^2 x, & x \geq 2. \end{cases}$$

$$8. \quad y = \begin{cases} 2^{x-1} + 2,71, & \pi \leq x < 8,5; \\ \sqrt{|\pi - x|}, & 8,5 < x < \pi; \\ 2,7, & 8,5 \leq |x|. \end{cases}$$



$$9. y = \begin{cases} 0, & x \leq -10; \\ ctg \frac{x-1}{e}, & -10 < x \leq 0; \\ \ln x, & 10 \leq x; \\ \sqrt{x^3}, & 0 < x < 10. \end{cases}$$

$$10. y = \begin{cases} \sqrt[3]{\lg x + \ln x^2}, & x > 1; \\ e^{-x} + 1, & x \leq 1. \end{cases}$$

$$11. y = \begin{cases} \sin e^x - 2, & |x| \leq 4; \\ \frac{x^2 - 1,2}{x + 4}, & 10 > |x| > 4; \\ x, & x \geq 10; \\ 1,5, & x \leq -10. \end{cases}$$

$$12. y = \begin{cases} \arccos \frac{2}{\pi - x}, & x < -1; \\ e^{-x^2}, & |x| < 1; \\ \pi \ln^2 x, & x > 1; \\ 10^{-3}, & |x| = 1. \end{cases}$$

$$13. y = \begin{cases} \ln|x-z|, & x \geq 2z, \quad |z| > 1; \\ \lg^2 \left| \frac{x}{z} - 1 \right|, & 0 < |z| < 1, \quad x > z; \\ tg^3(x+z), & z \leq x, \quad |z| > 1; \\ 5,6 \cdot 10^{-3}, & x \leq z, \quad |z| < 1. \end{cases}$$

$$14. y = \begin{cases} 3x^{-3}, & 1 < x \leq 12,5; \\ 13,44, & x > 12,5; \\ \operatorname{arccctg}|x+1|, & -15,4 \leq x \leq 1; \\ 1, & x < -15,4. \end{cases}$$

$$15. y = \begin{cases} (2)^{3x-1} x^2, & |x| \geq 5; \\ \ln|x|^{-1}, & 0 < |x| < 1; \\ \cos|x-1|, & 1 < |x| < 5. \end{cases}$$

$$16. y = \begin{cases} \arcsin(-x^2 + 1), & x = 0, \quad x \leq 3; \\ \lg^2(2x) + 4,4, & 0 < x; \\ -e^x, & -3 < x < 0. \end{cases}$$

$$17. g = \begin{cases} \frac{\pi}{8} \sin^2\left(\frac{x-y}{3}\right), & 1 \leq |x|, \quad |y| \leq 1; \\ y^{-e}, & 1 < y, \quad x < 1; \\ e^{-x}, & |x| < 1, \quad |y| \leq 1; \\ 0,15, & y < -1, \quad x > 1. \end{cases}$$

$$18. y = \begin{cases} \sqrt{\sin^3(x-1)}, & -2 \leq x \leq 2; \\ e^{-x}, & x > 6; \\ 4,4 \lg^3|x|, & 6 > |x| > 2; \\ \ln x^2, & x < -6. \end{cases}$$

$$19. y = \begin{cases} \arctg(\pi x), & 0 < x < \pi; \\ \ln(x-3,18), & 2\pi \leq x; \\ \frac{1}{\sqrt{x-\pi}}, & \pi < x < 2\pi; \\ \pi, & x \leq 0. \end{cases}$$

$$20. y = \begin{cases} e^{-\pi} + x^{-e}, & 1 < x \leq e; \\ \ln^2(x-e), & e < x < 10^3; \\ \frac{x^2}{e}, & 0 \leq x \leq 1; \\ 2^x \operatorname{ctg}^3 x^2, & x < 0; \\ 0, & x \geq 10^3. \end{cases}$$

$$21. y = \begin{cases} \sqrt{x}, & x > 0; \\ 2 - x^2, & x \leq 0. \end{cases}$$

$$22. y = \begin{cases} e^{-x}, & 1 < x < 2; \\ x^e + 1, & 2 \leq x \leq 5; \\ 1, & x < 1, \quad x > 5. \end{cases}$$

$$23. y = \begin{cases} \sin^2 x, & x \leq -1; \\ \sqrt{-x}, & -1 < x < 0; \\ x - \lg x, & x > 1. \end{cases}$$

$$24. y = \begin{cases} \frac{\sin x}{x}, & x > 0; \\ 2x^2 + \ln|x|, & x < 0; \\ 0, & x = 0. \end{cases}$$

$$25. y = \begin{cases} 124 - e^x, & |x| < 1; \\ \operatorname{tg}(x-1), & 1 < x < 10; \\ 1, & x \leq -1, x \geq 10. \end{cases}$$



Національний університет  
водного господарства  
та природокористування

### Завдання 3.2

Розробити алгоритм та записати відповідну програму для поданих нижче завдань.

#### Вимоги до програм:

- обчислити і вивести на екран три числа, тут  $k$  - номер групи і  $l$  - номер варіанта вводяться з клавіатури;
- виконати завдання в кожному конкретному варіанті.

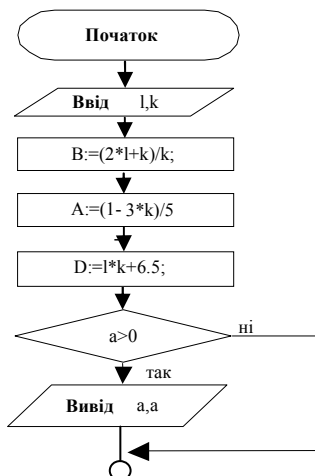
#### Приклад 3.2

$$a = \frac{1-3k}{5}, \quad b = \frac{2+k}{k}, \quad d = lk + 6,5.$$

Вибрати серед чисел додатні та надрукувати їх та їхні квадрати.

#### Розв'язання:

Запишемо приклад розв'язання даного завдання на мові програмування Pascal.



**Program** l3z2;

**Uses**

crt;

**Var**

a,b,d,l,k:real;

**Begin**

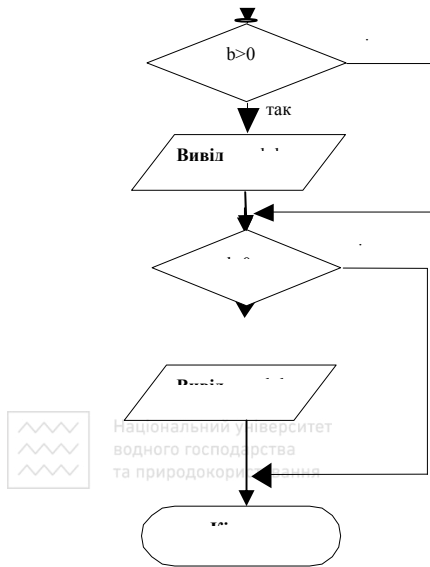
Clrscr;

Writeln('Введіть l,k');

Readln(l,k);

a:=(1-3\*k)/5;

b:=(2\*1+k)/k;



```

d:=l*k+6.5;
If a>0 then writeln('a>0   a=',a:6:2,'
a^2=',a*a:8:2); {Перевірка чи A>0}
If b>0 then writeln('b>0   b=',b:6:2,'
b^2=',b*b:8:2);
If d>0 then writeln('d>0   d=',d:6:2,'
d^2=',d*d:8:2);
End.

```

**Результат:** l=1; k=3; b>0; b=1,67; b<sup>2</sup>=2,78; d>0; d=9,5; d<sup>2</sup>=90,25.

### Завдання для самостійної роботи:

1.  $p = \frac{l-2k}{4}$ ,  $q = \frac{l+k}{k}$ ,  $r = lk - 12$ .

Вибрати серед цих чисел додатні, якщо вони є, та надрукувати їх потроєння.

2.  $t = \frac{2l+3k}{l}$ ,  $s = \frac{2l-3k}{9}$ ,  $\alpha = lk - 8,3$ .

Знайти і вивести на друк число  $x = \min(t, s, \alpha)$ .

3.  $a = \frac{2l+k}{l}$ ,  $b = lk - 9,8$ ,  $c = \frac{2l+3k}{2l}$ .

Вибрати серед них додатні, якщо вони є, та надрукувати їх та їхні квадратні корені.

4.  $x = \frac{2l-k}{7}$ ,  $y = \frac{l+k}{2l}$ ,  $z = lk - 9$ .

Знайти і вивести на друк число  $p = \max(x, y, z)$ .

5.  $a = \frac{l-k}{3}$ ,  $b = \frac{l+k}{l}$ ,  $c = lk - 7$ .

Вибрати серед чисел від'ємні, якщо вони є, та надрукувати їх та їхні кубічні корені.

6.  $x = \frac{l+k}{4}$ ,  $y = \frac{lk-14}{l+k}$ ,  $z = \frac{k-2l}{k}$ .

Обчислити та надрукувати число  $a = \frac{\max(x, y)}{\min^2(y, z) + 1}$ .

$$7. a = \frac{l-3k}{4}, \quad b = \frac{k+2l}{7k}, \quad c = 2lk - 5,6.$$

Вибрати ті з них, модулі яких більше 2 та надрукувати їх та їхні синуси.

$$8. x = \frac{l+2k}{k}, \quad y = \frac{k-2l}{k}, \quad z = lk - 12.$$

Обчислити та надрукувати число  $p = \frac{\min^2(x,y)}{\max^2(y,z) + 1}$ .

$$9. a = \frac{l-3k}{2}, \quad b = \frac{l+k}{l}, \quad c = \frac{2k-4l}{3kl}.$$

Вибрати ті з них, модулі яких більше 5 та надрукувати їх та їхні куби.

$$10. p = \frac{l+2k}{l}, \quad q = lk - 9,5, \quad s = \frac{3l-k}{2}.$$

Вибрати ті з них, які більше 1, та надрукувати їх та їхні квадратні корені.

$$11. a = \frac{l+5k}{l}, \quad b = \frac{l-3k}{8}, \quad c = 3lk - 15.$$

Вибрати ті з них, які по модулю більші 1 та надрукувати їхні куби.

$$12. a = \frac{l+k}{l}, \quad b = lk - 8, \quad c = \frac{l-3k}{2}.$$

Знайти і вивести на друк число  $x = \max(a,b,c) - \min(b,c)$ .

$$13. x = \frac{l+2k}{7}, \quad y = \frac{l-2k}{l}, \quad z = lk - 6,8.$$

Вибрати серед них від'ємні та надрукувати їх і їх подвоєння.

$$14. x = \frac{l+k}{5}, \quad b = \frac{l-k}{k}, \quad c = lk + 4,2.$$

Знайти число  $p = \min(a,b) + \max(b,c)$ .

$$15. a = \frac{2l-4k}{81}, \quad b = \frac{l+8k}{k}, \quad c = 2lk - 12.$$

Впорядкувати числа по зростанню.

$$16. x = \frac{l+2k}{l}, \quad y = \frac{2l-3k}{4}, \quad z = 3lk + 1.$$

Обчислити і надрукувати число  $q = \min(x,y,z) + 5$ .

$$17. a = \frac{l-2k}{l}, \quad b = \frac{l+k}{5}, \quad c = lk - 3,5.$$

Вибрати серед них ті, які належать відрізка  $[-5;8]$  та їх надрукувати.

$$18. \quad a = \frac{2l+k}{l}, \quad b = \frac{2l-3k}{5}, \quad c = lk + 2,4 .$$

Обчислити і надрукувати число  $z = \frac{\min(a,b)+1}{|\max(b,c)|+|c|}$ .

$$19. \quad a = \frac{l+k}{5}, \quad b = lk - 15, \quad c = \frac{k-2l}{l} .$$

Вибрати серед них від'ємні і надрукувати їх та квадратні корені з їх модулів.

$$20. \quad a = \frac{l+k}{4}, \quad b = \frac{l-k}{l}, \quad c = 2lk + 1 .$$

Впорядкувати їх по спаданню.

$$21. \quad a = \frac{l+4k}{5}, \quad b = \frac{4l-k}{l}, \quad c = lk - 11,3 .$$

Обчислити і надрукувати число  $r = \min(a,b) + \max(2 \cdot a, c)$ .

$$22. \quad a = \frac{2l-4k}{l}, \quad b = \frac{l+2k}{4}, \quad c = 3lk - 8,7 .$$

Обчислити і надрукувати число  $p = \sin\left(\frac{\min(a,b)}{\max^2(a,c) + 1}\right)$ .

$$23. \quad a = \frac{2l-3k}{5}, \quad b = \frac{2l+k}{l}, \quad c = 2lk - 8,5 .$$

Обчислити і надрукувати число  $q = \max(|a|, |b|, |c|)$ .

$$24. \quad x = \frac{l-3k}{2}, \quad y = \frac{l+k}{l}, \quad z = 2lk - 4 .$$

Обчислити і вивести на друк число  $p = \max(2 \cdot x, y) + \max(y, 2 \cdot z)$ .

$$25. \quad a = \frac{l+2k}{k}, \quad b = \frac{2l-k}{l+1}, \quad c = lk - 2 ,$$

Вибрати ті з них, що лежать поза проміжком  $[-1;5]$ , та надрукувати їх

### Завдання 3.3

Розробити алгоритм і програму на одній з мов програмування, щоб виявити належність точки  $M(x,y)$  геометричній фігури. Координати точки  $M$  та вигляд фігури наведені нижче.

#### Вимоги до програми

- вхідні дані ввести за допомогою стандартних процедур введення;
- на друк вивести значення вхідної інформації та результати роботи

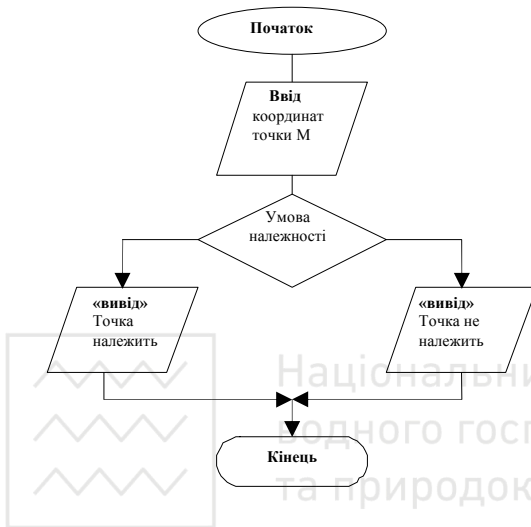
програми у вигляді повідомлення про належність або неналежність точки фігури.

### Приклад 3.3

$M(0;0.5)$ ;



### Розв'язання:



### Program l3z3;

*Uses crt;*

*Var X,y:real;*

**Begin**

**Clrscr;** «»

**Writeln(' введіть x,y: ');**

**Readln(x,y);**

**If (x>=0) and (y<=1)and (y<=x+1) and (y<=-x+1) then writeln('точка M('x, ';', y, ') належить фігурі')**

**else writeln('точка M('x, ';', y, ') не належить фігурі');**

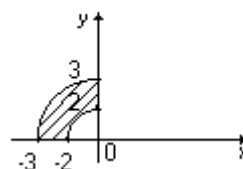
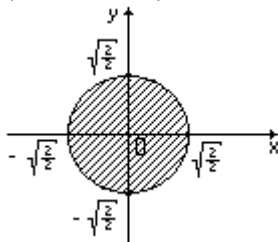
**End.**

**Результат:** Точка  $M(0; 0,5)$  належить фігурі.

### Завдання для самостійної роботи:

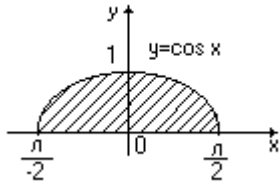
1.  $M(\pi/8; 2/6\pi)$ ;

2.  $M(-2.5;0.3)$ ;

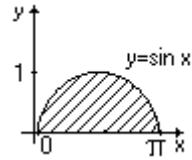


3.  $M(-0.3;0.8)$ ;

4.  $M(3\pi/4; \pi/12)$ ;



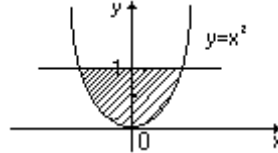
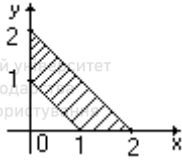
5.  $M(0.8; 1.4)$ ;



6.  $M(-0.3; 0.5)$ ;

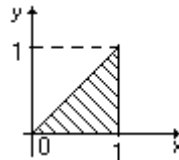


Національний університет  
водного господарства та природокористування



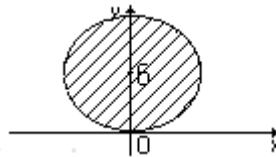
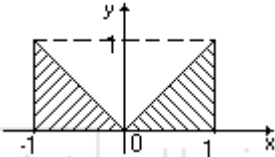
7.  $M(0.3; 0.6)$ ;

8.  $M(0.4; 0.5)$ ;



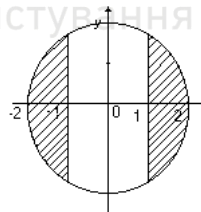
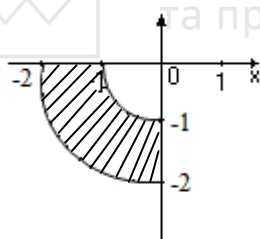
9.  $M(0.4; 0.7)$ ;

10.  $M(5.3; 6.4)$ ;



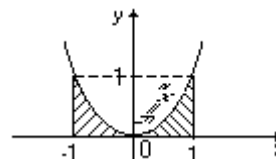
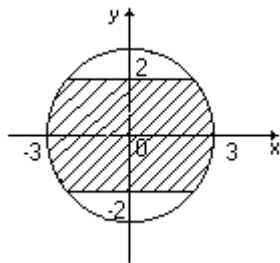
11.  $M(\sqrt{\frac{\pi}{4}})$ ;

12.  $M(\sqrt{\frac{\pi}{2}}, \sqrt{\frac{3\pi}{2}})$ ;



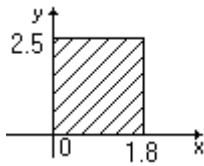
13.  $M(1.8; \sqrt{2}, 1)$ ;

14.  $M(0.4; \frac{\sqrt{2}}{2})$ ;

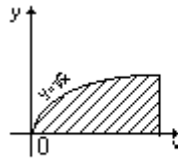




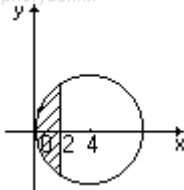
15.  $M(\frac{2\pi}{3}; \sqrt{5});$



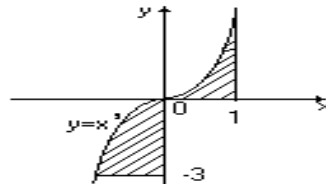
16.  $M(\sqrt{2}; \sqrt{3});$



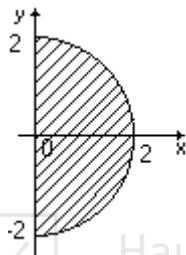
17.  $M(1.8; 1.7);$



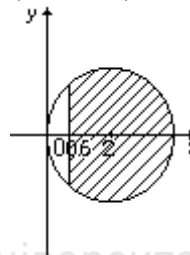
18.  $M(0.3; 0.4);$



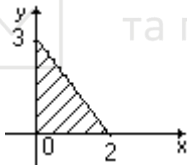
19.  $M(\sqrt{\pi}; \sqrt{3});$



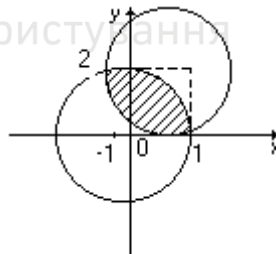
20.  $M(1.7; \pi/2);$



21.  $M(\sqrt{3}; \sqrt{5});$



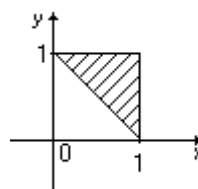
22.  $M(1; 2);$



23.  $\Phi=[a1;b1] \cup [a2;b2];$   
 $M((l-8*k); 2);$

де  $a_1 \leq b_1, a_2 \leq b_2$  – довільні числа, що вводяться з клавіатури.

24.  $M((\frac{\sqrt{2}-0,1}{2}, \sqrt{\pi});$



25. Множина  $\Phi$  складається з дискретних цілочисельних точок декартового добутку відрізків  $[0.5;8]$ ,  $[2;7]$ ;  $M([\pi], [\sqrt{5}])$ .



Національний університет  
водного господарства  
та природокористування

### Завдання 3.4

Для даного завдання розробити алгоритм та написати відповідну програму з виведенням результатів на екран.

#### Вимоги до програми:

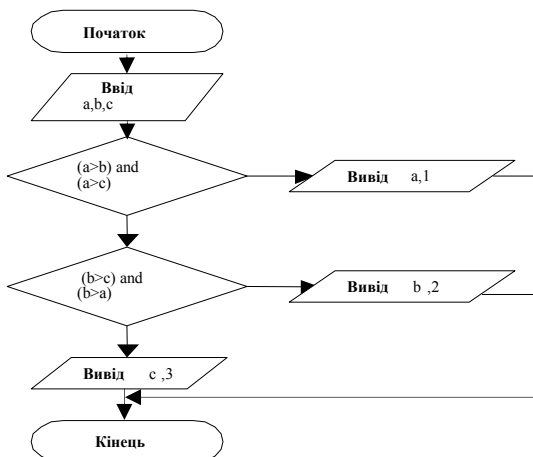
- вхідні дані ввести за допомогою стандартних процедур введення;
- на друк вивести значення вхідної інформації та результати обчислень;
- вхідні дані взяти довільними, в межах допустимих, якщо вони явно не задані в умові задачі.

#### Приклад 3.4

Прибуток трьох підприємств становить відповідно a,b,c (грошових одиниць). Визначити найбільший прибуток підприємств та вказати підприємство, яке його отримало.

#### Розв'язання:

Запишемо приклад розв'язання даного завдання на мові програмування Pascal.



**Program 13z4;**

**Uses**

Crt;

**Var**

a,b,c,max:real; N:byte;

**Begin**

Clrscr;

Writeln('Введіть

прибутки

підприємств a,b,c');

Readln(a,b,c);

**If** (a>b) **and** (a>c) **then** {умова, коли прибуток a буде  
максимальним}

**Begin**

Max:=a;

N:=1;

**End;**

**If** (b>a) **and** (b>c) **then**

**Begin**

Max:=b;

N:=2;

**End;**

**If** (c>b) **and** (c>a) **then**

**Begin**

Max:=c;

N:=3;

**End;**

**Writeln**('Підприємство ',n,' отримало максимальний прибуток '  
,max:9:1,' одиниць');

**End.**

**Результат:** a=23; b=3; c=18;

Максимальний прибуток 23 отримало підприємство 1.

### **Завдання для самостійної роботи:**

1. Прибуток трьох підприємств становить відповідно **a**, **b**, **c** (грошових одиниць). Визначити найбільший прибуток підприємств та вказати підприємство, яке його отримало.
2. Прибуток трьох підприємств становить відповідно **a**, **b**, **c** (грошових одиниць). Визначити підприємство, яке отримало найменший прибуток, та значення цього прибутку.
3. Прибуток трьох підприємств становить відповідно **a**, **b**, **c** (грошових одиниць). Визначити наскільки найбільший прибуток відрізняється від найменшого з вказівкою підприємств, які їх отримали.
4. Прибуток трьох підприємств становить відповідно **a**, **b**, **c** (грошових одиниць). Визначити, наскільки найбільший прибуток (з вказівкою підприємства, яке його отримало) відрізняється від середнього.
5. Прибуток трьох підприємств становить відповідно **a**, **b**, **c** (грошових одиниць). Визначити, наскільки середній прибуток підприємств відрізняється від найменшого.
6. Прибуток трьох підприємств становить відповідно **a**, **b**, **c** (грошових одиниць). Визначити, наскільки середній прибуток підприємств відрізняється від найбільшого.

7. Прибуток трьох підприємств становить відповідно  $a, b, c$  (грошових одиниць). Визначити суму двох більших прибутків підприємств.
8. Прибуток трьох підприємств становить відповідно  $a, b, c$  (грошових одиниць). Визначити суму двох менших прибутків підприємств.
9. Відомі середні заробітні плати трьох працівників  $x, y, z$ . З'ясувати, чиї зарплати не перевищують середню по підприємству, яка становить  $t$  (грошових одиниць).
10. Відомі середні заробітні плати трьох працівників  $x, y, z$ . З'ясувати, чиї зарплати перевищують середню по підприємству, яка становить  $t$  (грошових одиниць).
11. Відомі середні заробітні плати трьох працівників  $x, y, z$ . З'ясувати, чиї зарплати перевищують середню по підприємству, яка становить  $t$  (грошових одиниць) не більше, ніж на 100.
12. Відомі середні заробітні плати трьох працівників  $x, y, z$ . Розташувати дані суми в порядку зростання.
13. Відомі середні заробітні плати трьох працівників  $x, y, z$ . З'ясувати, чиї зарплати менші за  $q1$  або більші  $q2$  ( $q1 < q2$ ) (грошових одиниць).
14. Серед натуральних чисел, менших деякого  $n$ , обчислити кількість чисел, що є квадратами натуральних чисел.
15. Серед натуральних чисел, менших деякого  $n$ , обчислити добуток чисел, що діляться на 19 і не діляться на 4.
16. Відомі середні заробітні плати трьох працівників  $x, y, z$ . З'ясувати, чиї зарплати більші за  $q1$  і менші  $q2$  ( $q1 < q2$ ) (грошових одиниць).
17. Скласти алгоритм і програму розрахунку заробітної плати, якщо  $Z$  - сума до видачі,  $T$  - погодинний тариф,  $G$  - кількість відпрацьованих годин,  $A$  - сума отриманого авансу.
18. Фірма надає знижку оптовим покупцям. Сума знижки становить 10% від вартості проданого товару для кількості проданого товару, починаючи від 100 шт. Визначити відпускну ціну, якщо початкове значення ціни та кількість проданого товару вводяться з клавіатури.
19. Скласти алгоритм і програму розрахунку прибуткового податку, який визначається за формулою

$$P = \begin{cases} 0, & Z \leq 17; \\ (Z - 17) \cdot 0.1, & 17 \leq Z < 85; \\ 6.8 + (Z - 85) \cdot 0.15, & 85.01 \leq Z < 170; \\ 19.55 + (Z - 170) \cdot 0.2, & 170.01 \leq Z < 1020; \\ 189.95 + (Z - 1020) \cdot 0.3, & 1020.01 \leq Z < 1700; \\ 393.55 + (Z - 1700) \cdot 0.4, & 1700.01 \leq Z. \end{cases}$$

Тут  $P$  - прибутковий податок,  $Z$  - заробітна плата. Значення заробітної плати вводиться з клавіатури.

20. Сума нарахованих штрафів підприємствам за порушення фінансової дисципліни становить відповідно  $a, b, c, d$  (грошових одиниць).

Обчислити суму перших двох та двох останніх і визначити меншу серед них.

21. Сума нарахованих штрафів підприємствам за порушення фінансової дисципліни становить відповідно  $a, b, c, d$  (грошових одиниць). Обчислити суму перших двох та двох останніх і визначити більшу серед них.

22. Сума нарахованих штрафів підприємствам за порушення фінансової дисципліни становить відповідно  $a, b, c, d$  (грошових одиниць).

Обчислити середній розмір штрафу. Вивести ті штрафи, які є найменші та за середній.

23. Фірмі перерахували платежі в сумах  $w1, w2, w3$  (грошових одиниць). Необхідно впорядкувати їх за зростанням.

24. Фірмі перерахували платежі в сумах  $w1, w2, w3$  (грошових одиниць). Необхідно впорядкувати їх за спаданням.

25. Фірмі перерахували платежі в сумах  $w1, w2, w3$  (грошових одиниць). Необхідно вивести на екран найбільшу, а потім найменшу суму.

### Завдання 3.5

Для даного завдання розробити алгоритм та написати відповідну програму з виведенням результатів на екран.

#### *Вимоги до програми:*

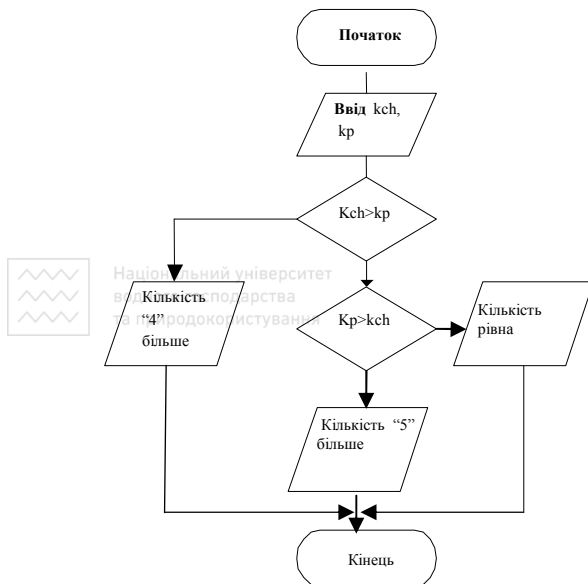
- вхідні дані ввести за допомогою стандартних процедур введення;
- на друк вивести значення вхідної інформації та результати обчислень;
- вхідні дані взяти довільними, в межах допустимих, якщо вони явно не задані в умові задачі.

#### Приклад 3.5

Скласти програму, яка визначала б: яких оцінок більше отримано під час іспиту з інформатики “4” чи “5”.

#### Розв’язання:

Запишемо приклад розв’язання даного завдання на мові програмування Pascal.



**Program 13z5;**

**Uses**

crt;

**Var**

Kp, Kch: integer;

**Begin**

**Clrscr;**

**Writeln**('Введіть                    кількість  
студентів,  
які отримали "4" та "5" на  
іспиті');

**Readln**(kch, kp);

**If** kch > kp **then writeln**("4"  
більше') {Умова, коли кількість 4  
більша}

**Else if** kp > kch **then writeln**("5"  
більше')

**Else writeln**('Порівню');

**End.**

**Результат:** kch=10; kp=5; "4" більше.

### **Завдання для самостійної роботи:**

1. Скласти програму, що визначала б: яких оцінок менше отримано студентами на іспиті з інформатики: "5" чи "4".
2. З клавіатури вводяться координати  $x$ ,  $y$  п'яти пар чисел. Скільки з них лежить в 2 координатній чверті?
3. Дано три числа  $x$ ,  $y$ ,  $z$ , які задають довжини сторін деякого трикутника. Перевірити, чи буде даний трикутник рівнобедреним.
4. Дано три числа  $x$ ,  $y$ ,  $z$ , які задають довжини сторін деякого трикутника. Перевірити, чи буде даний трикутник прямокутним.
5. Дано три числа  $x$ ,  $y$ ,  $z$ , які задають довжини сторін деякого трикутника. Визначити, яким буде кут, протилежний до сторони  $y$ : тупий, прямий, гострий?
6. З клавіатури вводяться координати  $(x, y)$  п'яти точок на площині. Скільки з них лежить в 1-й координатній чверті, а які в 3 координатній чверті?
7. Дано чотири числа  $a, b, c, d$ . Ці числа задають довжини сторін деякого чотирикутника. З'ясувати, чи буде даний чотирикутник паралелограмом.
8. Дано чотири числа  $a, b, c, d$ . Ці числа задають довжини сторін деякого

- чотирикутника. З'ясувати, чи буде даний чотирикутник ромбом.
9. Дано координати точок  $a, b, c, d$ . Визначити, чи буде даний чотирикутник прямокутником.
  10. З клавіатури вводяться координати  $(x, y)$  п'яти точок на площині. Скільки з них лежить в 3-тій координатній чверті та які їх координати?
  11. Скласти програму, що визначала б, яких оцінок більше отримано студентами на іспиті з інформатики: "5", "4" чи "3".
  12. Дано три числа  $x, y, z$ , які задають довжини сторін деякого трикутника. Перевірити, чи буде даний трикутник рівностороннім?
  13. З клавіатури вводяться координати  $(x, y)$  чотирьох точок на площині. Визначити максимальну площу серед всіх можливих комбінацій трикутників.
  14. З клавіатури вводяться координати  $(x, y)$  п'яти точок на площині та радіус кола з центром в початку координат. Визначити, які з точок належать відповідному колу.
  15. З клавіатури вводяться координати  $(x, y)$  п'яти точок на площині та радіус кола з центром в початку координат. Визначте, які з точок лежать в колі, які за межами кола, та підрахуйте їх кількість.

#### **Контрольні запитання:**

1. Запишіть оператор вибору та поясніть його виконання.
2. Зобразіть алгоритмічну структуру, що відповідає оператору Case.
3. Запишіть умовний оператор з його різновидами та поясніть його роботу.
4. Наведіть відомі вам способи введення даних в TP-програму, проілюструйте це на прикладі.
5. Яку алгоритмічну структуру реалізує вкладений оператор If?
6. Навести приклад програми з використанням вкладеного оператора If.
7. Чому не бажано використовувати в TP-програмах оператор безумовного переходу goto?

## Лабораторна робота № 4

### Тема: Циклічні алгоритми та програми

**Мета роботи:** Розглянути циклічні алгоритми і програми та реалізувати запропоновані програми на ПК за допомогою циклів з передумовою та післяумовою та оператора for.



Національний університет

водного господарства

та природоохоронного

### Хід роботи:

1. Розробити циклічні алгоритми згідно нижче наведених завдань.
2. Написати відповідні програми з використанням циклічних операторів на мові програмування (*Pascal, C*)
3. Визначені викладачем завдання виконати на комп'ютері.
4. Захистити лабораторну роботу.

### Завдання 4.1

Розробити алгоритм та записати програму для табулювання функції і обчислення значення функції від аргументу, що змінюється на вказаному проміжку із заданим кроком, а також для значень аргументу, починаючи із заданої точки.



Національний університет

водного господарства

та природоохоронного

### Вимоги до програми:

- вхідні дані (початкове, кінцеве значення аргументу та крок його зміни, кількість обчислюваних значень функції) вписати в розділ Const;
- вивести у вигляді таблиці значення аргументу та відповідне йому значення функції.

### Приклад 4.1

$$y = \frac{\cos^2 x}{x^2 + 1},$$

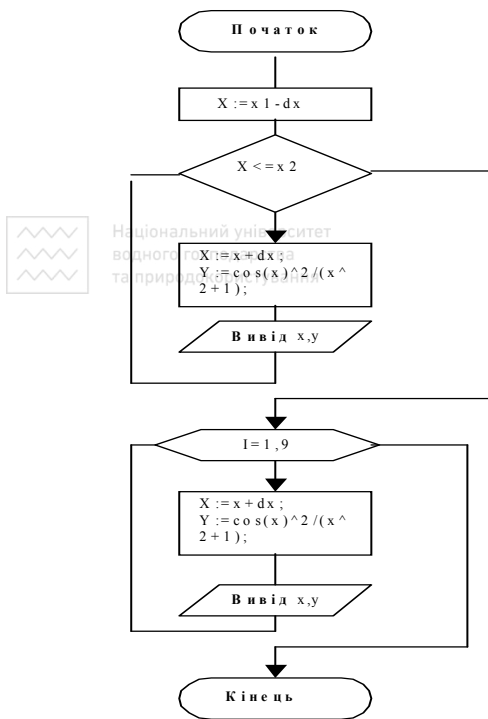
а)  $3,8 \leq x \leq 7,6$  ;  $\Delta x = 0,6$  ;

б)  $0,5 \leq x$  ,  $\Delta x = 0,1$  ,  $n = 9$  .

### Розв'язання:

Запишемо приклад розв'язання даного завдання на мові програмування Pascal.





**Program** l4z1;

**Uses**

Crt;

**Const**

x1=3.8;x2=7.6;dx1=0.6;dx2=0.1;  
x3=0.5;n=9;

**Var**

X,y:real;

I:integer;

**Begin**

Clrscr;

Writeln('табулювання а');

x:=x1-dx1;

**While** x<=x2 **do** {Цикл з передумовою}

**Begin**

X:=x+dx1; {Збільшення X на  
крок dx}

Y:=cos(x)\*cos(x)/(x\*x+1);

Writeln('x=',x:4:1,' y= ',y:6:1);

{Виведення значень функції}

**End;**

X:=x3-dx2;

Writeln('табулювання б');

**For** i:=1 to n **do**

**Begin**

X:=x+dx2;

Y:=cos(x)\*cos(x)/(x\*x+1);

Writeln('x=',x:4:1,' y= ',y:6:1);

**End;** **End.**

**Результати:**

x= 3.8 y=0.0;

x= 4.4 y=0.0;

x= 4.4 y=0.0;

x= 5.0 y= 0.0;

x= 5.6 y=0.0;

x= 6.2 y=0.0;

x= 6.8 y=0.0;

x= 7.4 y=0.0;

x= 0.5 y=0.6;

x= 0.6 y= 0.5;

x= 0.7 y= 0.4;

x= 0.8 y= 0.3;

x= 0.9 y= 0.2;

x= 0.9 y= 0.2;

x= 1.0 y= 0.1;

x= 1.1 y=0.1;

x= 1.2 y=0.1;

x= 1.3 y=0.0.

**Завдання для самостійної роботи:**

1.  $y = \frac{\cos^3 t^2}{1,5t + 2}$ ,

а)  $2,3 \leq t \leq 7,2$ ,  $\Delta t = 0,8$ ;

б)  $t \geq 0$ ,  $\Delta t = 0,3$ ,  $n = 5$ .

2.  $z = \frac{x^3 + 2x}{3\cos\sqrt{x} + 1}$ ,

а)  $0 \leq x \leq 2$ ,  $\Delta x = 0,4$ ;

б)  $x \geq 0,3$ ,  $\Delta x = 0,8$ ,  $n = 7$ .

3.  $z = \frac{t + \sin 2t}{t^2 - 3}$ ,

а)  $2,4 \leq t \leq 6,9$ ,  $\Delta t = 0,4$ ;

б)  $t \geq 3,1$ ,  $\Delta t = 0,8$ ,  $n = 6$ .

4.  $y = \frac{x^3 - 2}{3\ln x}$ ,

а)  $4,5 \leq x \leq 16,4$ ,  $\Delta x = 2,2$ ;

б)  $x \geq 2$ ,  $\Delta x = 1,5$ ,  $n = 5$ .

5.  $z = \frac{2,3t + 8}{|2\cos t| + 1}$ ,

а)  $0 \leq t \leq 6,5$ ,  $\Delta t = 1,1$ ;

б)  $t \geq 0,4$ ,  $\Delta t = 0,9$ ,  $n = 7$ .

6.  $y = \frac{\arccos x}{2x + 1}$ ,

а)  $0,1 \leq x \leq 0,9$ ,  $\Delta x = 0,1$ ;

б)  $x \geq 0$ ,  $\Delta x = 0,2$ ,  $n = 4$ .

7.  $y = \frac{5\operatorname{tg}(x + 7)}{(x + 3)^2}$ ,

а)  $1,2 \leq x \leq 6,3$ ,  $\Delta x = 0,2$ ;

б)  $x \geq 0,2$ ,  $\Delta x = 0,1$ ,  $n = 5$ .

8.  $y = \frac{1,5t - \ln 2t}{3t + 1}$ ,

а)  $2,5 \leq t \leq 9$ ,  $\Delta t = 0,8$ ;

б)  $t \geq 0,8$ ,  $\Delta t = 1,2$ ,  $n = 6$ .

9.  $y = \frac{2,5x^3}{e^{2x} + 2}$ ,

а)  $0 \leq x \leq 0,5$ ,  $\Delta x = 0,1$ ;

б)  $x \geq -0,8$ ,  $\Delta x = 0,25$ ,  $n = 6$

$$10. y = \frac{3x - 2}{2\arctg|x| + 1},$$

a)  $3,2 \leq x \leq 5,2$ ,  $\Delta x = 0,4$ ;

b)  $x \geq 2,5$ ,  $\Delta x = 0,6$ ,  $n = 5$ .

$$11. y = \frac{5 \lg x}{x^2 - 1}$$

a)  $1,2 \leq x \leq 3,8$ ,  $\Delta x = 0,4$ ;

b)  $x \geq 5,5$ ,  $\Delta x = 1,5$ ,  $n = 8$ .

$$12. z = \frac{6x + 4}{\sin 3x - x},$$

a)  $2,3 \leq x \leq 7,8$ ,  $\Delta x = 0,9$ ;

b)  $x \geq 2,8$ ,  $\Delta x = 0,3$ ,  $n = 6$ .

$$13. z = \frac{2 \sin^2(x + 2)}{x^2 + 1},$$

a)  $7,2 \leq x \leq 12$ ,  $\Delta x = 0,5$ ;

b)  $x \geq 0$ ,  $\Delta x = 0,1$ ,  $n = 5$ .

$$14. y = \frac{(3x + 2)^2}{\sin x + 3},$$

a)  $4,8 \leq x \leq 7,9$ ,  $\Delta x = 0,4$ ;

b)  $x \geq 0,2$ ,  $\Delta x = 0,7$ ,  $n = 6$ .

$$15. y = \frac{2 \sin^3 x}{3|x| + 1},$$

a)  $-1 \leq x \leq 1$ ,  $\Delta x = 0,25$ ;

b)  $x \geq -2,5$ ,  $\Delta x = 0,15$ ,  $n = 6$ .

$$16. y = \frac{\operatorname{tg} 2t - 3t}{t + 3},$$

a)  $0,2 \leq t \leq 0,8$ ,  $\Delta t = 0,1$ ;

b)  $t \geq -0,5$ ,  $\Delta t = 0,2$ ,  $n = 5$ .

$$17. y = \frac{3x + 1}{\arctg x},$$

a)  $0,1 \leq x \leq 1,5$ ,  $\Delta x = 0,2$ ;

b)  $x \geq 0,4$ ,  $\Delta x = 0,1$ ,  $n = 5$ .

$$18. y = \frac{2t + 8}{|\cos 3t| + 1},$$

a)  $2 \leq t \leq 6,5$ ,  $\Delta t = 0,8$ ;

b)  $t \geq 0,1$ ,  $\Delta t = 0,3$ ,  $n = 7$ .

19.  $y = \frac{\arccos x}{3x+1}$ ,  
 а)  $0,1 \leq x \leq 0,9, \Delta x = 0,1$ ;  
 б)  $x \geq 0,4, \Delta x = 0,05, n = 5$ .

20.  $y = \frac{(x+2)^2}{\sqrt{x^2+1}}$ ,

а)  $2,3 \leq x \leq 8,3, \Delta x = 0,6$ ;  
 б)  $x \geq 6,5, \Delta x = 0,3, n = 4$ .

21.  $y = \frac{t - \ln 2t}{3t+1}$ ,  
 а)  $2,1 \leq t \leq 8,5, \Delta t = 0,7$ ;  
 б)  $t \geq 0,6, \Delta t = 2,5, n = 5$ .

22.  $y = \frac{x^2+2x}{\cos 5x+2}$ ,  
 а)  $-2 \leq x \leq 4,5, \Delta x = 0,5$ ;  
 б)  $x \geq 0,6, \Delta x = 0,1, n = 5$ .

23.  $y = \frac{\ln|x+1|+5}{2x+3}$ ,  
 а)  $0,2 \leq x \leq 0,9, \Delta x = 0,15$ ;  
 б)  $x \geq 5, \Delta x = 0,4, n = 6$ .

24.  $y = \frac{x + \cos 2x}{3x}$ ,  
 а)  $2,7 \leq x \leq 8, \Delta x = 0,7$ ;  
 б)  $x \geq 0,8, \Delta x = 0,2, n = 6$ .

25.  $z = \frac{\arcsin 2x + |x|}{x^2+1}$ ,  
 а)  $0 \leq x \leq 0,4, \Delta x = 0,2$ ;  
 б)  $x \geq 0,1, \Delta x = 0,05, n = 6$ .

## Завдання 4.2

Розробити алгоритм та записати програму обчислення значення функції для аргументу, що змінюється із заданим кроком на вказаних проміжках.

### Вимоги до програми:

- межі області визначення функції і крок зміни аргументу задати в розділі const;
- вивести у вигляді таблиці пари чисел: аргумент, значення функції.

### Приклад 4.2

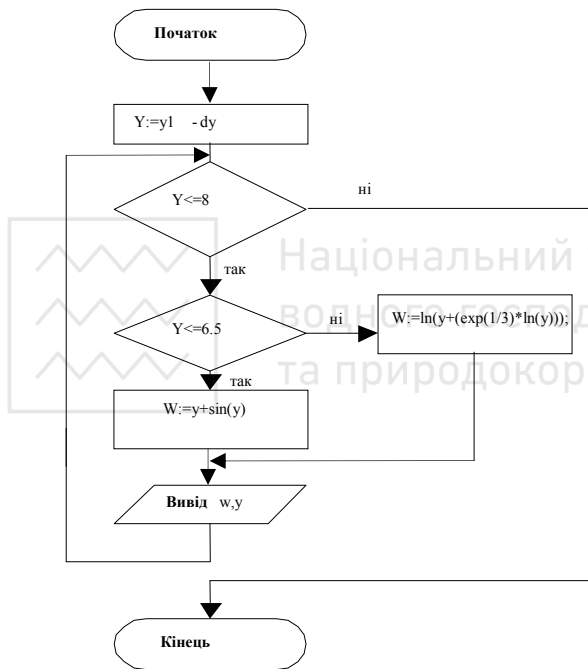


$$-6,5 < y < 0,5$$

$$0,5 \leq y \leq 8; \Delta y = 0,5;$$

### Розв'язання:

Запишемо приклад розв'язання даного завдання на мові програмування Pascal.



```

Program l4z2;
Uses crt;
Const y1=-6.5;y2=0.5;
        dy=0.5;y3=8;
Var w,y:real;
Begin
  Clrscr;
  y:=y1-dy;
  While y<=8 do
    Begin
      y:=y+dy;
      If (y<y2) then
        w:=y+sin(y);
        {Приналежності Y
        першому інтервалові
        значення W=y+sin y}
      If (y>=y2) then
        w:=ln(y+exp(1/3*ln(y)));
        Writeln('y=',y:5:1,'
        w=',w:5:1);
    End;
  End.
  
```

### Результати:

y= -6.5 w= 0.0;  
y= -6.0 w= -5.7;

y= 1.0 w= 0.7;  
y= 1.5 w= 1.0;

$$\begin{aligned}
y &= -5.5 & w &= -4.8; \\
y &= -5.0 & w &= -4.0; \\
y &= -4.5 & w &= -3.5; \\
y &= -4.0 & w &= -3.2; \\
y &= -3.5 & w &= -3.1; \\
y &= -3.0 & w &= -3.1; \\
y &= -2.5 & w &= -3.1; \\
y &= -2.0 & w &= -2.9; \\
y &= -1.5 & w &= -2.5; \\
y &= -1.0 & w &= -1.8; \\
y &= -0.5 & w &= -1.0; \\
y &= 0.0 & w &= 0.0; \\
y &= 0.5 & w &= 0.3;
\end{aligned}$$

$$\begin{aligned}
y &= 2.0 & w &= 1.2; \\
y &= 2.5 & w &= 1.3; \\
y &= 3.0 & w &= 1.5; \\
y &= 3.5 & w &= 1.6; \\
y &= 4.0 & w &= 1.7; \\
y &= 4.5 & w &= 1.8; \\
y &= 5.0 & w &= 1.9; \\
y &= 5.5 & w &= 2.0; \\
y &= 6.0 & w &= 2.1; \\
y &= 6.5 & w &= 2.1; \\
y &= 7.0 & w &= 2.2; \\
y &= 7.5 & w &= 2.2; \\
y &= 8.0 & w &= 2.3.
\end{aligned}$$

**Завдання для самостійної роботи:**

1. 
$$v = \begin{cases} t + \cos t, & 0 \leq t \leq 0,5; \\ \operatorname{arctg}(t + \ln t), & 0,5 < t \leq 2; \Delta t = 0,3. \end{cases}$$
2. 
$$y = \begin{cases} \operatorname{arccctg} x + e^x, & 0 \leq x \leq 0,5; \\ \ln(x + \sin x), & 0,5 < x \leq 8; \Delta x = 0,5. \end{cases}$$
3. 
$$w = \begin{cases} 0,3^v - v^2 + \cos v, & -3 \leq v \leq 1; \\ \operatorname{ctg}(0,34v - 0,2), & 1 < v \leq 7; \Delta v = 1. \end{cases}$$
4. 
$$z = \begin{cases} x^3 + \sin x, & 0 \leq x \leq 0,3; \\ \operatorname{arctg}(x + \ln x), & 0,3 < x \leq 2; \Delta x = 0,1. \end{cases}$$
5. 
$$w = \begin{cases} 0,6v - 0,3^v, & -2 \leq v \leq 0,3; \\ \ln(v + \sqrt{v + \cos v}), & 0,3 < v \leq 5; \Delta v = 0,5. \end{cases}$$
6. 
$$u = \begin{cases} x - 0,8 \sin x, & 0 \leq x \leq 2,2; \\ \operatorname{arctg}(\ln x + 0,3), & 2,2 < x \leq 3; \Delta x = 0,4. \end{cases}$$
7. 
$$v = \begin{cases} \cos z - z, & 0 \leq z \leq 0,5; \\ \ln(z + \sqrt{z}), & 0,5 < z \leq 7; \Delta z = 4; z \neq 4. \end{cases}$$
8. 
$$u = \begin{cases} 1,3t - \sin t, & -4 \leq t \leq 0; \\ \lg(t + \sqrt{t}), & 0 < t \leq 4; \Delta t = 0,5. \end{cases}$$
9. 
$$u = \begin{cases} 0,2t + \operatorname{arctg} t, & -2 \leq t \leq 0; \\ \operatorname{arcsin}(0,25t), & 0 < t \leq 5; \Delta t = 0,8. \end{cases}$$

$$10. \quad y = \begin{cases} \operatorname{arctg} z + z, & -2 \leq z \leq 0; \\ \lg z + \sqrt{z}, & 0 < z \leq 5; \Delta z = 0,5. \end{cases}$$

$$11. \quad r = \begin{cases} z + \cos z, & -1 < z \leq 0; \\ \operatorname{arctg}(z + \ln z), & 0 < z \leq 1; \Delta z = 0,4. \end{cases}$$

$$12. \quad w = \begin{cases} v^2 + \sqrt[3]{v}, & 0 \leq v \leq 0,5; \\ \ln(v + \sin v), & 0,5 < v \leq 8; \Delta v = 0,5. \end{cases}$$

$$13. \quad y = \begin{cases} x - e^x, & -2 \leq x < 2; \\ \operatorname{arctg}(x + \sqrt{x} - 1,4), & 2 \leq x \leq 5; \Delta = 0,5. \end{cases}$$

$$14. \quad t = \begin{cases} 1,3y + \sin y, & 0 \leq y \leq 0,3; \\ \operatorname{arctg}(y + \sqrt{y}), & 0,3 \leq y \leq 2; \Delta y = 0,3. \end{cases}$$

$$15. \quad x = \begin{cases} w + \cos w, & 0 \leq w < 0,5; \\ \operatorname{arctg} w - \lg(w + \sqrt{w}), & 0,5 \leq w \leq 2; \Delta w = 0,2. \end{cases}$$

$$16. \quad t = \begin{cases} x^2 - e^x, & 0 \leq x < 0,4; \\ \ln(\operatorname{arctg} x + x), & 0,4 \leq x \leq 2; \Delta x = 0,2. \end{cases}$$

$$17. \quad f = \begin{cases} t + e^x, & -0,5 \leq t \leq 0,5; \\ \sqrt{t} - \ln(t + \operatorname{arctg} t), & 0,5 < t \leq 4,5; \Delta t = 0,5. \end{cases}$$

$$18. \quad u = \begin{cases} \operatorname{arctg} v + e^v, & 0 \leq v < 1; \\ \lg(v + \cos v), & 1 \leq v \leq 3; \Delta v = 0,5. \end{cases}$$

$$19. \quad x = \begin{cases} \sin z - z^2, & 0 \leq z < 0,2; \\ \operatorname{arctg}(\ln z + \sqrt{z}), & 0,2 \leq z \leq 3; \Delta z = 0,2. \end{cases}$$

$$20. \quad f = \begin{cases} v^2 - \sqrt[3]{v}, & -2 \leq v < 0; \\ \operatorname{arctg}(v + \ln v), & 0 \leq v < 3; \Delta v = 0,5. \end{cases}$$

$$21. \quad z = \begin{cases} x - \ln x, & 1 \leq x < 2; \\ x^2 - 2, & 2 \leq x \leq 5; \Delta x = 0,2. \end{cases}$$

$$22. \quad y = \begin{cases} t^2 - \sin t, & -\pi \leq t < 0; \\ \sin t - t^2, & 0 \leq t \leq \pi; \Delta t = 0,1. \end{cases}$$

$$23. \quad f = \begin{cases} x|x^2 - 3|, & 0 \leq x < \sqrt{3}; \\ \sqrt{x^4}, & \sqrt{3} \leq x \leq 10; \Delta x = 0,5. \end{cases}$$

$$24. \quad y = \begin{cases} x^3 \cos x, & -\pi \leq x \leq 0; \\ x^2 \sin x^2, & 0 < x \leq \pi; \Delta x = 0,1. \end{cases}$$

$$25. \quad z = \begin{cases} t - 5t^2, & 5 \leq t < 7; \\ \sqrt{t} + t^2, & 7 \leq t \leq 10; \Delta t = 1. \end{cases}$$



### Завдання 4.3

Для даного завдання розробити алгоритм та написати відповідну програму з виведенням результатів на екран.

#### Вимоги до програми:

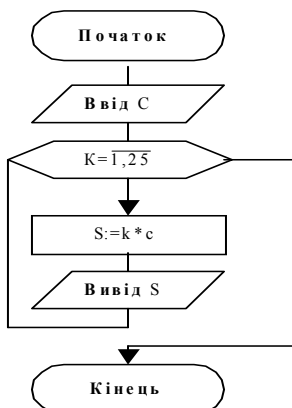
- вхідні дані ввести за допомогою стандартних процедур введення;
- на друк вивести значення вхідної інформації та результати обчислень;
- вхідні дані взяти довільними, в межах допустимих, якщо вони явно не задані в умові задачі.

### Приклад 4.3

Скласти таблицю для визначення суми виторгу залежно від кількості проданого товару. Ця залежність визначається формулою  $S=K \cdot C$ , де  $S$  – сума продажу,  $K$  – кількість товару,  $C$  – ціна одиниці товару (вводиться з клавіатури). В таблиці значення  $K$  змінюється від 1 кг до 25 кг з кроком 1 кг.

#### Розв'язання:

Запишемо приклад розв'язання даного завдання на мові програмування Pascal.



```

Program l4z3;
Uses crt;
Var s,c:real;
    k:integer;
Begin
    Clrscr;
    Writeln('Введіть ціну одиниці товару');
    Readln(c);
    Writeln('Кількість Ціна Сума ');
    For k:=1 to 25 do
    Begin
  
```



$s=k*c$ ; {Обчислення ціни для кожного конкретного  $C$ }

**Writeln**(k:8,c:8:2,s:10:2);

**End; End.**

**Результати:**

<b>k</b>	<b>c</b>	<b>s</b>	<b>k</b>	<b>c</b>	<b>s</b>
1	5,6	5,6	13	5,6	72,80
2	5,6	11,2	14	5,6	78,40
3	5,6	16,80	15	5,6	84,00
4	5,6	22,40	16	5,6	89,60
5	5,6	28,0	17	5,6	95,20
6	5,6	33,60	18	5,6	100,80
7	5,6	39,20	19	5,6	106,40
8	5,6	44,80	20	5,6	112,00
9	5,6	50,40	21	5,6	117,60
10	5,6	56,00	22	5,6	123,20
11	5,6	61,60	23	5,6	128,80
12	5,6	67,20	24	5,6	134,40
			25	5,6	140,00

**Завдання для самостійної роботи:**

1. Скласти таблицю для визначення суми виторгу залежно від кількості проданого товару. Ця залежність визначається формулою  $S = K * C$ , де  $S$  – сума виторгу,  $K$  – кількість товару,  $C$  – ціна одиниці товару (вводиться з клавіатури). В таблиці значення  $K$  змінюється від **1 кг** до **25 кг** з кроком **1 кг**.
2. Скласти таблицю для визначення суми виторгу залежно від кількості проданого товару. Ця залежність визначається формулою  $S = K * 2000$ , де  $K$  – кількість товару,  $S$  – сума продажу. В таблиці значення  $K$  змінюється від **10 кг** до **300 кг** з кроком **20 кг**.
3. Фірма надає знижку гуртовим покупцям. Сума знижки становить **10%** від вартості проданого товару для кількості проданого товару, починаючи від **100 шт.** Скласти таблицю визначення відпускної вартості залежно від кількості продажу – для **10 шт.** і більше (до **200 шт.**) з кроком **10 шт.**
4. Фірма надає знижку гуртовим покупцям. Сума знижки становить **5%** від вартості проданого товару для кількості проданого товару, починаючи від **20 кг.** Скласти таблицю визначення відпускної вартості залежно від кількості продажу - для **10 кг** і більше ( до **40 кг**) з кроком **3 кг.**
5. Постачальник повинен придбати товар на суму, яка не перевищує  $S$

грошових одиниць. Скласти програму розрахунку суми закупок, якщо постачальник робить закупки поступово, щоразу звіряючи сумарну вартість закупок із  $S$ . Ціна одиниці та кількість товару для кожної закупки вводиться з клавіатури, вартість закупки дорівнює добутку ціни на кількість товару.

6. Скласти алгоритм і програму визначення заробітної плати працівника згідно заданої формули для різних значень тарифів:  $Z=T*G$ , де  $Z$ - сума заробітної плати,  $T$  - погодинний тариф,  $G$ - кількість відпрацьованих годин ( $G=160; 10.6 \leq T \leq 20.6; \Delta T = 0.5$ ).



7. Скласти алгоритм і програму визначення заробітної плати працівника згідно заданої формули для різних значень тарифів:  $Z=T/8*G$ , де  $Z$  – сума заробітної плати,  $T$  – денний тариф,  $G$  – кількість відпрацьованих годин ( $G=186; 15.4 \leq T \leq 25.4; \Delta T = 0.5$ ).

8. Скласти алгоритм і програму визначення заробітної плати працівника згідно заданої формули для різних значень тарифів:  $Z=T*8*G$ , де  $Z$  – сума заробітної плати,  $T$  – погодинний тариф,  $G$  – кількість відпрацьованих днів ( $G=21; 9.0 \leq T \leq 29.0; \Delta T = 1.0$ ).

9. Скласти алгоритм і програму визначення заробітної плати  $N$  працівників згідно заданої формули:  $Z=T*8*G$ , де  $Z$  – сума заробітної плати,  $T$  – погодинний тариф,  $G$  – кількість відпрацьованих днів. Значення  $T$  і  $G$  вводяться з клавіатури.

10. Скласти алгоритм і програму визначення заробітної плати працівника за  $K$  місяців згідно заданої формули:  $Z=T/8*G'$ , де  $Z$  – сума заробітної плати,  $T$  – денний тариф,  $G$  – кількість відпрацьованих годин. Значення  $T$  і  $G$  вводяться з клавіатури для кожного місяця.

11. Скласти алгоритм і програму визначення сумарної заробітної плати  $N$  працівників, якщо заробітна плата кожного працівника розраховується згідно формули  $Z=T*G$ , де  $Z$  – сума заробітної плати,  $T$  – погодинний тариф,  $G$  – кількість відпрацьованих годин. Значення  $T$  і  $G$  вводяться з клавіатури для кожного працівника.

12. Скласти алгоритм і програму визначення середньої заробітної плати  $N$  працівників, якщо заробітна плата кожного працівника розраховується згідно формули  $Z=T*G/8$ , де  $Z$  – сума заробітної плати,  $T$  – денний тариф,  $G$  – кількість відпрацьованих годин. Значення  $T$  і  $G$  вводяться з клавіатури для кожного працівника.

13. Скласти алгоритм і програму визначення найменшої заробітної плати серед  $N$  працівників, якщо заробітна плата кожного працівника розраховується згідно формули  $Z=T*G/8$ , де  $Z$  – сума заробітної плати,  $T$  – денний тариф,  $G$  – кількість відпрацьованих годин. Значення  $T$  і  $G$  вводяться з клавіатури для кожного працівника.

14. Скласти алгоритм і програму визначення найбільшої заробітної плати серед  $N$  працівників, якщо заробітна плата кожного працівника

розраховується згідно формули  $Z=T*G$ , де  $Z$  - сума заробітної плати,  $T$  - погодинний тариф,  $G$  - кількість відпрацьованих годин. Значення  $T$  і  $G$  вводяться з клавіатури для кожного працівника.

15. Скласти алгоритм і програму визначення різниці між найбільшою та найменшою заробітною платою  $N$  працівників, якщо заробітна плата кожного працівника розраховується згідно формули  $Z=T*G$ , де  $Z$  - сума заробітної плати,  $T$  - погодинний тариф,  $G$  - кількість відпрацьованих годин. Значення  $T$  і  $G$  вводяться з клавіатури для кожного працівника.



16. Визначити секундну витрату води у населеному пункті  $Q_c = \frac{N * g}{24 * 3600}$ , де  $N$  - кількість жителів,  $g$  - середні витрати води на одного жителя за секунду (л/чол.)

$$(g = 90; 1000 \leq N \leq 10000; \Delta N = 1000).$$

17. Визначити максимальну годинну витрату води  $Q_{max} = a * \frac{S_{max}}{24}$ , де  $a$  - коефіцієнт годинної нерівномірності,  $S_{max}$  - максимальна добова витрата води ( $m^3$ ) ( $a = 1,5; 20000 \leq S_{max} \leq 30000; \Delta S_{max} = 1000$ ).

18. Господарство частину державного замовлення з молока виконує у вигляді вершків. У залік виконання замовлення іде кількість молока, яка вираховується за формулою:  $M = \frac{C * (Ж_B - Ж_O)}{(Ж_B - Ж_O)}$ . Тут  $M$  - кількість

молока, яка іде у залік (тонн);  $C$  - фактична кількість проданих вершків (тонн);  $Ж_в$  - процент жирності вершків;  $Ж_б$  - початкова (базисна) жирність молока до переробки (%);  $Ж_о$  - жирність молока після переробки (%). Визначити кількість молока, яка іде у залік, для різних значень кількості проданих вершків:

$$(Ж_B = 20.4; Ж_б = 5.8; Ж_о = 1.5; 10 \leq C \leq 60; \Delta C = 5)$$

19. Господарство частину державного замовлення з молока виконує у вигляді вершків. У залік виконання замовлення іде кількість молока, яка вираховується за формулою  $M = \frac{C * (Ж_B - Ж_O)}{(Ж_B - Ж_O)}$ . Тут  $M$  - кількість

молока, яка іде у залік (тонн);  $C$  - фактична кількість проданих вершків (тонн);  $Ж_в$  - процент жирності вершків;  $Ж_б$  - початкова (базисна) жирність молока до переробки (%);  $Ж_о$  - жирність молока після переробки (%). Визначити кількість молока, яка іде у залік, для різних значень кількості проданих вершків

$$(C = 50; Ж_B = 35.0; Ж_б = 5.8; 0.05 \leq Ж_о \leq 2.05; \Delta Ж = 0.2).$$

20. Ви придбали  $K$  акцій інвестиційної компанії вартістю  $V$  (грошових одиниць) кожна. Компанія щорічно сплачує на кожен акцію дивіденди

у розмірі  $P$  відсотків її вартості. Розрахувати суму дивідендів, яку Ви отримаєте за  $R$  років ( $R = 5; P = 23; K = 100; 15.2 \leq V \leq 45.2; \Delta V = 5$ ).

#### Завдання 4.4

Розробити алгоритм та записати програму обчислення значення аргументів функції на вказаних проміжках із заданими кроками, а також значень функції (табулювання функції двох змінних), починаючи із заданої точки.



Національний університет  
водного господарства  
та природокористування

#### Вимоги до програми:

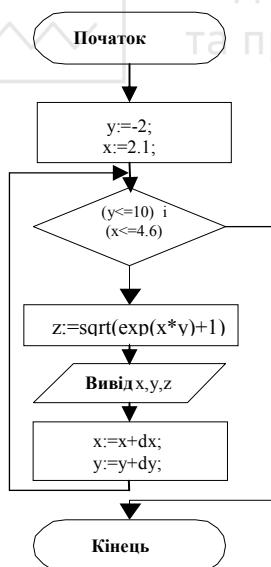
- вхідні дані (початкові, кінцеві значення аргументів та кроки їх зміни) ввести стандартною процедурою введення або в розділі Const;
- передбачити визначення області допустимих значень функції;
- вивести у вигляді таблиці значення аргументів та відповідне їм значення функції.

#### Приклад 4.4

$$z = \sqrt{e^{xy} + 1}; \quad -2 \leq y \leq 10, \Delta y=1, \quad 2.1 \leq x \leq 4.6, \Delta x=0.2;$$

#### Розв'язання:

Запишемо приклад розв'язання даного завдання на мові програмування Pascal.



```

Program l4z4;
Uses crt;
Const dy=1;dx=0.2;
Var z,x,y:real;
Begin
  Clrscr;
  y:=-2;
  x:=2.1;
  Writeln('x= y= z='); {Шапка таблиці}
  While (y<=10) and (x<=4.6)do
    Begin
      z:=sqrt(exp(x*y)+1);
      {Обчислення Z для функції 2-ох змінних}
      Writeln(x:6:2 ,y:6:2,z:6:2);
      {Наступне значення X, Y}
      y:=y+dy;
      x:=x+dx;
    End;
  Readln; End.
  
```

### Завдання для самостійної роботи:

1.  $z = \sqrt{e^{xy} + 1}$ ;  $-2 \leq y \leq 1$ ,  $\Delta y = 0.1$ ,  $2.1 \leq x \leq 4.6$ ,  $\Delta x = 0.2$ ;
2.  $z = \sqrt[3]{x^2 + 3 \cdot y}$ ;  $-2 \leq y \leq 10$ ,  $\Delta y = 1$ ,  $2 \leq x \leq 6$ ,  $\Delta x = 0.2$ ;
3.  $z = \cos^2 \frac{x+3}{y+1}$ ;  $2 \leq y \leq 10$ ,  $\Delta y = 1$ ,  $2.1 \leq x \leq 4.6$ ,  $\Delta x = 0.2$ ;
4.  $z = \ln(\cos(x \cdot y - 1) + 2)$ ;  $1 \leq x \leq 4$ ,  $\Delta x = 0.3$ ,  $0.5 \leq y \leq 1.5$ ,  $\Delta y = 0.2$ ;
5.  $z = \operatorname{tg}^2(x + y) + x^3$ ;  $1 \leq x \leq 4$ ,  $\Delta x = 0.3$ ,  $-0.5 \leq y \leq 1.5$ ,  $\Delta y = 0.2$ ;
6.  $z = 1,57 \cdot 10^3 \cdot x + \sqrt{y}$ ;  $1 \leq x \leq 4$ ,  $\Delta x = 0.3$ ,  $0.5 \leq y \leq 1.5$ ,  $\Delta y = 0.2$ ;
7.  $z = \operatorname{tg}(5 \cdot x) - \sin y$ ;  $0 \leq x \leq 2$ ,  $\Delta x = 0.2$ ,  $-1 \leq y \leq 1$ ,  $\Delta y = 0.1$ ;
8.  $z = \operatorname{arctg}(3 \cdot x \cdot y) + e^2$ ;  $0 \leq x \leq 2$ ,  $\Delta x = 0.2$ ,  $-2 \leq y \leq 2$ ,  $\Delta y = 0.4$ ;
9.  $z = 5^x + \sqrt[3]{y^2 + 7}$ ;  $0 \leq x \leq 2$ ,  $\Delta x = 0.2$ ,  $-2 \leq y \leq 2$ ,  $\Delta y = 0.4$ ;
10.  $z = \sqrt[5]{\cos^2 3 \cdot r + 2 \cdot \varphi^2}$ ;  $-4 \leq \varphi \leq 4$ ,  $\Delta \varphi = 1.5$ ,  $0 \leq r \leq 6.5$ ,  $\Delta r = 1.1$ ;
11.  $z = \log_3 |r^2 - 6,57 - \varphi|$ ;  $-4 \leq \varphi \leq 4$ ,  $\Delta \varphi = 1.5$ ,  $0 \leq r \leq 6.5$ ,  $\Delta r = 1.1$ ;
12.  $z = \sin(r + \varphi) - 0,42 \cdot \varphi$ ;  $-4 \leq \varphi \leq 4$ ,  $\Delta \varphi = 1.5$ ,  $0 \leq r \leq 6.5$ ,  $\Delta r = 1.1$ ;
13.  $z = x^2 + 2,08 \cdot y$ ;  $-0.8 \leq x \leq 0$ ,  $\Delta x = 0.1$ ,  $0 \leq y \leq 1.5$ ,  $\Delta y = 0.2$ ;
14.  $z = e^{\cos(x)} + e^{\sin(y)}$ ;  $-0.8 \leq x \leq 0$ ,  $\Delta x = 0.1$ ,  $-1 \leq y \leq 1$ ,  $\Delta y = 0.1$ ;
15.  $z = \operatorname{arctg}(x \cdot y)$ ;  $-0.8 \leq x \leq 0$ ,  $\Delta x = 0.1$ ,  $-2 \leq y \leq 1.5$ ,  $\Delta y = 0.2$ ;
16.  $z = \log_2(x^2 + y^2)$ ;  $0 \leq x \leq 0.5$ ,  $\Delta x = 0.1$ ,  $2.3 \leq y \leq 5.4$ ,  $\Delta y = 0.6$ ;
17.  $z = \cos^2(2 \cdot x^2 + 4 \cdot y)$ ;  $0 \leq x \leq 0.5$ ,  $\Delta x = 0.1$ ,  $2.3 \leq y \leq 5.4$ ,  $\Delta y = 0.6$ ;
18.  $z = \sin \frac{x+y}{x^2+y}$ ;  $0 \leq x \leq 0.5$ ,  $\Delta x = 0.1$ ,  $2.3 \leq y \leq 5.4$ ,  $\Delta y = 0.6$ ;
19.  $z = (x^2 - y^2) \cdot \ln|x \cdot y|$ ;  $2 \leq x \leq 12$ ,  $\Delta x = 0.5$ ,  $1 \leq y \leq 10$ ,  $\Delta y = 9$ ;
20.  $z = 3 \cdot x^2 - 4 \cdot x \cdot y$ ;  $-2 \leq y \leq 10$ ,  $\Delta y = 1$ ,  $2.1 \leq x \leq 4.6$ ,  $\Delta x = 0.2$ ;
21.  $z = 9x^5 + 2y$ ;  $-0.7 \leq x \leq 0$ ,  $\Delta x = 0.05$ ,  $0 \leq y \leq 1.5$ ,  $\Delta y = 0.1$ ;
22.  $z = \operatorname{ctg}(8 \cdot x - 3) - 5 \cos(y)$ ;  $0 \leq x \leq 2$ ,  $\Delta x = 0.2$ ,  $-1 \leq y \leq 1$ ,  $\Delta y = 0.1$ ;
23.  $z = \lg(5x^3 - y)$ ;  $0 \leq x \leq 0.5$ ,  $\Delta x = 0.1$ ,  $2.3 \leq y \leq 5.4$ ,  $\Delta y = 0.6$ ;
24.  $z = \ln(\sin(x \cdot y - 1) + 2x - 1) + \cos(x)$ ;  $1 \leq x \leq 4$ ,  $\Delta x = 0.1$ ,  $0.5 \leq y \leq 3.5$ ,  $\Delta y = 0.2$ ;
25.  $z = \sqrt{e^{\cos(xy)} + 1} - \sin(y)$ ;  $-2 \leq y \leq 1$ ,  $\Delta y = 0.1$ ,  $2.1 \leq x \leq 4.6$ ,  $\Delta x = 0.2$ ;

### Контрольні запитання

1. Що таке цикл?
2. Що таке тіло циклу?

3. Які цикли реалізуються в TP?
4. Запишіть програмно оператора циклу з передумовою?
5. Запишіть програмно оператор циклу з післяумовою?
6. Як працює цикл з передумовою?
7. Як працює цикл з післяумовою?
8. Навести приклад програми з використанням вкладеного оператора If?
9. В яких випадках краще застосувати той чи інший цикл?



Національний університет  
водного господарства  
та природокористування

## Лабораторна робота №5

### Тема: Типові прийоми програмування

**Мета роботи:** Ознайомитися з основними типовими прийомами програмування – обчислення значень сум та добутків та навчитися їх застосовувати на практиці.

#### Хід роботи:

1. Розробити алгоритми для поданих нижче завдань обчислення сум та добутків.
2. Написати відповідні програми на мові програмування (Pascal, C)
3. Визначені викладачем завдання виконати на ПК.
4. Захистити лабораторну роботу.

#### Завдання 5.1

Розробити алгоритм та записати програму обчислення значення суми та добутку.

##### **Вимоги до програми:**

- вхідні дані ( початкове і кінцеве значення індексної змінної в сумі чи добутку ) ввести в розділі Const;
- вивести обчислені значення суми та добутку.

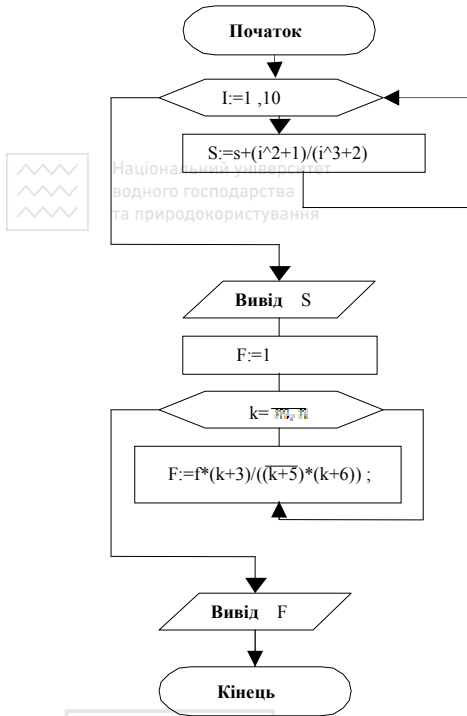
#### Приклад 5.1

$$y = \sum_{i=1}^{10} \frac{i^2 + 1}{i^3 + 2}$$

$$f = \prod_{k=m}^n \frac{k+3}{(k+5)(k+6)}$$

$m = 5, n = 13;$

**Розв'язання:**



```

Program l5z1;
Uses crt;
Const m=5;n=13;
Var i,k:integer;
    f,y:real;

Begin
  Clrscr;
  For i:=1 to 10 do
    y:=y+(i*i+1)/(i*i*i+2); {Обчислення суми}
  Writeln('y=',y:5:2);
  f:=1;
  For k:=m to n do
    f:=f*(k+3)/((k+5)*(k+6));
  Writeln('f=',f:8:1);
End.

```

**Результати:** y=2.63 f=0.03.

**Завдання для самостійної роботи:**

1.  $S = \sum_{i=4}^{16} \frac{i^3 - 2i + 3}{i + 4}, \quad y = \prod_{n=m}^l \frac{n^2 + 2n + 3}{n + 3},$

$m = 4, l = 12;$

2.  $y = \sum_{n=i}^k \frac{n - 2}{n^2 + 16}, \quad p = \prod_{k=2}^{21} (-1)^k \frac{k + 3}{k + 1}.$

$i = 2, k = 9;$

3.  $S = \sum_{i=10}^{30} \frac{i^3 + 3i^2 + 7}{3i + 8}, \quad z = \prod_{k=i}^j \frac{(k + 3)^2}{(k + 7)(k + 9)},$

$i = 5, j = 11;$

4.  $S = \sum_{k=3}^{10} (-1)^k \frac{(k - 1)(k - 2)}{k + 3}, \quad p = \prod_{n=l}^m \frac{(n + 3)^2}{(n - 5)^3}.$

$l = 6, m = 14;$

$$5. \quad y = \sum_{n=i}^k \frac{n}{n^2 + 5n + 6}, \quad p = \prod_{l=11}^{17} (-1)^l \frac{3l-4}{l^2+7},$$

$i=4, k=11;$

$$6. \quad S = \sum_{k=6}^{18} \frac{k^4 + 2}{2k^2 - 1}, \quad f = \prod_{n=r}^m \frac{n}{(n+2)(n+5)}.$$

$r=3, m=9;$

$$7. \quad y = \sum_{n=k}^i \frac{(n+3)^2}{(n+5) \cdot (n+7)}, \quad p = \prod_{l=8}^{17} (-1)^l \cdot \frac{l^2-2}{l+3},$$

$k=5, i=15;$

$$8. \quad s = \sum_{i=4}^{16} \frac{i^3 - 2i + 3}{i + 4}, \quad w = \prod_{k=n}^m \frac{k+1}{(k+5) \cdot (k+7)}.$$

$n=1, m=9;$

$$9. \quad y = \sum_{n=k}^j \frac{n+3}{(n+5) \cdot (n+6)}, \quad p = \prod_{l=1}^{14} \frac{l+5}{2l},$$

$k=1, j=6;$

$$10. \quad s = \sum_{k=5}^{16} (-1)^k \frac{4k-5}{2k+2}, \quad y = \prod_{k=n}^m \frac{k^2}{(k+3) \cdot (k+4)}.$$

$n=2, m=2;$

$$11. \quad s = \sum_{i=k}^n \frac{i^2 - i + 3}{i + 5}, \quad p = \prod_{s=5}^{11} (-1)^s \frac{s^2 + 3}{3s + 2},$$

$k=4, n=17;$

$$12. \quad y = \sum_{n=i}^k \frac{n^2 - n}{n^2 + n + 6}, \quad p = \prod_{l=3}^{10} e^{-l+2}.$$

$l=3, k=17;$

$$13. \quad s = \sum_{l=2}^{12} (-1)^l \frac{l^4 - 2}{l^2 + 3}, \quad y = \prod_{l=i}^k \frac{5l^2 - 2l + 1}{3l + 5},$$

$i=2, k=6;$

$$14. \quad s = \sum_{m=n}^l \frac{m^3 + 2m + 1}{m - 1}, \quad p = \prod_{k=3}^9 \frac{(k+2) \cdot (3k+1)}{k+6}.$$

$n=2, l=11;$

$$15. \quad s = \sum_{l=5}^{32} (-1)^{l-1} \frac{l^3 + 3}{l^2 + 3l + 7}, \quad y = \prod_{n=i}^l \frac{n}{(n+3) \cdot (n+8)},$$

$i=2, l=7;$



$$16. s = \sum_{i=k}^j \frac{i \cdot (i+1)}{(i+7) \cdot (i+3)}, \quad p = \prod_{n=1}^{10} (-1)^n \frac{n^2 + 2}{n}.$$

$$k = 3, \quad j = 14;$$

$$17. s = \sum_{n=1}^{30} (-1)^{n-1} \frac{n^2 - 2}{3n + 1}, \quad p = \prod_{i=k}^l \frac{(i+1)^2}{2i^3 + 3i + 1},$$

$$k = 2, \quad l = 9;$$

$$18. y = \sum_{i=k}^n \frac{i^3 + 1}{(i-2) \cdot (i+5)}, \quad p = 15!.$$

$$k = 3, \quad n = 9;$$

$$19. S = \sum_{m=2}^{19} (-1)^m \cdot \frac{(2m+3)(m+2)}{m^2 + 2}, \quad Z = \prod_{k=m}^l \frac{k+1}{(k+8)(k+6)},$$

$$m = 5, \quad l = 11;$$

$$20. z = \sum_{i=k}^n \frac{i^2 - 1}{2i^2 + 5i + 1}, \quad p = \prod_{i=2}^9 (-1)^i \cdot \frac{i+1}{i}.$$

$$k = 2, \quad n = 6;$$

$$21. S = \sum_{l=1}^9 (-1)^{l+2} \cdot \frac{l^3 + 3}{l+1}, \quad p = \prod_{n=m}^l \frac{n^2 + 2n + 3}{n+3},$$

$$m = 4, \quad l = 12;$$

$$22. y = \sum_{j=n}^i \frac{j+1}{(2j+4)(j+6)^2}, \quad p = \prod_{i=2}^9 \frac{(i+2)(i-3)}{2i+3}$$

$$n = 3, \quad i = 10;$$

$$23. S = \sum_{j=1}^9 (-1)^j \cdot \frac{2j}{j^2 + 3}, \quad V = \prod_{k=m}^n \frac{k+3}{k^2 + 10k + 24},$$

$$m = 3, \quad n = 7;$$

$$24. S = \sum_{i=k}^m \sin\left(\frac{\pi}{6} i\right), \quad f = \prod_{k=5}^{12} \frac{k+3}{(k+5)(k+6)}.$$

$$k = 1, \quad m = 9;$$

$$25. S = \sum_{l=3}^{13} (-1)^l \frac{l^2 + 1}{l+2}, \quad p = \prod_{i=k}^n \frac{i-1}{i^2 + 5},$$

$$k = 3, \quad n = 8.$$

## Завдання 5.2

Для даного завдання розробити алгоритм та написати відповідну

програму з виведенням результатів на екран.

### Вимоги до програми:

- вхідні дані ввести за допомогою стандартних процедур введення;
- на друк вивести значення вхідної інформації та результати обчислень;
- вхідні дані взяти довільними, в межах допустимих, якщо вони явно не задані в умові задачі.

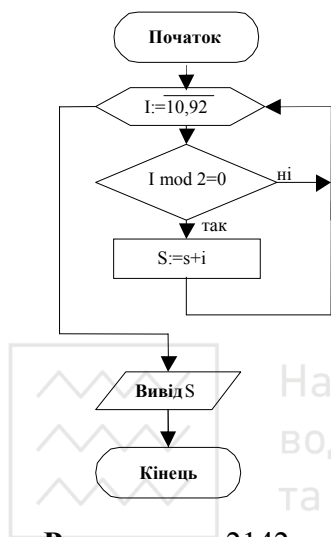


Національний університет  
водного господарства  
та природокористування

### Приклад 5.2

Знайти суму всіх парних чисел натурального ряду від 10 до 92.

### Розв'язання:



```
Program l5z2;
Uses crt;
Var s,i:integer;
```

**Begin**

```
Clrscr;
```

```
For i:=10 to 92 do
```

```
  If i mod 2=0 then s:=s+i; {Умова, що i— парне}
```

```
Writeln('s=',s:7);
```

```
End.
```

Результати: 2142.

### Завдання для самостійної роботи:

1. Знайти середнє математичне елементів натурального ряду від 1 до 100.
2. Обчислити середнє геометричне елементів натурального ряду від 1 до 10.
3. Підрахувати суму всіх чисел натурального ряду від 9 до 89, кратних 3.
4. Обчислити середнє математичне всіх непарних натуральних чисел від 1 до 50.
5. Знайти середнє геометричне всіх непарних чисел від 1 до 29.
6. Обчислити середнє математичне всіх парних чисел від 1 до 100.
7. Підрахувати середнє геометричне всіх парних чисел від 10 до 30.
8. Знайти суму чисел натурального ряду від 1 до 5, кратних 3.
9. Обчислити добуток чисел натурального ряду від 1 до 50, кратних 5.

10. Знайти різницю між добутками чисел натурального ряду від 1 до 10, що стоять на парних і непарних місцях.
11. Кожен член натурального ряду від 1 до 10 збільшити в два рази. Підрахувати суму.
12. Кожен член натурального ряду від 10 до 20 зменшити в десять разів. Підрахувати добуток.
13. Знайти суму цілих чисел від -5 до 15.
14. Аргументом функції  $y = \sin x$  служать числа натурального ряду від 1 до 10. Знайти суму значень цієї функції при  $n = 1 \dots 10$ .
15. Знайти добуток значень функції  $y = \cos n$ ,  $n=1..15$ .
16. Знайти суму цілочислових координат точок, що належать відрізку  $[0,5;11,9]$ .
17. Знайти добуток цілочислових координат відрізка  $[-7,5;-0,5]$ .
18. Обчислити середнє математичне цілочислових координат дискретних точок відрізка  $[-10,5;10,5]$ .
19. Обчислити  $n! = 1 \cdot 2 \cdot \dots \cdot n$  при  $n = 10$ .
20. Обчислити розміщення із  $n$  елементів по  $m$ :  
 $A_n^m = n(n-1) \cdot \dots \cdot (n-(m-1))$  при  $n = 10$ ,  $m = 4$ .
21. Підрахувати кількість елементів натурального ряду від 11 до 50, кратних 4.
22. Обчислити число комбінацій  $C_n^m$ , користуючись формулою  

$$C_n^m = \frac{n!}{m!(n-m)!}$$
23. Обчислити суму квадратів чисел натурального ряду від 1 до 10.
24. Визначити, чи є різниця сум чисел натурального ряду від 1 до 100, кратних 3 і 5 числом парним чи непарним. Залежно від результату вивести: ТАК або НІ.
25. Обчислити суму квадратів чисел натурального ряду чисел від 10 до 100 кратних 3.

### Завдання 5.3

Для даного завдання розробити алгоритм та написати відповідну програму для обчислення значення суми без використання масиву з виведенням результатів обчислення на екран.

#### Вимоги до програми:

- вхідні дані (початкове значення аргументу, крок зміни аргументу) ввести в розділі Const;

➤ вивести обчислене значення суми.

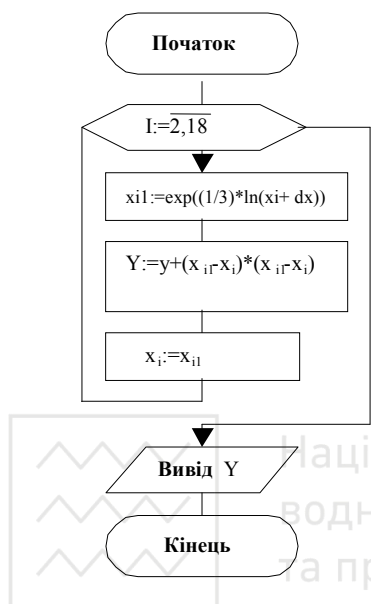
### Приклад 5.3

$$y = \sum_{i=2}^{18} (x_i - x_{i-1})^2, \text{ де } x_1 = 0,15; x_{i+1} = \sqrt[3]{x_i + \Delta x}; x = 0,05.$$



Национальний університет  
водного господарства  
та природозахисту

**Розв'язання:**



**Program l5z3;**

**Uses crt;**

**Const** x1=0.15;dx=0.05;

**Var** xi,xi1,y:real;

i:integer;

**Begin**

**Clrscr;**

xi:=x1;

**For** i:=2 to 18 **do**

**Begin**

xi1:=exp((1/3)\*ln(xi+dx));{Обчислення  
xi+1}

y:=y+(xi1-xi)\*(xi1-xi);

xi:=xi1; {Попереднє значення x  
присвоюється наступному}

**End;**

**Writeln**('y=',y:8:2);

**End.**

**Результати:** Y=0,28.

**Завдання для самостійної роботи:**

1.  $z = \sum_{j=2}^5 \frac{\ln|x_{j+1}|}{|x_j| + 8}, \text{ де } x_1 = -6,25; x_{j+1} = x_j + \Delta x; \Delta x = 1,15.$

2.  $y = \sum_{j=3}^{10} \frac{\sqrt{|x_{j+1}|} - \sqrt{|x_j|}}{2x_j + 6,25}, \text{ де } x_1 = -13,81; x_j = x_{j-1} + \Delta x; \Delta x = 1,05.$

3.  $x = \sum_{i=4}^{12} (y_i^2 - y_{i-1})^2, \text{ де } y_1 = -0,84; y_i = y_{i-1} + \Delta y.$

$$4. \quad p = \sum_{m=3}^{21} \left| \frac{x_m \cdot x_{m-1}}{x_m + x_{m-1}} \right|, \text{ де } x_1 = -1,18; x_m = 3x_{m-1} - 1,1\Delta x; \Delta x = 0,65.$$

$$5. \quad h = \sum_{i=3}^{16} \operatorname{tg}(y_{i+1}^2 - y_i), \text{ де } y_1 = -14,3; y_i = \frac{y_{i-1}}{3} + \Delta y; \Delta y = 2,6.$$

$$6. \quad y = \sum_{j=6}^{25} (x_{j+2} - x_j)^2, \text{ де } x_1 = -13,5; x_j = 2x_{j-1} + \Delta x; \Delta x = 1,25.$$

$$7. \quad y = \sum_{i=7}^{14} \frac{x_{i+2}^2}{x_i} + 1; \text{ де } x_1 = -12,45; x_i = x_{i-1} + \Delta x; \Delta x = 1,33.$$

$$8. \quad g = \sum_{j=5}^{16} \left| \frac{y_{j-1}^3}{y_{j+1}^2} + 2 \right|; \text{ де } y_1 = -6,44; y_j = 2y_{j-1} + \Delta y; \Delta y = 2,12.$$

$$9. \quad t = \sum_{i=8}^{22} \left| \frac{s_{i-1} \cdot s_{i+1}}{s_i + s_{i-1}} \right|; \text{ де } s_1 = -25,4; s_i = s_{i-1} + \Delta s; \Delta s = 3,3.$$

$$10. \quad y = \sum_{n=24}^{42} (x_n^2 - x_{n-2}^2); \text{ де } x_1 = 100; x_i = x_{i-1} + \Delta x; \Delta x = -2,2.$$

$$11. \quad h = \sum_{i=17}^{27} \frac{y_i^3 + 3}{7y_i + y_{i+1} + y_{i+2}}; \text{ де } y_1 = -100; y_i = y_{i-1} + \Delta y; \Delta y = 5.$$

$$12. \quad y = \sum_{j=13}^{24} \frac{2x_{j+2}}{x_j + x_{j+1}}; \text{ де } x_1 = -122,5; x_j = x_{j-1} + 12,5.$$

$$13. \quad g = \sum_{k=15}^{25} \frac{\operatorname{tg} x_k^2}{|x_{k-1} + x_{k+1}|}; \text{ де } x_1 = -8,25; x_k = x_{k-1} + \Delta x; \Delta x = 0,9.$$

$$14. \quad y = \sum_{i=10}^{19} (x_{i+1}^2 - x_{i-1}^2); \text{ де } x_1 = 0; x_i = \frac{1}{2}x_{i-1} + \Delta x; \Delta x = 0,35.$$

$$15. \quad y = \sum_{j=33}^{49} \frac{x_{i-2} + \sqrt[3]{x_{i-1}}}{x_{i+2} + \sqrt[3]{x_{i+1}}}; \text{ де } x_1 = -100,3; x_i = x_{i-1} + \Delta x; \Delta x = 3,1.$$

$$16. \quad z = \sum_{i=18}^{29} \left( \lg|x_{i+1}| - \lg|x_{i-1}| \right)^2; \text{ де } x_1 = -200,2; x_i = x_{i-1} + \Delta x; \Delta x = 4,4.$$

$$17. \quad g = \sum_{n=5}^{25} \frac{x_{n+2}^2 - x_n^2}{2x_{n+1} + 1}; \text{ де } x_1 = -5,5; x_n = \frac{1}{2}x_{n-1} + \Delta x; \Delta x = 0,4.$$

$$18. \quad y = \sum_{i=5}^{15} \frac{(x_i + x_{i-1})^3}{x_i - x_{i-1}}; \text{ де } x_i = 3,3; x_i = \frac{1}{3}x_{i-1} + \Delta x; \Delta x = 0,55.$$

$$19. y = \sum_{j=4}^{12} \frac{\lg x_{j-1}^2}{x_{j+1}}; \text{ де } x_1 = -3,3; x_j = x_{j-1} + \Delta x; \Delta x = 1,2.$$

$$20. y = \sum_{i=6}^{15} \frac{2x_{i+1} + x_i^2}{x_{i-1}}, \text{ де } x_1 = -12,1; x_i = x_{i-1} + \Delta x; \Delta x = 0,5.$$

$$21. y = \sum_{i=8}^{17} (x_{i-1}^2 + x_i + 2), \text{ де } x_1 = -3,5; x_i = 2x_{i-1} + \Delta x; \Delta x = 0,3.$$

$$22. y = \sum_{i=6}^{15} |x_{i-1}^3 - \ln x_i^2|, \text{ де } x_1 = 5,2; x_i = x_{i-1} + \Delta x; \Delta x = 0,5.$$

$$23. y = \sum_{i=6}^{11} \frac{x_i^2 + x_{i+1}^2}{x_{i-1} + x_i}, \text{ де } x_1 = 0,5; x_i = |x_{i-1}| + \Delta x; \Delta x = 0,1.$$

$$24. y = \sum_{i=8}^{18} (\sqrt{x_i} - \sqrt[3]{x_{i+1}}); x_1 = 1,2; x_i = \frac{1}{2}x_{i-1} + \Delta x; \Delta x = 0,2.$$

$$25. z = \sum_{i=15}^{28} \frac{\sin x_{i-1}}{\cos x_{i+1}}, \text{ де } x_1 = 1; x_i = x_{i-1} + \Delta x; \Delta x = 0,1.$$

#### Завдання 5.4

Скласти алгоритм і написати програму обчислення значення виразу (завдання з п. 5.3) з використанням всіх відомих операторів циклу.

#### Контрольні запитання:

1. Яким циклом краще користуватись при обчислення суми і добутку?
2. Яке початкове значення для циклу при обчисленні суми?
3. Яке початкове значення для циклу при обчисленні добутку?
4. Які циклічні оператори ви знаєте? Чим вони відрізняються?
5. У чому полягає специфіка оператора For?
6. Чи може приріст параметра циклу For бути відмінним від одиниці?
7. Які два варіанти циклу For ви знаєте?
8. Чи може параметр циклу For бути дійсного типу, символьного типу?
9. Як працює оператор циклу While?
10. Чи буде виконуватись цикл While, якщо записана у ньому умова одразу є хибною?

## Лабораторна робота №6

### Тема: Програмування ітераційних обчислювальних процесів

**Мета роботи:** ознайомитися з програмуванням ітераційних обчислювальних процесів, знаходженням границь, нескінченних сум .

#### Хід роботи:

1. Розробити алгоритми для поданих нижче завдань обчислення значення виразів з використанням ітераційних обчислювальних процесів.
2. Написати відповідні програми на мові програмування (*Pascal*)
3. Визначені викладачем завдання виконати на ПК.
4. Захистити лабораторну роботу.

#### Завдання 6.1

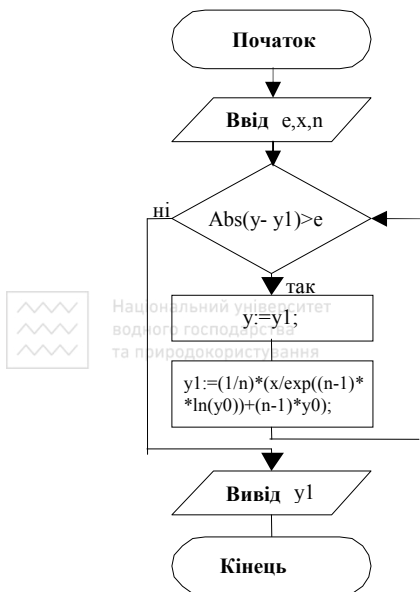
Розробити алгоритм та записати програму для обчислення значення кореня  $n$  – го степеня  $y = \sqrt[n]{x}$  із заданою точністю  $\varepsilon$ , користуючись ітераційною формулою:  $y_{i+1} = \frac{1}{n} \left( \frac{x}{y_i^{n-1}} + (n-1)y_i \right)$ ,  $y_0 = x$ ,  $i = 0, 1, \dots, n$

#### Вимоги до програми:

- вхідні дані (початкове значення аргументу, крок зміни аргументу) ввести стандартною процедурою введення;
- вивести обчислене значення кореня;
- обчислення по даній формулі завершити, якщо  $|y_{i+1} - y_i| \leq \varepsilon$ .

#### Приклад 6.1

```
Program l6z1;  
Uses crt;  
Var y,y1,e,x:real;  
      n:integer;  
Begin  
  Clrscr;  
  Writeln('введіть X'); Readln(x);  
  Writeln('введіть N'); Readln(n);  
  Writeln('введіть E'); Readln(e);  
  y1:=x;
```



**While**  $\text{abs}(y_1 - y) > \epsilon$  **do** {Обчислення завершується, коли модуль різниці між попередніми обчисленнями стане  $\leq \epsilon$ }

**Begin**

$y_0 := y_1;$

$y_1 := (1/n) * (x / \exp((n-1) * \ln(y_0))) + (n-1) * y_0;$

**End;**

**Writeln**('корінь рівний',  $y_1$ :5:2);

**End.**

**Результат:** 1,32.

### Завдання для самостійної роботи:

- |  |   |
|--|---|
| 1. $x = 0,66; n = 3; \epsilon = 0,001$ | 13. $x = 0,61; n = 6; \epsilon = 0,1$   |
| 2. $x = 9,3; n = 3; \epsilon = 0,001$  | 14. $x = 0,33; n = 9; \epsilon = 0,001$ |
| 3. $x = 8,9; n = 7; \epsilon = 0,1$    | 15. $x = 0,99; n = 2; \epsilon = 0,1$   |
| 4. $x = 3,2; n = 5; \epsilon = 0,1$    | 16. $x = 8,93; n = 3; \epsilon = 0,1$   |
| 5. $x = 2,9; n = 3; \epsilon = 0,01$   | 17. $x = 6,92; n = 2; \epsilon = 0,01$  |
| 6. $x = 9,3; n = 5; \epsilon = 0,1$    | 18. $x = 8,3; n = 5; \epsilon = 0,01$   |
| 7. $x = 11,6; n = 6; \epsilon = 0,1$   | 19. $x = 6,2; n = 6; \epsilon = 0,001$  |
| 8. $x = 6,2; n = 3; \epsilon = 0,001$  | 20. $x = 7,1; n = 7; \epsilon = 0,001$  |
| 9. $x = 8,03; n = 8; \epsilon = 0,1$   | 21. $x = 6,1; n = 7; \epsilon = 0,001$  |
| 10. $x = 7,9; n = 2; \epsilon = 0,01$  | 22. $x = 7,2; n = 3; \epsilon = 0,1$    |
| 11. $x = 6,9; n = 3; \epsilon = 0,001$ | 23. $x = 9,3; n = 2; \epsilon = 0,1$    |
| 12. $x = 0,21; n = 5; \epsilon = 0,01$ | 24. $x = 0,1; n = 3; \epsilon = 0,1$    |
|  | 25. $x = 0,01; n = 4; \epsilon = 0,01$  |

### Завдання 6.2

Скласти програму обчислення кореня нелінійного рівняння  $f(x) = 0$ , користуючись однією з ітераційних формул, якщо початкове наближення кореня  $x^{(0)}$  або проміжок ізоляції кореня  $[a, b]$  задані. Обчислення припинити, якщо різниця двох послідовних наближень до кореня не перевищує заданої точності  $\epsilon$ .

$$|x_{i+1} - x_i| < \epsilon.$$



### Вимоги до програми:

- вхідні дані (початкове значення кореня, точність обчислень) ввести в розділі;
- вивести обчислене значення кореня та кількість ітерацій;
- у кожному конкретному варіанті для уточнення кореня скористатись заданою ітераційною формулою.

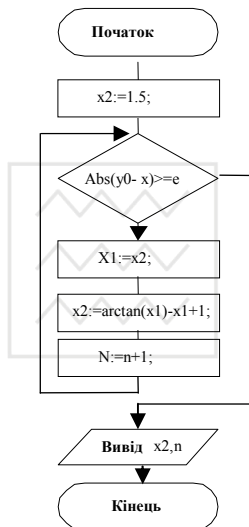


Національний університет  
водного господарства  
та природокористування

### Приклад 6.2

$$X^{(i+1)}=f(x^{(i)}), i=0,1,2,\dots$$
$$X=\arctg x+1, X^{(0)}=1.5$$

### Розв'язання:



**Program** l6z2;

**Uses** crt;

**Const** e=0.001;

x0=1.5;

**Var** x1,x2:real;

n:integer;

**Begin**

**Clrscr**;

x2:=x0;

**While** abs(x2-x1)>=e **do**

**Begin**

x1:=x2;

x2:=arctan(x1)-x1+1;

n:=n+1;{Обчислення кількості ітерацій}

**End**;

**Writeln**(n,' ітерацій');

**Writeln**(x2:8:2,' корінь');

**End.**

**Результат:** 9 ітерацій, 0.85 корінь.

### Завдання для самостійної роботи:

а) У поданих нижче варіантах 1-5 для уточнення кореня скористатись ітераційною формулою методу простих ітерацій

$$x^{(i+1)} = f(x^{(i)}), i = 0, 1, 2, \dots$$

1.  $x = \operatorname{tg} x + 4,$   $x^{(0)} = 3$
2.  $x = 2 - \ln x,$   $x^{(0)} = 2$
3.  $x = e^x - 20,$   $x^{(0)} = 3,15$
4.  $x = -\sqrt[3]{x} + 6,09,$   $x^{(0)} = 4,5$
5.  $x = S(100 - 10^x),$   $x^{(0)} = 2;$

б) у поданих нижче варіантах 6-10 для уточнення кореня скористатись ітераційною формулою Ньютона



Національний університет  
водного господарства  
та природокористування

$$x^{(i+1)} = x^{(i)} - f(x^{(i)}) / f'(x^{(i)}), i = 0, 1, 2, \dots$$

6.  $x^2 - \cos x = 0,$   $x^{(0)} = \pi / 2$
7.  $e^x + x = 0,$   $x^{(0)} = 0$
8.  $x^3 - x^3 + 3 = 0,$   $x^{(0)} = -2$
9.  $\ln x + x = 0,$   $x^{(0)} = 0,4$
- 10)  $x^3 - 3x - 1 = 0,$   $x^{(0)} = 2;$

в) у поданих нижче варіантах 11-15 для уточнення кореня скористатись модифікованою ітераційною формулою Ньютона

$$x^{(i+1)} = x^{(i)} - f(x^{(i)}) / f'(x^{(i)}), i = 0, 1, 2, \dots$$

11.  $x + 3^x + 1 = 0,$   $x^{(0)} = 1$
12.  $x^3 + x - 3 = 0,$   $x^{(0)} = 2$
13.  $e^x + 3 \sin x = 0,$   $x^{(0)} = 0$
14.  $e^{-x} - x = 0,$   $x^{(0)} = 0$
- 15)  $10^x + 2x - 5 = 0,$   $x^{(0)} = 1;$

г) у поданих нижче варіантах 16-20 для уточнення кореня скористатись ітераційною формулою методу половинного поділу

$$x^{(i)} = (a + b) / 2, i = 0, 1, 2, \dots;$$

$$\begin{aligned} \text{де } a = x^{(i)}, \text{ якщо } & f(a) \cdot f(x^{(i)}) \geq 0, \\ b = x^{(i)}, \text{ якщо } & f(a) \cdot f(x^{(i)}) < 0. \end{aligned}$$

16.  $2^x - 8x + 2 = 0,$   $[0; 0,25]$
17.  $e^{-x} + \ln x = 0,$   $[0,1; 1]$
18.  $\cos x - 2x = 0,$   $[0; 1,5]$
19.  $e^x - \ln x - 10 = 0,$   $[1; 3]$

20)  $x + 2^x = 0$ ,  $[-1; 0]$ ;

д) у поданих нижче варіантах 21-25 для уточнення кореня скористатись ітераційною формулою методу хорд (1-й варіант)

$$x^{(i+1)} = (a f(b) - b f(a)) / (f(b) - f(a)), i = 1, 2, \dots$$



де  $a = x^{(i)}$ , якщо  
 $b = x^{(i)}$ , якщо

$$f(a) \cdot f(x^{(i)}) > 0,$$

$$f(b) \cdot f(x^{(i)}) > 0;$$

- |                                 |              |
|---------------------------------|--------------|
| 21. $2x^3 + 4x - 1 = 0$ ,       | $[0; 1]$     |
| 22. $\cos x + 1/(x+2) = 0$ ,    | $[1; 2]$     |
| 23. $x^3 - 3x^2 + 1 = 0$ ,      | $[0; 1]$     |
| 24. $\cos(x-1) - 3x + 2 = 0$ ,  | $[0,9; 1,1]$ |
| 25. $x^3 + 6x^2 + 9x + 1 = 0$ , | $[-1; 0]$    |

### Завдання 6.3

Скласти програму обчислення значення функції, яка представляється у вигляді нескінченного ряду  $f(x) = \sum_{k=0}^{\infty} a_k(x)$ , із заданою

точністю  $\varepsilon = 0,01$  і числа  $k$  членів вказаної суми. Ітераційний процес завершити, якщо

$$|a_k(x)| \leq \varepsilon.$$

#### Вимоги до програми:

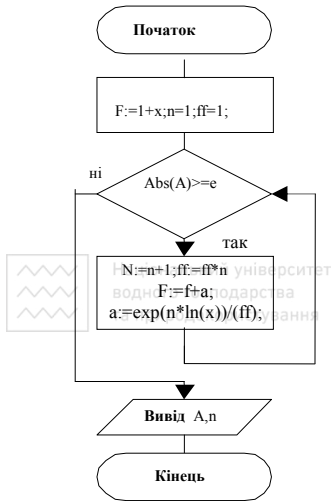
- вхідні дані (початкове значення змінної, точність обчислень) розділі Const;
- вивести обчислене значення функції та значення змінної, в якій вона обчислюється, а також кількість проведених ітерацій;
- у кожному конкретному варіанті для обчислення значення функції при проведенні проміжних обчислень члена суми по можливості скористатися рекурентною формулою.

#### Приклад 6.3

$$f(x) = l^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots; \quad x = 1,5$$

Розв'язання:

**Program** l6z3;  
**Uses** crt;

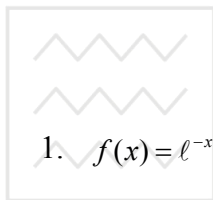


```

Const e=0.01;
  x=1.5;
Var f,a:real;
  n,ff:integer;
Begin
  Clrscr;
  f:=1+x;
  n:=1;
  ff:=1;
  a:=1;
  While abs(a)>=e do {Обчислення припиняється, коли
                    a<e}
  Begin
    n:=n+1;
    ff:=ff*n;
    a:=exp(n*ln(x))/(ff);
    f:=f+a;
  End;
  Writeln('n= ',n);
  Writeln('f=', f:8:2);
End.

```

**Результат:** n=7; x=4.48.



Національний університет водного господарства та природокористування

**Завдання для самостійної роботи:**

1.  $f(x) = e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots + (-1)^n \frac{x^n}{n!} + \dots; \quad x = 0,3$
2.  $f(x) = \cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + (-1)^{2n-1} \frac{x^{2n}}{(2n)!} + \dots; \quad x = \frac{\pi}{3}$
3.  $f(x) = \sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + (-1)^{n-1} \frac{x^{2n-1}}{(2n-1)!} + \dots; \quad x = \frac{\pi}{6}$
4.  $f(x) = \operatorname{Sh}(x) = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots + \frac{x^{2n-1}}{(2n-1)!} + \dots; \quad x = 0,1$
5.  $f(x) = \operatorname{Ch}(x) = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + \frac{x^{2n}}{(2n)!} + \dots; \quad x = 0,5$
6.  $f(x) = \frac{1}{1-x} = 1 + x + x^2 + x^3 + \dots + x^n + \dots; \quad x = 0,6$
7.  $f(x) = \frac{1}{1+x} = 1 - x + x^2 - x^3 + \dots + (-1)^{n-1} x^{n-1} + (-1)^n x^n + \dots; \quad x = 0,4$

$$8. f(x) = \ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots + (-1)^{n+1} \frac{x^n}{n} + \dots; \quad x=0,8$$

$$9. f(x) = \ln(1-x) = -x - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} - \dots - \frac{x^n}{n} - \dots; \quad x=0,3$$

$$10. f(x) = \sqrt{1+x} = 1 + \frac{1}{2}x - \frac{1}{2 \cdot 4}x^2 + \frac{1 \cdot 3}{2 \cdot 4 \cdot 6}x^3 - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6 \cdot 8}x^4 - \dots; \quad x=0,2$$

$$11. f(x) = \frac{1}{\sqrt{1-x^2}} = 1 + \frac{1}{2}x^2 + \frac{1 \cdot 3}{2 \cdot 4}x^4 + \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6}x^6 + \dots + \frac{1 \cdot 3 \cdot \dots \cdot (2n-1)}{2 \cdot 4 \cdot 6 \cdot \dots \cdot (2n)}x^{2n} + \dots; \quad x=0,3$$

$$12. f(x) = \frac{1}{\sqrt{1+x}} = 1 - \frac{1}{2}x + \frac{1 \cdot 3}{2 \cdot 4}x^2 - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6}x^4 + \dots + (-1)^{2n-1} \frac{1 \cdot 3 \cdot \dots \cdot (2n-1)}{2 \cdot 4 \cdot \dots \cdot (2n)}x^{2n}; \quad x=0,4$$

$$13. f(x) = e^{-x^2} = 1 - \frac{x^2}{1!} + \frac{x^4}{2!} - \frac{x^6}{3!} + \dots + (-1)^n \frac{x^{2n}}{n!} + \dots; \quad x=0,1$$

$$14. f(x) = \frac{\sin x}{x} = 1 - \frac{x^2}{3!} + \frac{x^4}{5!} - \frac{x^6}{7!} + \dots + (-1)^n \frac{x^{2n}}{(2n+1)!} + \dots; \quad x=0,2$$

$$15. f(x) = \frac{1+x}{1-x} = 1 + 2x + 2x^2 + 2x^3 + \dots + 2x^n + \dots; \quad x=0,4$$

$$16. f(x) = \frac{1}{1-x^2} = 1 + x^2 + x^4 + \dots; \quad x=0,7$$

$$17. f(x) = \arctg x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots + (-1)^n \frac{x^{2n+1}}{2n+1} + \dots; \quad x=0,3$$

$$18. f(x) = \frac{1}{2} \ln \frac{1+x}{1-x} = x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots + \frac{x^{2n+1}}{2n+1} + \dots$$

$$19. f(x) = \arcsin x = x + \frac{1}{2 \cdot 3}x^3 + \frac{1 \cdot 3}{2 \cdot 4 \cdot 5}x^5 + \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6 \cdot 7}x^7 + \dots + \frac{1 \cdot 3 \cdot 5 \cdot \dots \cdot (2n-1)}{2 \cdot 4 \cdot 6 \cdot \dots \cdot (2n)} * \frac{x^{2n+1}}{2n+1} + \dots$$

$$20. f(x) = 1 - \frac{x^2}{2} + \frac{x^4}{4} - \frac{x^6}{6} + \dots + (-1)^{2n-1} \frac{x^{2n}}{2n} + \dots; \quad x = \frac{\pi}{3}$$

$$21. f(x) = x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots + \frac{x^{2n-1}}{2n-1} + \dots; \quad x = \frac{\pi}{6}$$

$$22. f(x) = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots + (-1)^n \frac{1}{2n+1} + \dots;$$

$$23. f(x) = \frac{\sin(x)}{1} - \frac{\sin(2 \cdot x)}{2} + \frac{\sin(3 \cdot x)}{3} - \dots + (-1)^{n+1} \frac{\sin(n \cdot x)}{n \cdot x} + \dots; \quad x = \frac{\pi}{3}$$

$$24. f(x) = \frac{x^2}{2} + \frac{x^4}{4} + \frac{x^6}{6} + \dots + \frac{x^{2n}}{2 \cdot n} + \dots; \quad x=0,2$$

$$25. f(x) = \frac{1}{1 \cdot 2} - \frac{x}{2 \cdot 3} + \frac{x^2}{3 \cdot 4} - \frac{x^3}{4 \cdot 5} + \dots ; x = 0,15$$

### Завдання 6.4

Скласти програму обчислення границі функції  $\lim_{x \rightarrow a} f(x)$

із заданою точністю  $\varepsilon = 10^{-4}$ . Обчислення завершити і вважати, що функція  $f(x)$  має скінченну границю в заданій точці, якщо  $|f(x_k) - f(x_{k-1})| < \varepsilon$ . Вважати, що функція  $f(x)$  немає в заданій точці скінченної границі, якщо  $|f(x_k) - f(x_{k-1})| > M$ , де  $M=1000$ , або якщо не виконується умова  $|f(x_k) - f(x_{k-1})| < \varepsilon$  за задане число ітерацій  $N=100$ . В нижче наведених варіантах вважати  $a=const$ .

#### Вимоги до програми:

- вхідні дані (початкове значення змінної, точність обчислень) задати в розділі Const;
- вивести обчислене значення функції та значення змінної, в якій вона обчислюється, а також кількість проведених ітерацій;
- у кожному конкретному варіанті для прямування змінної  $x$  до заданої точки скористатися рекурентною формулою, сконструйованою самостійно.

#### Приклад 6.4

$$\lim_{x \rightarrow 0} \frac{x - \sin x}{x^3};$$

Розв'язання:

```

Program l6z4;
Uses crt;
Label m1;
Const m=1000;
        e=0.0001;
Var x,y:real;

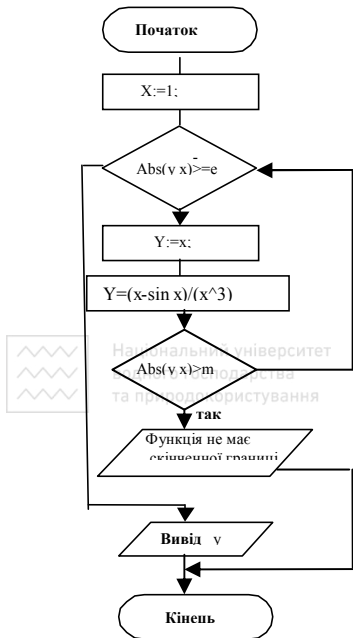
```

**Begin**

```

    Clrscr;
    y:=1;
    While abs(y-x)>=e do

```



```

Begin
  X:=y;
  Y:=(x-sin(x))/(x*x*x);      {Обчислення
  значення функції}
  If abs(y-x)>m then
    Begin
      Writeln('Функція не має скінченної
  границі');
    Goto m1;
    End;
  End;
  Writeln('lim=',y:8:4);
M1:End.

```

Результат: lim=0,1664;

**Завдання для самостійної роботи:**

1.  $\lim_{x \rightarrow a} \frac{\sqrt{x^3 - a^3}}{\sqrt{x - a}}$ ;

2.  $\lim_{x \rightarrow 0} \frac{\sqrt{a^2 + ax + x^2} - \sqrt{a^2 - ax + x^2}}{\sqrt{a+x} - \sqrt{a-x}}$ ;

3.  $\lim_{x \rightarrow 1} \frac{x^2 - x}{1 - x + \ln x}$ ;

4.  $\lim_{x \rightarrow 0} \frac{e^x - e^{-x} - 2x}{x - \sin x}$ ;

5.  $\lim_{x \rightarrow 0} \frac{\sin x - x * \cos x}{\sin^3 x}$ ;

6.  $\lim_{x \rightarrow 0} \frac{e^x - e^{\sin x}}{x - \sin x}$ ;

7.  $\lim_{x \rightarrow 0} \frac{1 - \cos x}{x - \sin x}$ ;

8.  $\lim_{x \rightarrow \infty} \frac{\frac{\pi}{2} - \arctg x}{\frac{1}{2} \ln \frac{x-1}{x+1}}$ ;

9.  $\lim_{x \rightarrow \frac{\pi}{2}} \frac{\tg 3x}{\tg 5x}$ ;

$$10. \lim_{x \rightarrow \infty} \frac{e^x}{x^2};$$

$$11. \lim_{x \rightarrow 0} \frac{x^{-1}}{\operatorname{ctgx}};$$

$$12. \lim_{x \rightarrow 1} \left( \frac{1}{\ln x} - \frac{1}{x-1} \right);$$

$$13. \lim_{x \rightarrow 1} \left( \frac{2}{x^2 - 1} - \frac{1}{x-1} \right);$$

$$14. \lim_{x \rightarrow \infty} \frac{2x^3 + 3}{\sqrt{x^6 - 3x^2 + 2}};$$

$$15. \lim_{x \rightarrow 0} (\operatorname{ctg} 3x * \sin x);$$

$$16. \lim_{n \rightarrow \infty} \frac{1}{\sqrt{n}} \cdot \sin(n+1);$$

$$17. \lim_{x \rightarrow 1} \frac{(2^x + 3)^2}{\sin\left(\frac{\pi}{2}\right)x};$$

$$18. \lim_{x \rightarrow \infty} \frac{x^2 + 1}{x^2 + 3x + 2};$$

$$19. \lim_{x \rightarrow 0} \frac{\sin^2(x/2)}{2x^2};$$

$$20. \lim_{x \rightarrow 0} (1 + 2x)^{1/x};$$

$$21. \lim_{n \rightarrow \infty} \left( \frac{n}{n+1} \right)^n;$$

$$22. \lim_{x \rightarrow 3} \frac{3x^2 - 1}{2x^3 + 6x^2 - 5};$$

$$23. \lim_{x \rightarrow 0} \left( \frac{2 + \cos x}{x^3 \cdot \sin x} - \frac{3}{x^4} \right);$$

$$24. \lim_{x \rightarrow 0} \left( \operatorname{ctg}^2 x - \frac{1}{x^2} \right);$$

$$25. \lim_{x \rightarrow 0} \left( x - x^2 \cdot \ln\left(1 + \frac{1}{x}\right) \right).$$



### Контрольні запитання:

1. Чим відрізняються циклічні оператори While та Repeat-until.
2. Скільки разів буде виконуватись оператор циклу Repeat-until, якщо умова після слова until істинна при входженні в цикл?
3. Що таке оператор безумовного переходу? Для чого він використовується?
4. Що таке мітка? Де потрібно її описувати?
5. Чому використання оператора Goto в програмі є небажаним?
6. Що таке розгалуження? За допомогою якого оператора, розгалуження здійснюється на мові Pascal?
7. Чим відрізняються повний та неповний умовний оператор?
8. Чи можна використовувати вкладені умовні оператори?
9. Що таке оператор вибору і як він працює?
10. Чи можуть мітки повторюватись у одному операторі Case?



Національний  
водного господарства  
та природоохорони

## Лабораторна робота №7

### Тема: Одновимірні масиви

**Мета роботи:** Ознайомитися з особливостями організації одновимірних масивів, навчитися виконувати обробку одновимірних масивів, здійснювати сортування масивів.

### Хід роботи:

1. Розробити алгоритми обробки одновимірної масиву згідно варіантів завдань для самостійної роботи.
2. Написати відповідні програми на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.

### Завдання 7.1

Скласти алгоритм і написати програму для обчислення суми та добутку з використанням масивів. Варіанти завдань взяти з п. 5.1.

#### Приклад 7.1

$$y = \sum_{i=1}^{10} \frac{i^2 + 1}{i^3 + 2}, \quad f = \prod_{k=m}^n \frac{k + 3}{(k + 5)(k + 6)},$$

$m = 5, \quad n = 13.$

## Розв'язання:

**Program** l7z1;

**Uses** crt;

**Var** i:integer;

{Опис масивів в розділі опису змінних}

**A:array**[1..10] of real;

**B:array**[5..13] of real;

**Y,f:real;**

**Begin**

**Clrscr;**

{Обчислення суми}

**Y:=0;**

**For** i:=1 **to** 10 **do**

**A[i]:=(i\*i+1)/(i\*i+2);**

**For** i:=1 **to** 10 **do**

**Y:=y+a[i];**

**Writeln**('Сума рівна ',y:8:2);

{Обчислення добутку}

**F:=1;**

**For** i:=5 **to** 13 **do**

**B[i]:=(i+3)/((i+5)\*(i+6));**

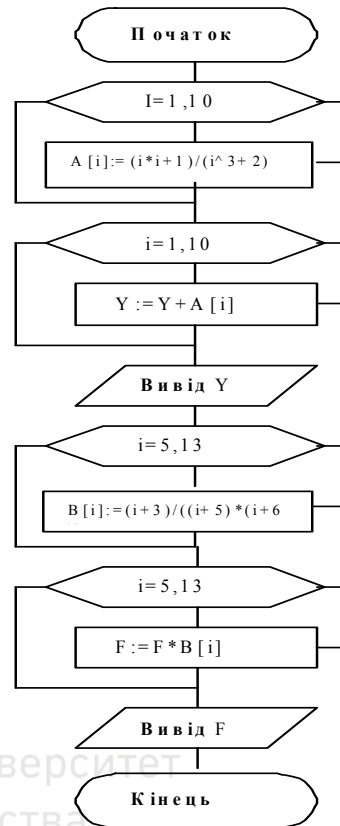
**For** i:=5 **to** 13 **do**

**F:=f\*b[i];**

**Writeln**('Добуток рівний ',f:8:8);

**End.**

**Результат:** Y=2,64; F=0,0.



## Завдання 7.2

Скласти алгоритм і написати програму для обробки одновимірного масиву згідно варіанту.

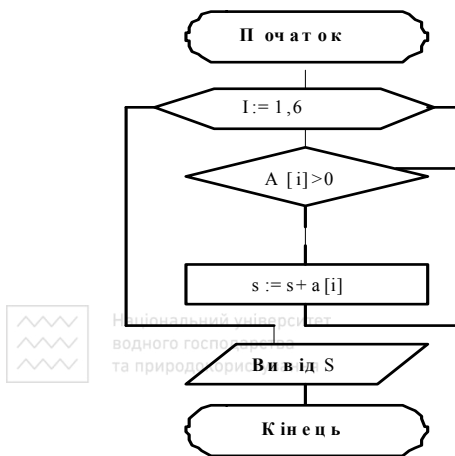
### Приклад 7.2

Знайти суму додатніх елементів мавиву  $B(6)=(5.0,-2.3,-6.9,-1.1,2.0,6.6)$ .

## Розв'язання:

**Program** l7z2;

**Uses** crt;



*{Запис масиву за допомогою типізованої константи}*

**Const**

**b:array [1..6] of real=(5.0,-2.3,-6.9,-1.1,2.0,6.6);**

**Var i:integer; s:real;**

**Begin**

**Clrscr;**

*{Обчислення добутку елементів масив}*

**s:=0;**

**For i:=1 to 6 do**

**If b[i]>0 then s:=s+b[i];**

**Writeln('Добуток додатніх елементів=',s:6:2);**

**End.**

**Результат:** s=13,6.

***Варіанти завдань для самостійної роботи:***

1. Знайти і надрукувати кількість додатних елементів масиву  $B(6) = (5.0; -2.3; -6.9; -1.1; 2.0; 6.6)$ .
2. Підрахувати і надрукувати кількість додатних елементів, які стоять на парних місцях  $C(8) = (-6.3; -1.0; 10.3; -8.8; 6.3; -1.1; 0.0; 0.1)$ .
3. Вивести на друк середнє арифметичне від'ємних елементів масиву:  $A(6) = (6.3; -2.1; 4.2; 5.3; -7.2; -4.5)$ .
4. Знайти мінімальний елемент масиву  $B(7) = (6.3; -1.6; 1.1; 0.1; -2.0; 2.3; 6.3)$ .
5. Надрукувати суму від'ємних елементів, які стоять на парних місцях в масиві  $X(12) = (-2.3; 4.0; -8.9; 6.3; 4.9; -7.8; -6.5; 5.1; 3.8; -4.3; -5.1; 7.2)$ .
6. Надрукувати середнє арифметичне невід'ємних елементів масиву, які стоять на непарних місцях  $B(10) = (6.3; 0.0; -8.3; 7.2; 6.1; -4.2; 5.7; 6.4; 5.6; -4.8)$ .
7. Знайти і надрукувати кількість додатних елементів масиву  $C(9) = (1.6; 2.1; -3.1; 0.0; 1.1; -2.2; 3.7; 8.9; 9.2)$ .
8. Обчислити добуток додатних елементів масиву  $D(5) = (1.1; -6.2; 0.0; 2.3; 5.1)$ .
9. Знайти суму елементів масиву  $X(6) = (3.5; -6.3; 2.1; 0.1; 5.1; -2.1)$ , значення яких менше 0,25.
10. Обчислити добуток модулів значень елементів масиву  $Y(7) = (-2.2; 0.2; 3.1; 2.1; -3.1; 6.1; 0.5)$ .
11. Визначити кількість елементів масиву  $B(5) = (2.2; 3.1; -3.6; 0.1; 2.1)$ , значення яких менше 0,99.
12. Визначити кількість від'ємних елементів масиву

$$D(5) = (1,2;25,3;-2,3;-3,1;0,0).$$

13. Обчислити добуток елементів масиву

$$B(6) = (2,3;4,3;-15,2;1,1;-1,2;-3,3), \text{ значення яких більше } 2,0.$$

14. Надрукувати порядкові номери від'ємних елементів масиву

$$Y(7) = (-7,9;1,0;1,1;-2,2;5,0;-1,1;2,0).$$

15. Підрахувати кількість елементів масиву

$$C(6) = (2,1;3,6;-6,3;4,1;2,2;-2,3), \text{ значення яких більше } 2,3.$$

16. Обчислити добуток елементів масиву  $A(5) = (3,1;-7,8;6,2;-3,3;1,1)$ , значення яких більші 5,4.

17. Обчислити суму значень від'ємних елементів

$$\text{масиву } X(8) = (-1,2;6,3;0,2;-0,7;1,1;2,3;-3,6;2,2).$$

18. Визначити номери додатних елементів масиву

$$C(7) = (1,1;2,3;-6,4;0,0;2,1;2,3;1,2).$$

19. Обчислити добуток елементів масиву

$$A(5) = (1,3;6,3;2,4;-3,6;-2,5).$$

20. Визначити мінімальний елемент масиву

$$X(6) = (2,1;-3,6;-2,0;0,0;-6,3;1,0) \text{ та номер цього елемента.}$$

21. Надрукувати номер першого від'ємного елемента масиву

$$A(8) = (3,2;-6,3;2,0;-3,3;-6,6;-2,2;0,2,1).$$

22. Визначити номер максимального елемента масиву

$$C(6) = (2,3;7,9;12,3;-6,8;-22,3;0,0).$$

23. Вивести номери від'ємних елементів масиву

$$D(7) = (2,2;-3,3;2,1;-3,0;-7,1;-5,1;0,0), \text{ які стоять на парних місцях.}$$

24. Знайти квадрат мінімального елемента масиву

$$B(6) = (2,1;3,3;0,5;-6,8;0,3;-1,2;5,3).$$

25. Визначити максимальний по модулю елемент масиву

$$C(8) = (-3,6;-5,3;2,1;0,1;-0,7;5,3;6,6;-2,2).$$

### Завдання 7.3

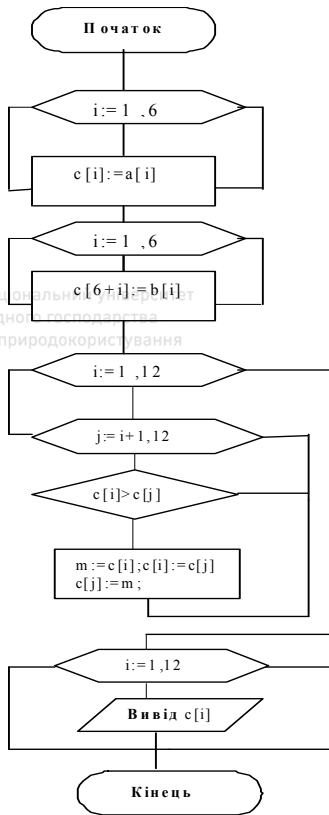
Маємо два масиви  $A(6) = (6,0; 4,3; 2,2; 1,3; 1,0; 0,1)$  та  $B(6) = (-5,3; 1,2; -0,1; -4,3; 1,0; 6,1)$ . Необхідно злити ці масиви в  $C(12)$ , розмістивши елементи в порядку зростання значень.

### Приклад 7.3

**Розв'язання:**

**Program l7z3;**

**Uses crt;**



**Const**

**a:array [1..6] of**

**real=(6.0,4.3,2.2,1.3,1.0,0.1);**

**b:array [1..6] of real=(5.3,1.2,-0.1,-4.3,1.0,6.1);**

**Var i,j:integer; c:array [1..12] of real;**

**m:real;**

**Begin**

**Clrscr;**

*{Злиття двох масивів}*

**For i:=1 to 6 do**

**c[i]:=a[i];**

**For i:=1 to 6 do**

**c[6+i]:=b[i];**

*{Сортування методом «бульбашки»}*

**For i:=1 to 12 do**

**For j:=i+1 to 12 do**

**If c[i]>c[j] then**

**Begin**

**m:=c[i];**

**c[i]:=c[j];**

**c[j]:=m;**

**End;**

*{Вивід відсортованого масиву}*

**For i:=1 to 12 do**

**Write(c[i]:6:2); readln;**

**End.**

**Результат:** -4.3, -0.1, 0.1, 1.0, 1.0, 1.2, 1.3, 2.2, 4.3, 5.3, 6.0, 6.1.

### **Варіанти завдань для самостійної роботи:**

1. Обчислити суму квадратів додатних та від'ємних елементів масиву  $A(6) = (6,0; -5,3; 1,2; -3,3; 0,0; 6,3)$ . Надрукувати добуток цих чисел.
2. Впорядкувати елементи масиву  $D(5) = (3,2; -6,3; -8,3; 0,0; 1,2)$ , розмістивши їх в порядку спадання значень в тому ж масиві.
3. Обчислити значення функції по формулі  $y = 1,5x^2 \cos x^3 - 38 \sin^3 x$ ,  $-2,2 \leq x \leq 2,4$ ,  $\Delta x = 0,2$ . Отримані значення у організувати у вигляді масиву, елементи якого розмістити в порядку спадання їх значень.
4. Маємо два масиви  $X(4) = (5,3; 2,3; 3,6; 7,9)$  та  $Y(4) = (-3,2; -6,3; 3,6; 7,9)$ . Необхідно додати поелементно ці масиви, сформувавши масив  $Z(8)$ , розмістивши елементи в порядку зростання значень.

5. Вивести на друк значення 2-го і 4-го від'ємного елемента масиву  $B(7) = (-2,3; 0,1; -2,3; 4,1; -3,2; -2,0; -4,0)$  разом із їх порядковими номерами.
6. Надрукувати перші три додатні елементи масиву  $A(8) = (3,2; -6,3; 2,1; 6,2; -2,1; 3,3; 7,8; 8,3)$  разом із їх порядковими номерами.
7. Переставити елементи в масиві  $B(8) = (1,1; 2,3; -6,2; 3,6; 5,6; -3,3; -2,1; 5,5)$  так, щоб спочатку йшли елементи з парними номерами, а потім - з непарними.
8. Обчислити  $s = 1 + 2 \cdot \frac{x}{2} + \dots + \frac{n^2 + 1}{n} \left(\frac{x}{2}\right)^2$ , де  $n=5$  для  $0,3 \leq x \leq 2,7$ ,  $\Delta x = 0,3$ .  
Результати записати в масив  $A(9)$ .
9. Обчислити  $s = 1 - \frac{x^2}{2} + \dots + (-1)^n \frac{x^{2n}}{(2n)!}$  при  $n=5$ , якщо  $x$  змінюється на відрізьку  $0,1 \leq x \leq 1,6$  з кроком  $\Delta x = 0,1$ . Результати обчислень записати в масив  $B(16)$ .
10. Знайти найменший серед додатніх елементів масиву  $A(11) = (-3,2; 2,1; 0,0; 1,3; -4,2; -6,6; 7,1; 0,1; 0,3; 0,2; 0,11)$ .
11. В масиві  $A(7) = (-3,2; 0; 3,1; 0,0; 1,2; 3,0)$  замінити всі нульові елементи значеннями  $z = \sqrt{\frac{75+n}{75-n}}$ , де  $n$ -порядковий номер нульового елемента.
12. Утворити масив  $C(10)$  на основі злиття двох заданих масивів  $A(5) = (-1,2; 1,3; 2,6; -3,9; 2,6)$  та  $B(5) = (0,0; 1,3; 6,3; -2,6; 5,6)$  так, щоб елементи масиву  $A$  стали на парних місцях, а масиву  $B$  - на непарних.
13. В заданому масиві  $B(8) = (6,3; -2,2; 3,1; 0,0; 2,1; 1,3; -3,3; 2,1)$  зробити заміну: від'ємні елементи замінити на число 10, а додатні - на 0.
14. Знайти окремо середнє арифметичне від'ємних та середнє арифметичне додатніх елементів масиву  $A(8) = (1,1; -3,3; -6,9; 1,3; 5,3; 6,3; -3,2; 8,1)$  і надрукувати їх різницю.
15. В заданому масиві  $T(8) = (3,2; -3,6; 2,1; -2,1; -2,1; 6,1; 2,1; -3,2)$  на місці від'ємних елементів поставити нулі, а на місці додатніх - їх квадрати.
16. В заданому масиві  $C(10) = (3,6; -3,2; 9,3; 4,2; 9,3; 4,2; -2,1; 6,3; 7,8; -8,9)$  переставити елементи так, щоб спочатку йшли всі від'ємні елементи, а потім - додатні.
17. Надрукувати останні три від'ємні елементи масиву  $A(9) = (1,3; -2,3; 2,1; -2,1; -2,3; 6,3; -2,1; 1,0; -2,0)$ .
18. Знайти середнє арифметичне елементів масиву  $B(5) = (-2,1; 3,1; 2,2; -2,2; -3,6)$  і на місці від'ємних елементів поставити це значення.
19. Впорядкувати масиви  $X(5) = (3,2; -2,1; 0,0; -1,1; 3,2)$  та  $Y(5) = (9,3; 9,2; 2,1; -3,1; 8,7)$  по зростанню значень елементів і на їх основі утворити масив  $Z(10)$ , в якому спочатку йдуть елементи впорядкованого масиву  $Y$ , а потім  $-X$ .
20. Обчислити модуль різниці між найбільшим та найменшим елементами масиву  $B(6) = (3,2; 3,4; -6,8; -5,3; 0; 1,1)$ .

21. Знайти найменший серед додатніх елементів масиву  $A(7) = (-3,6; 2,1; 0,0; -2,1; 2,4; 2,1; 7,2)$ .
22. Обчислити добуток різниць між сусідніми елементами масиву  $B(6) = (3,2; 2,1; 2,1; 3,1; 4,2; -3,0)$ .
23. Обчислити величину  $Z = \left( \sum_{i=1}^8 (y_i - 6)^2 \right) / \left( \sum_{i=1}^8 y_i \right)$ , де  $y(8) = (3,2; -6,3; 2,1; 3,2; 5,6; -3,1; 2,1; 4,3)$ .



24. Обчислити величину  $P = \left( \sum_{i=1}^6 A_i \right) / \left( \sum_{j=1}^4 B_j \right)$ , де  $A(6) = (1,2; -3,1; 0,1; 2,3;$

$5,6; 6,1)$ ,  $B(4) = (3,2; -6,3; 2,4; -2,2)$ .

25. Замінити додатні елементи масиву  $B(6) = (3,2; 3,4; -6,8; -5,3; 0; 1,1)$  на -1, від'ємні - 0, нулі замінити 2. Результат відсортувати в порядку зростання значень.

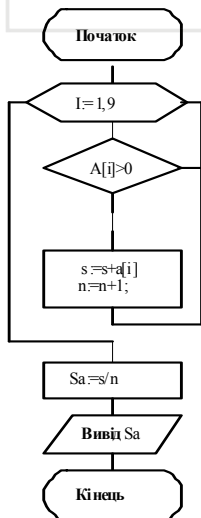
### Завдання 7.4

Скласти алгоритм та написати програму обробки одновимірного масиву згідно варіанта.

#### Приклад 7.4

Знайти середнє арифметичне значення додатних елементів масиву  $C(6) = (6,3; 5,3; -2,2; -3,3; 0,0; 2,1)$ .

#### Розв'язання:



**Program 17z4;**

**Uses crt;**

**Const a:array [1..6] of real =(6.3,5.3, -2.2, -3.3, 0.0, 2.1);**

**Var i,n:integer;**

**s,sa:real;**

**Begin**

**Clrscr;**

**For i:=1 to 6 do**

**if a[i]>0 then**

**Begin**

*{Знаходження суми додатніх елементів}*

**s:=s+a[i];**

**n:=n+1;**

**End;**

Sa:=s/n; {Знаходження середнього арифметичного}

Writeln('Середнє арифметичне додатніх елементів',sa:8:2);

End.

Результат: 4,57.



Національний університет  
водного господарства  
та природокористування

**Варіанти завдань для самостійної роботи:**

1. У масиві A(n) ( $n \leq 9$ ) визначити квадрати додатних елементів.
2. У масиві A(n) ( $n \leq 8$ ) є хоча б один від'ємний елемент. Визначити суму значень елементів до першого від'ємного.
3. У масиві A(n) ( $n \leq 9$ ) визначити мінімальний елемент та його номер.
4. У масиві A(n) ( $n \leq 10$ ) визначити кількість додатних та від'ємних елементів.
5. У масиві A(n) ( $n \leq 7$ ) хоча б один нуль. Визначити кількість додатних та кількість від'ємних елементів до першого нуля.
6. У масиві A(n) ( $n \leq 10$ ) визначити суму від'ємних елементів.
7. У масиві A(n) ( $n \leq 7$ ) визначити максимальний по модулю елемент.
8. У масиві A(n) ( $n \leq 9$ ) є хоча б один нульовий елемент. Визначити кількість елементів після першого нуля.
9. У масиві A(n) ( $n \leq 12$ ) визначити різницю між максимальним та мінімальним елементами.
10. З додатних елементів масиву A(n) ( $n \leq 14$ ) сформувати масив B. Роздрукувати утворений масив та підрахувати кількість елементів у ньому.
11. У масиві A(n) ( $n \leq 11$ ) є хоча б один нуль. Знайти мінімальний по модулю елемент у частині масиву після останнього нуля.
12. Із елементів масиву A(n) ( $n \leq 13$ ) знайти мінімальний серед додатних елементів масиву.
13. У масиві A(n) ( $n \leq 13$ ) знайти мінімальний серед від'ємних елементів та вказати його номер.
14. У масиві A(n) ( $n \leq 15$ ) знайти максимальний серед від'ємних елементів та вказати його номер.
15. У масиві A(n) ( $n \leq 14$ ) є хоча б один нульовий елемент. Визначити кількість елементів, більших за 10, які слідують за цим нулем.
16. З масиву A(n) ( $n \leq 15$ ) надрукувати перші три від'ємні елементи.
17. В масиві C(n) ( $n \leq 16$ ) впорядкувати в порядку зростання додатні елементи.
18. В масиві A(n) ( $n \leq 13$ ) впорядкувати в порядку зростання додатні елементи.



19. В масиві  $V(n)$  ( $n \leq 14$ ) впорядкувати в порядку спадання від'ємні елементи.
20. Дано два масиви  $A(n)$  ( $n \leq 16$ ) і  $B(m)$  ( $m \leq 14$ ). Злити їх в один масив та відсортувати його в порядку зростання його елементів.
21. На основі двох масивів  $A(n)$  ( $n \leq 6$ ) та  $B(m)$  ( $m \leq 6$ ) сформувати третій  $C(6)$ , елементи якого дорівнюють добутку відповідних елементів масивів  $A$  і  $B$ .
22. Масив  $V(n)$  ( $n \leq 10$ ) складається з елементів, значення яких співпадають з їх порядковими номерами. Побудувати цей масив і знайти середнє арифметичне його елементів.
23. В масиві  $C(k)$  ( $k \leq 10$ ) переставити елементи за правилом: 10-й міняється місцями з 1-м, 9-й з 2-м, ...5-й із 6-м.
24. Задано два масиви  $A(n)$  ( $n \leq 5$ ) та  $B(m)$  ( $m \leq 5$ ); 8,3; 9,4; -3,2; 6,1). Елементи масиву  $B$  збільшити на 10, якщо відповідний елемент масиву  $A$  більший за нуль.
25. Обчислити суму квадратів додатніх елементів масиву  $V(m)$  ( $m \leq 7$ ) і суму квадратів від'ємних елементів. Надрукувати добуток цих чисел.

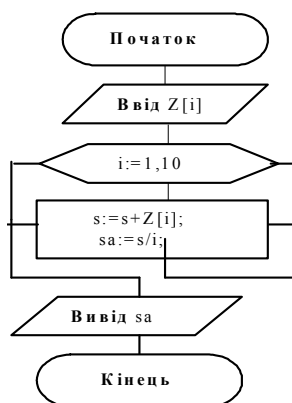
### Завдання 7.5

Скласти алгоритм та написати програму обробки одновимірного масиву згідно варіанту.

#### Приклад 7.5

Дані про заробітну плату  $N$  працівників зберігаються в масиві  $Z(N)$ . Визначити середню зарплату та вивести її на екран.

#### Розв'язання:



```

Program 17z5;
Uses crt;
Var i:integer;
      s,sa:real;
      Z:array [1..10] of real;
  
```

**Begin**

**Clrscr;**

**Writeln**('Введіть 10 працівників');  
{Введення елементів масиву Z}

**For** i:=1 to 10 **do**

**Readln**(Z[i]);

**For** i:=1 to 10 **do**

```

s:=s+Z[i];
{Знаходження середньої зарплати}
sa:=s/i;
Writeln('Середня зарплата',sa:8:2);
End.

```

### **Варіанти завдань для самостійної роботи:**

1. Дані про заробітну плату  $N$  працівників зберігаються в масиві  $Z(N)$ . Визначити номер працівника, який має найбільшу зарплату, та вивести на екран номер та цю зарплату.
2. Дані про заробітну плату  $N$  працівників зберігаються в масиві  $Z(N)$ . Визначити номер працівника, який має найменшу зарплату, та вивести на екран номер та цю зарплату.
3. Дані про заробітну плату  $N$  працівників зберігаються в масиві  $Z(N)$ . Підрахувати суму зарплат усіх працівників.
4. Дані про заробітну плату  $N$  працівників зберігаються в масиві  $Z(N)$ . Визначити середню зарплату та вивести її на екран.
5. Дані про заробітну плату  $N$  працівників зберігаються в масиві  $Z(N)$ . Визначити номери та зарплати тих працівників, зарплата яких не перевищує заданої  $Z_0$ .
6. Дані про заробітну плату  $N$  працівників зберігаються в масиві  $Z(N)$ . Визначити номери та зарплати тих працівників, зарплата яких не менша заданої  $Z_0$ .
7. Дані про заробітну плату  $N$  працівників зберігаються в масиві  $Z(N)$ . Визначити номери та зарплати тих працівників, зарплата яких перевищує  $Z_1$  не більша за  $Z_2 (Z_1 < Z_2)$ .
8. Дані про заробітну плату  $N$  працівників зберігаються в масиві  $Z(N)$ . Визначити номери та зарплати тих працівників, зарплата яких не перевищує  $Z_1$  або не менша за  $Z_2 (Z_1 < Z_2)$ .
9. В масиві  $S(M)$  зберігаються сальдо по  $M$  рахунках (при цьому дебетове сальдо виражене додатньою величиною, кредитове - від'ємною). Вивести на екран усі дебетові сальдо та підрахувати їх кількість.
10. В масиві  $S(M)$  зберігаються сальдо по  $M$  рахунках (при цьому дебетове сальдо виражене додатньою величиною, кредитове - від'ємною). Вивести на екран усі кредитові сальдо та підрахувати їх кількість.
11. В масиві  $S(M)$  зберігаються сальдо по  $M$  рахунках (при цьому дебетове сальдо виражене додатньою величиною, кредитове - від'ємною). Вивести на екран суму всіх дебетових сальдо та їх номери.
12. В масиві  $S(M)$  зберігаються сальдо по  $M$  рахунках (при цьому дебетове сальдо виражене додатньою величиною, кредитове - від'ємною). Вивести на екран суму всіх кредитових сальдо та їх номери.

13. В масиві  $S(M)$  зберігаються сальдо по  $M$  рахунках ( при цьому дебетове сальдо виражене додатньою величиною, кредитове – від'ємною). Вивести на екран спочатку всі дебетові сальдо, а потім – усі кредитові.
- 14 В масиві  $S(M)$  зберігаються сальдо по  $M$  рахунках ( при цьому дебетове сальдо виражене додатньою величиною, кредитове – від'ємною) Вивести на екран спочатку всі кредитові сальдо, а потім – усі дебетові.
- 15 В масиві  $S(M)$  зберігаються сальдо по  $M$  рахунках ( при цьому дебетове сальдо виражене додатньою величиною, кредитове – від'ємною). Вивести на екран найменше серед дебетових сальдо.
16. В масиві  $S(M)$  зберігаються сальдо по  $M$  рахунках ( при цьому дебетове сальдо виражене додатньою величиною, кредитове – від'ємною) Вивести на екран найбільше серед дебетових сальдо.
17. В масиві  $S(M)$  зберігаються сальдо по  $M$  рахунках ( при цьому дебетове сальдо виражене додатньою величиною, кредитове – від'ємною). Вивести на екран найменше серед кредитових сальдо.
18. В масиві  $S(M)$  зберігаються сальдо по  $M$  рахунках ( при цьому дебетове сальдо виражене додатньою величиною, кредитове – від'ємною) Вивести на екран найбільше серед кредитових сальдо.
19. В масиві  $S(M)$  зберігаються сальдо по  $M$  рахунках ( при цьому дебетове сальдо виражене додатньою величиною, кредитове – від'ємною) Вивести на екран найбільше серед кредитових та найменше серед дебетових сальдо, додати знайдені значення.
20. Дані про нараховану заробітну плату усіх працівників одного підрозділу ( $N$  працівників) зберігаються в масиві  $Z1(N)$ , другого підрозділу ( $M$  працівників) – в масиві  $Z2(M)$ . Необхідно сформуванати єдиний масив  $Z3(N+M)$ , який містить інформацію про зарплату працівників обох підрозділів.
21. Дані про нараховану заробітну плату усіх працівників одного підрозділу ( $N$  працівників) зберігаються в масиві  $Z1(N)$ , другого підрозділу ( $M$  працівників) – в масиві  $Z2(M)$ . Необхідно сформуванати єдиний масив  $Z3(N+M)$ , який містить інформацію про зарплату працівників обох підрозділів та знайти середню.
22. Дані про нараховану заробітну плату працівників підрозділу ( $N$  працівників) зберігаються в масиві  $Z1(N)$ . В масиві  $Z2(M)$  зберігається список тих працівників, які вже отримали заробітну плату. Необхідно сформуванати масив  $Z3(N-M)$ , який містить інформацію про працівників, які її ще не отримали.
23. Дані про заробітну плату  $N$  працівників зберігаються в масиві  $Z(N)$ . Визначити порядкові номери та зарплати тих працівників, зарплата яких перевищує задане  $Z_1$ .
24. Дані про заробітну плату  $N$  працівників зберігаються в масиві  $Z(N)$ . Визначити суму двох найменших заробітних плат та порядкові номери працівників, яким вони належать.

25. Дані про заробітну плату  $N$  працівників зберігаються в масиві  $Z(N)$ . Визначити, на скільки середня заробітна плати відрізняється від найменшої.

### **Контрольні запитання:**

1. Дайте означення масиву.
2. Як здійснюється опис масиву?
3. Наведіть приклад введення, виведення одновимірного масиву.
4. Як розміщуються елементи одновимірного масиву в пам'яті ПК?
5. Чим характеризується кожен масив?
6. Що розуміють під сортуванням масиву?
7. На які типи поділяються алгоритми сортування?
8. Наведіть відомі Вам методи сортування масивів.
9. Наведіть алгоритм сортування масиву методом "бульбашок".
10. Проаналізуйте сортування масиву згідно алгоритму Шелла.

### **Лабораторна робота №8**

#### **Тема: Багатовимірні масиви**

**Мета роботи:** Ознайомитися з особливостями роботи двовимірних масивів, навчитися виконувати найпростіші дії над елементами масиву.

#### **Хід роботи:**

1. Розробити алгоритми обробки двовимірного масиву згідно варіантів завдань для самостійної роботи.
2. Написати відповідні програми на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.

#### **Завдання 8.1**

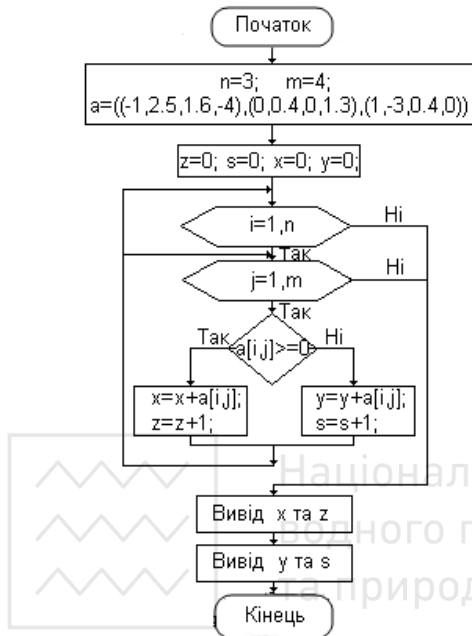
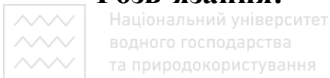
Для кожного завдання, наведеного нижче, розробити алгоритм і написати відповідну програму.

#### **Приклад 8.1**

В матриці  $C(3,4) = \begin{pmatrix} -1 & 2,5 & 1,6 & -4 \\ 0 & 0,5 & 0 & 1,3 \\ 1 & -3 & 0,4 & 0 \end{pmatrix}$  підрахувати і вивести на друк

суму додатних та від'ємних елементів, підрахувати їх кількість.

### Розв'язання:



### Program l8z1;

```

uses crt;
const n=3; m=4;
a:array[1..n,1..m] of real=
((-1,2.5,1.6,-4),(0,0.5,0,1.3),(1,-3,0.4,0));
var x,y:real; z,s,i,j:integer;
Begin
  clrscr;
  z:=0; s:=0; x:=0; y:=0;
  for i:=1 to n do
    for j:=1 to m do
      Begin
        {Визначення додатних елементів}
        if (a[i,j])>=0 then begin
          {Сума додатних елементів}
          x:=x+a[i,j];
          {Кількість додатних елементів}
          z:=z+1; end
        else begin
          {Обчислення суми від'ємних елементів
          та їх кількості}
          y:=y+a[i,j];
          s:=s+1;
        end;
      end;
      writeln('summa dodatnih elementiv
      =' ,x,' kilkist dodatnih elementiv =' ,z);
      writeln('summa videmnih elementiv
      =' ,y,' kilkist videmnih elementiv =' ,s);
    readln;
  end.
  
```

**Варіанти завдань для самостійної роботи:**

1. Впорядкувати елементи другого стовпця матриці

$$A(3,3) = \begin{pmatrix} 2,1 & 1,5 & 0,3 \\ 0,2 & 0,5 & 0,4 \\ 1,3 & 2,0 & 3,1 \end{pmatrix} \text{ за зростанням. Вивести стару та нову матриці.}$$

2. Для заданого двовимірного масиву знайти середнє арифметичне  $\bar{x}$  і

середнє квадратичне відхилення  $\delta$  за формулами  $\bar{x} = \frac{1}{mn} \sum_{i=1}^n \sum_{j=1}^m x_{ij}$ ,

$$\delta = \sqrt{\frac{1}{mn-1} \sum_{i=1}^n \sum_{j=1}^m (x_{ij} - \bar{x})^2}, \quad x(2,4) = \begin{pmatrix} 1,4 & 1,2 & 1,1 & 1,5 \\ 1,7 & 1,5 & 1,3 & 1,6 \end{pmatrix}.$$

3. Знайти найбільший елемент матриці  $z = \{z_{ij}\}$ , елементи якої

обчислюються за формулою  $z_{ij} = x_i y_j$ ,  $i = \overline{1,5}$ ,  $j = \overline{1,3}$ ,

$$x(5) = (2,1; 0,5; -1,3; 4,2; 0,1), \quad y(3) = (0; -0,2; 1,5).$$

4. В матриці В знайти суму елементів, що лежать вище головної

діагоналі  $B(4,4) = \begin{pmatrix} 3 & 4 & -1,5 & 0 \\ 2 & -1,2 & 3 & 1 \\ 1 & 2 & 0,3 & 4 \\ 1,1 & 2,5 & 0 & 1 \end{pmatrix}.$

5. В матриці В (див. варіант 4) знайти добуток елементів головної діагоналі.

6. Знайти квадрат найбільшого елемента масиву

$$Y(3,4) = \begin{pmatrix} 3 & 0 & 0,1 & 2,5 \\ -1 & 2 & 5 & 1,3 \\ 4 & -2,5 & 3 & 0,5 \end{pmatrix}.$$

7. Знайти суму двох матриць:  $A(2,3) = \begin{pmatrix} 3 & 1 & 5 \\ -2 & 0 & 3 \end{pmatrix}$ ,  $B(2,3) = \begin{pmatrix} 1 & 3 & 4 \\ 3 & 0 & 2 \end{pmatrix}$ ,

користуючись формулою  $c_{ij} = a_{ij} + b_{ij}$ ,  $i = \overline{1,2}$ ,  $j = \overline{1,3}$ .

8. В матриці В поміняти місцями 1-й і 2-й стовпці  $B(3,4) = \begin{pmatrix} 2 & 0 & 3 & 4 \\ 1 & 2 & 0 & 5 \\ -2 & 3 & 4 & 5 \end{pmatrix}.$

9. Знайти суму елементів матриці  $A = \{a_{ij}\}$ , що задовольняють

умові:  $-1 \leq a_{ij} \leq 1$ , якщо  $A(3,4) = \begin{pmatrix} -0,7 & 1,2 & 3,5 & 0,3 \\ 0,5 & 0,7 & -1,1 & -3,5 \\ 0,9 & 0,8 & 2,7 & 1 \end{pmatrix}.$

10. Задана матриця  $z(3,3) = \begin{pmatrix} -1,5 & 2 & 3,5 \\ 0,5 & 0 & 1,2 \\ -2 & 1 & 0,7 \end{pmatrix}$ . Знайти  $c = \frac{z_{\min} + z_{\max}}{2}$ , де

$$z_{\min} = \min \{z_{ij}\}, \quad z_{\max} = \max \{z_{ij}\}, \quad i = \overline{1,3}, \quad j = \overline{1,3}.$$

11. Сформувати матрицю у вигляді таблиці значень функції  $z = x^3 e^{-yx} \sin x$  для таких значень аргументів:  $x(4) = \{1; 1,2; -0,3; 0,7\}$ ;  $y(4) = \{3; 8; 2; -1,5\}$ . Вивести її на екран.

12. Задана матриця  $A = \begin{pmatrix} 2,1 & 1,5 & 0,3 \\ 0,2 & 0,5 & 0,4 \\ 1,3 & 2,0 & 3,1 \end{pmatrix}$ . Обчислити елементи нової матриці,

які визначаються за формулою  $x_{ij} = \frac{a_{ij}}{S}$ , де  $S$  – сума елементів, які стоять на її головній діагоналі.

13. В заданому масиві  $A(3,4) = \begin{pmatrix} 3 & 4 & 1,3 & 0,7 \\ 1 & 3 & 2,2 & 0,8 \\ 5 & 2 & 3,5 & 0,6 \end{pmatrix}$  підрахувати окремо суму

кожного стовпця. З отриманих сум сформувати масив  $X$ .

14. Обчислити елементи матриці  $c = \{c_{ij}\}$ , якщо

$$c_{ij} = \begin{cases} a_{ij} & , \text{якщо } |a_{ij}| > |b_{ij}| \\ b_{ij} & , \text{якщо } |a_{ij}| \leq |b_{ij}| \end{cases}, \quad B(3,3) = \begin{pmatrix} -2 & 2,3 & -7,1 \\ 0,8 & 3,6 & -5,6 \\ 3 & 6 & 4 \end{pmatrix}.$$

15. Провести перетворення матриці  $A$  таким чином: додатні елементи замінити на число 2, від'ємні - на 1, нульові - на 3:

$$A(3,3) = \begin{pmatrix} -1,5 & 0,6 & -2,1 \\ 0,5 & 0 & 0,7 \\ 0 & -1,3 & 0 \end{pmatrix}$$


16. Транспонувати матрицю (зробити рядки стовпцями)

$$Z(3,4) = \begin{pmatrix} 2 & 5 & 1 & 0 \\ 0 & 3 & -2 & 4 \\ 1 & 2 & 0 & 3 \end{pmatrix}.$$

17. Знайти добуток двох матриць  $A = \{a_{ij}\} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 2 & 1 \end{pmatrix}$  і  $B = \{b_{ij}\} = \begin{pmatrix} 1 & 0 \\ 2 & 1 \\ 3 & 5 \end{pmatrix}$ ,

користуючись формулою  $c_{ik} = \sum_{j=1}^3 a_{ij} \cdot b_{jk}$ .

18. Вивести на друк у вигляді масиву X всі невід'ємні елементи матриці

  $A(3,4) = \begin{pmatrix} -2 & 0 & 1 & 5 \\ 1,3 & -1 & 3 & -2 \\ 2,1 & 5 & 0 & 1 \end{pmatrix}$ .

19. Обчислити компоненти вектора  $C = A \cdot b$ , де  $A = \{a_{ij}\}$ ,

$$b = \{b_j\}, c_i = \sum_{j=1}^3 a_{ij} b_j, A(3,3) = \begin{pmatrix} 1,8 & 0,1 & 1,3 \\ 0,9 & -1 & 3,1 \\ -1,5 & 0,5 & 1 \end{pmatrix}, b = \begin{pmatrix} 1 \\ 3,2 \\ 2 \end{pmatrix}.$$

20. Задана матриця  $A(3,3) = \begin{pmatrix} 1 & 0,7 & 3,2 \\ -2 & 0,3 & 1,2 \\ 2 & 3,7 & 0,5 \end{pmatrix}$ . Утворити нову матрицю

$X = \{x_{ij}\}$ , кожен елемент якої дорівнює добутку відповідного елемента матриці A на номер рядка:  $x_{ij} = i \cdot a_{ij}$ .

21. Перетворити матрицю  $B(3,4) = \begin{pmatrix} -1 & 2 & 0 & 4 \\ 3 & -5 & 4 & 0 \\ 0 & 1 & -1 & 2 \end{pmatrix}$ , замінивши додатні

елементи номером рядка, в якому вони знаходяться, від'ємні елементи - номером стовпчика, в якому вони знаходяться, нульові елементи - сумою відповідного рядка і стовпця.

22. Сформувати масив X з квадратів тих елементів матриці

$$A(3,3) = \begin{pmatrix} 3 & 1 & 0 \\ 0 & 3 & 2 \\ 2 & 2 & 0 \end{pmatrix}, \text{ які стоять на побічній діагоналі.}$$

23. Всі елементи другого стовпця матриці B збільшити в 2 рази.

$$B(3,4) = \begin{pmatrix} 3,1 & 1,2 & 0,5 & -1,3 \\ 2 & -1,5 & 0,3 & 2,1 \\ 1,5 & 0,5 & 0 & 1,8 \end{pmatrix}. \text{ Вивести на друк стару і нову матрицю.}$$



24. В матриці  $Y(3,3) = \begin{pmatrix} 3,1 & 2,5 & 6 \\ -1 & 0 & 1 \\ 2,5 & 0,5 & 3 \end{pmatrix}$  знайти найбільший елемент і вивести

на друк разом з його індексами.

25. Задана матриця А. Отримати нову матриця шляхом ділення всіх її елементів на найбільший по модулю елемент.

$$A(2,3) = \begin{pmatrix} 3 & 1,5 & -1,3 \\ -8 & 5 & 0 \end{pmatrix}$$



Національний університет  
водного господарства  
та природокористування

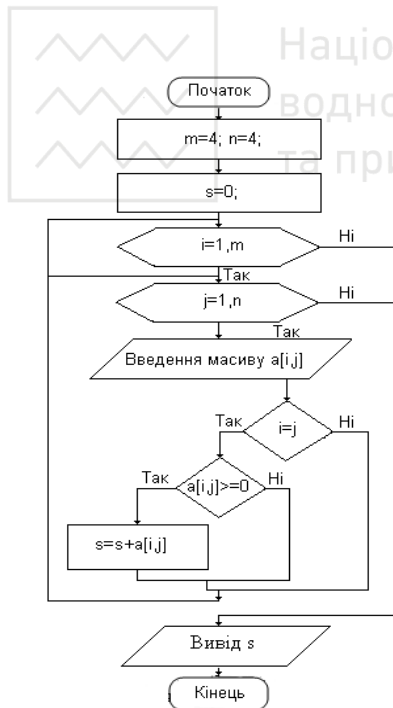
## Завдання 8.2

Скласти програму знаходження числових характеристик матриці згідно вказаної умови.

### Приклад 8.2

В матриці  $A(m,n)$  ( $m \leq 7, n \leq 5$ ) знайти суму невід'ємних елементів, які лежать на головній діагоналі.

### Розв'язання:



```

Program l8z2;
Uses crt;
const m=4; n=4;
var a:array[1..m,1..n] of real; s,z:real;
i,j,d:integer;
begin
  clrscr;
  s:=0;
  {Введення матриці A }
  for i:=1 to m do
    for j:=1 to n do begin
      writeln('введення element ',i,', ',j);
      readln(a[i,j]);
      {Визначення додатних елементів
      головної діагоналі}
      if (i=j) then begin
        {Обчислення суми додатних
        елементів головної діагоналі}
        s:=s+a[i,j];
      end;
    end;
  end;

```

```

end;
end;
writeln('suma nevidemnih
        elementiv=',s:3:2);
readln;
end.

```



**Варіанти завдань для самостійної роботи:**

1. У цілочисельній матриці  $A(n,n)$  ( $n \leq 5$ ) знайти найменший спільний дільник головної діагоналі.
2. У матриці  $A(m,n)$  ( $m \leq 5, n \leq 7$ ) визначити середні арифметичні значення елементів стовпців.
3. У матриці  $A(m,n)$  ( $m \leq 5, n \leq 4$ ) визначити максимальні елементи кожного стовпця та їх номери.
4. У матриці  $A(m,n)$  ( $m \leq 5, n \leq 5$ ) визначити добуток максимального та мінімального елементів.
5. У матриці  $A(m,n)$  ( $m \leq 5, n \leq 8$ ) визначити різниці максимального та мінімального елементів кожного стовпця.
6. У матриці  $A(m,n)$  ( $m \leq 5, n \leq 7$ ) знайти добуток елементів стовпця, в якому знаходиться максимальний елемент.
7. У матриці  $A(m,n)$  ( $m \leq 7, n \leq 4$ ) визначити найменший та найбільший елементи кожного рядка.
8. У матриці  $A(m,n)$  ( $m \leq 5, n \leq 5$ ) знайти добуток ненульових елементів, які лежать на головній діагоналі.
9. У матриці  $A(m,n)$  ( $m \leq 7, n \leq 5$ ) знайти суму максимальних елементів її рядків та їх індекси.
10. У цілочисельній матриці  $A(m,n)$  ( $m \leq 5, n \leq 6$ ) визначити мінімальні елементи кожного стовпця та їх індекси.
11. Замінити всі парні елементи цілочисельної матриці  $A(m,n)$  ( $m \leq 3, n \leq 6$ ) нулями.
12. У матриці  $A(m,n)$  ( $m \leq 7, n \leq 5$ ) визначити суми та добутки елементів рядків.
13. Замінити всі елементи матриці  $A(m,n)$  ( $m \leq 5, n \leq 4$ ), сума індексів яких парна, добутками відповідних індексів.
14. Піднести до квадрату всі від'ємні елементи матриці  $A(m,n)$  ( $m \leq 5, n \leq 5$ ), які лежать вище головної діагоналі.
15. Знайти середнє арифметичне додатних елементів матриці  $A(m,n)$  ( $m \leq 6, n \leq 6$ ), які лежать нижче головної діагоналі.
16. Знайти суму елементів масиву  $A(m,n)$  ( $m \leq 3, n \leq 5$ ), які мають хоча б один непарний індекс.

17. Визначити номери елементів масиву  $A(m,n)$  ( $m \leq 6, n \leq 2$ ), модуль яких більший 5.
18. Визначити мінімальний по модулю елемент та його індекси в масиві  $A(m,n)$  ( $m \leq 7, n \leq 5$ ).
19. У матриці  $A(n,n)$  ( $n \leq 5$ ) знайти максимальні елементи кожного стовпця.
20. У матриці  $A(n,n)$  ( $n \leq 7$ ) знайти кількість від'ємних елементів, що лежать нижче головної діагоналі.
21. У матриці  $A(m,n)$  ( $m \leq 5, n \leq 5$ ) знайти суму елементів головної та побічної діагоналей.
22. Знайти суму елементів матриці  $A(m,n)$  ( $m \leq 5, n \leq 5$ ), які стоять над і під головною діагоналлю.
23. Просумувати елементи кожного стовпця матриці  $A(n,n)$  ( $n \leq 4$ ). Отримані суми роздрукувати, занести в масив і вибрати серед них максимальну.
24. У матриці  $A(n,n)$  ( $n \leq 5$ ) впорядкувати другий рядок за зростанням. Вивести стару та нову матриці.
25. Задана матриця  $A(n,n)$  ( $n \leq 5$ ). Отримати нову матрицю шляхом ділення всіх її елементів на найбільший по модулю від'ємний елемент.

### Завдання 8.3

Скласти програму обробки та перетворення двовимірного масиву. Передбачити виведення початкової та перетвореної матриці.

#### Приклад 8.3

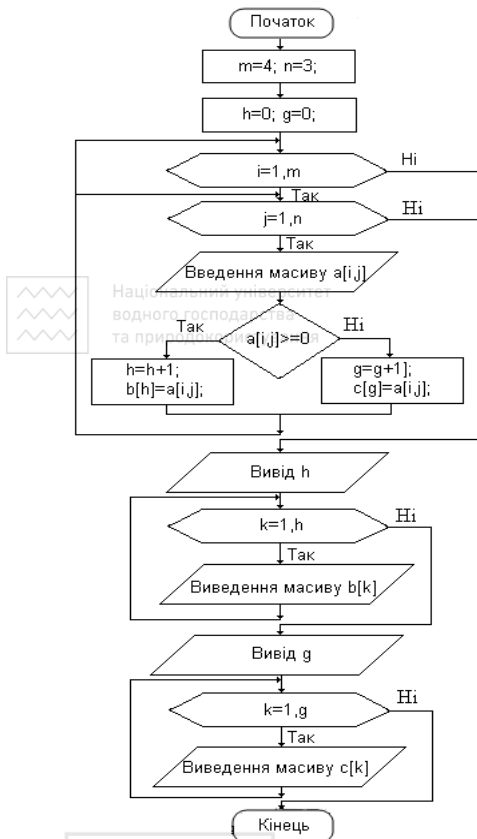
З елементів матриці  $A(m,n)$  ( $m \leq 4, n \leq 3$ ) утворити два масиви: В – масив додатних елементів матриці А та С – масив від'ємних елементів матриці А. Підрахувати кількість елементів в даних масивах.

#### Розв'язання:

```

Program l8z3;
Uses crt;
const m=4; n=3;
var a:array[1..m,1..n] of real;
    b,c:array[1..100] of real;
    i,j,k,h,g:integer; f,d:real;
begin
    clrscr;
    h:=0; g:=0;
    {Введення елементів матриці}
    writeln('vvedit elementi masivu');

```



```

for i:=1 to m do
  for j:=1 to n do begin
    readln(a[i,j]);
    {Визначення додатних елементів}
    if (a[i,j]>=0) then
      begin
        {Кількість додатних елементів}
        h:=h+1;
        b[h]:=a[i,j];
      end
      else
        {Кількість від'ємних елементів}
        begin g:=g+1; c[g]:=a[i,j];
      end; end;
    writeln('kilkist dodatnih=' ,h);
    for k:=1 to h do
      writeln('b[' ,k, ']=' ,b[k]:3:2);
    writeln('kikist videmnih=' ,g);
    for k:=1 to g do
      writeln('c[' ,k, ']=' ,c[k]:3:2);
    readln; end.
  
```

### Варіанти завдань для самостійної роботи:

1. Із матриці  $A(n,n)$  ( $n \leq 6$ ) отримати нову матрицю  $B(n,n)$  ( $n \leq 6$ ) шляхом ділення її елементів на найменший елемент головної діагоналі.
2. В матриці  $A(6,8)$  необхідно поміняти місцями рядок, який містить мінімальний елемент і рядок, що містить максимальний елемент. Вважати, що ці елементи єдині.
3. Із матриці  $A(m,n)$  ( $m \leq 5$ ,  $n < 6$ ) отримати числа  $a_1, \dots, a_m$ , де  $a_i$  - значення першого по порядку додатного елемента  $i$ -го рядка.
4. Транспонувати матрицю  $A(m,n)$  ( $m \leq 4$ ,  $n < 6$ ). Вивести початкову та перетворену матриці.
5. Координати  $m$  векторів задані матрицею  $A(m,n)$  ( $m \leq 6$ ,  $n < 7$ ). Необхідно обчислити довжини цих векторів, роздрукувати їх значення і серед цих значень знайти і вказати номер вектора мінімальної довжини.
6. Провести таке перетворення матриці  $A(m,n)$  ( $n \leq 4$ ), при якому всі додатні елементи замінюються на суму відповідних індексів, а від'ємні - на добуток індексів.

7. Знайти мінімальний елемент матриці  $A(m,n)$  ( $m \leq 5$ ) і вивести його. Елементи матриці, що лежать нижче головної діагоналі, замінити мінімальним елементом.
8. В матриці  $A(6,6)$  знищити 4-й рядок і вивести отриману матрицю.
9. Провести таке перетворення матриці  $A(m,n)$  ( $m \leq 5, n < 7$ ), при якому останній стовпчик займе місце першого, а всі інші змістяться на один стовпчик вправо.
10. Координати  $n$  векторів задані матрицею  $A(m,n)$  ( $m \leq 5, n \leq 6$ ). Обчислити довжини цих векторів, роздрукувати і занести їх значення в одновимірний масив. Серед елементів масиву знайти максимальний елемент та його номер.
11. В матриці  $A(3,7)$  знищити 5-й стовпчик і роздрукувати отриману матрицю.
12. Із матриці  $A(n,n)$  ( $n \leq 6$ ) отримати нову матрицю  $B(n,n)$ , шляхом ділення її елементів на максимальний по модулю елемент матриці.
13. Дана цілочисельна матриця  $A(n,n)$  ( $n \leq 5$ ). Знайти найменше із значень елементів стовпця, який має максимальну суму модулів елементів. Вказати номер стовпчика.
14. Провести таке перетворення матриці  $A(m,n)$  ( $m \leq 7, n \leq 3$ ), при якому останній рядок поміняється місцями з першим, передостанній з другим і т.д. Роздрукувати перетворену матрицю.
15. В матриці  $B(m,n)$  ( $m \leq 4, n \leq 6$ ), всі елементи якої різні, в кожному рядку вибирається елемент з найменшим значенням, потім серед цих чисел вибирається найбільше. Роздрукувати знайдений елемент.
16. Піднести до квадрата всі непарні елементи цілочисельної матриці  $A(m,n)$  ( $m \leq 5, n \leq 4$ ) і сформувати із цих квадратів одновимірний масив.
17. У матриці  $A(n,n)$  ( $n \leq 5$ ) замінити елементи головної діагоналі елементами побічної.
18. У матриці  $A(m,n)$  ( $m \leq 5, n \leq 4$ ) знайти суму елементів, які обрамляють дану матрицю і поміняти місцями мінімальний елемент лівої сторони на максимальний елемент правої.
19. Зробити таке перетворення матриці  $A(n,n)$  ( $n \leq 5$ ): замінити елементи, які стоять над головною діагоналлю, їм симетричними відносно головної діагоналі.
20. У матриці  $A(n,n)$  ( $n \leq 5$ ) замінити елементи, які стоять над побічно діагоналлю, їм симетричними відносно цієї діагоналі.
21. У матриці  $A(n,n)$  ( $n \leq 5$ ) поміняти елементи діагоналі, яка прилягає до головної діагоналі згори, на елементи діагоналі, яка прилягає до головної діагоналі знизу.
22. У матриці  $A(m,n)$  ( $m \leq 5, n \leq 4$ ) впорядкувати два останні рядки по спаданню.



На  
водни  
та й

23. Впорядкувати елементи матриці  $A(m,n)$  ( $m \leq 5, n \leq 4$ ), які стоять вище головної діагоналі, по зростанню і записати їх нижче головної діагоналі по рядках.
24. Впорядкувати елементи матриці  $A(m,n)$  ( $m \leq 5, n \leq 4$ ) по спаданню і розмістити їх по рядках.
25. Здійснити подвоєння всіх парних елементів матриці  $A(m,n)$  ( $m \leq 5, n \leq 4$ ) і записати всі подвоєння в одновимірний масив.



### Завдання 8.4

Інформація про прибутки  $M$  фірм за  $N$  місяців, задана у вигляді таблиці

<i>січень</i>	<i>лютий</i>	...	...	
$Z_{11}$	$Z_{12}$	...	$Z_{1n}$	- прибутки 1-ї фірми
$Z_{21}$	$Z_{22}$	...	$Z_{2n}$	- прибутки 2-ї фірми
...	...	...	...	
$Z_{m1}$	$Z_{m2}$	...	$Z_{mn}$	- прибутки $m$ -ї фірми

Скласти алгоритм та відповідну програму згідно нижче наведеного варіанту.

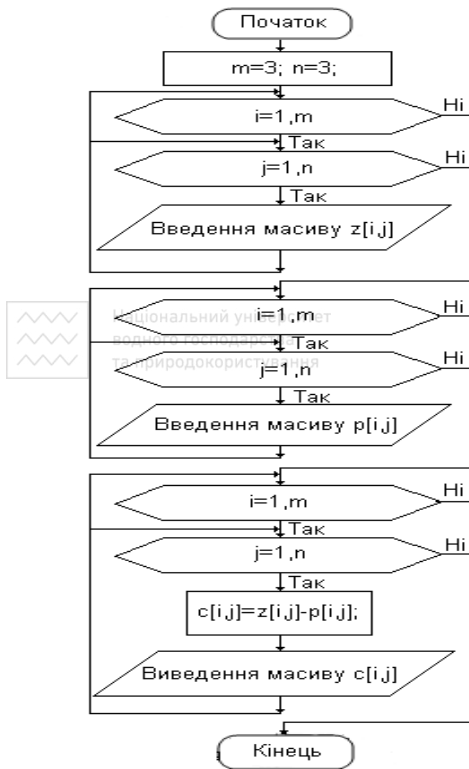
#### Приклад 8.4

Визначити заробітну плату кожного працівника за кожен місяць, яку вони отримували після вирахування прибуткового податку. Вивести на друк початкові дані та отримані результати.

**Розв'язання:**

```

Program L8z1;
const m=3; n=3;
var z,p,c:array[1..m,1..n] of real;
i,j,k:integer; h,l:real;
begin
  clrscr;
  writeln('vvedit zarplatu pracivnikiv');
  for i:=1 to m do begin
    for j:=1 to n do
      readln(z[i,j]);
  
```



```

end;
writeln('vvedit podatok pracivnukiv');
for i:=1 to m do begin
for j:=1 to n do
readln(p[i,j]);
end;
writeln('vuhidni rezultati:');
for i:=1 to m do
begin
for j:=1 to n do begin
{Визначення заробітної плати}
c[i,j]:=z[i,j]-p[i,j];
write(c[i,j]:3:2, ' ');
end;
writeln;
end; end.
  
```

### Варіанти завдань для самостійної роботи:

1. Підрахувати окремо сумарні прибутки кожної фірми за  $N$  місяців, запам'ятати їх і вивести разом із номерами фірм.
2. Підрахувати окремо сумарні прибутки всіх фірм за кожен місяць, запам'ятати їх і вивести, вказавши номер місяця.
3. Визначити найбільші прибутки кожної фірми протягом  $N$  місяців і вивести їх, вказавши номер фірми та номер місяця.
4. Визначити середні прибутки кожної фірми протягом  $N$  місяців і вивести їх, вказавши номер фірми.
5. Визначити найбільші прибутки серед усіх фірм за кожен місяць і вивести їх, вказавши номер фірми та номер місяця.
6. Визначити найменші прибутки серед усіх фірм за кожен місяць і вивести їх, вказавши номер фірми та номер місяця.
7. Визначити найбільший прибуток серед усіх фірм за всі місяці і вивести його, вказавши номер фірми та номер місяця.
8. Визначити найменший прибуток серед усіх фірм за всі місяці і вивести його, вказавши номер фірми та номер місяця.
9. Визначити сумарний прибуток усіх фірм за всі місяці і вивести його.
10. Визначити середній прибуток кожної фірми за всі місяці і вивести його, вказавши номер фірми.

11. Визначити середній прибуток усіх фірм за кожен місяць і вивести його, вказавши номер місяця.

Інформація про заробітну плату  $M$  працівників за  $N$  місяців подана у вигляді таблиці.

<i>Січень</i>	<i>Лютий</i>	...	...	
$Z_{11}$	$Z_{12}$	...	$Z_{1n}$	- заробітна плата 1-го працівника
$Z_{21}$	$Z_{22}$	...	$Z_{2n}$	- заробітна плата 2-го працівника
...	...	...	...	...
$Z_{m1}$	$Z_{m2}$	...	$Z_{mn}$	- заробітна плата $m$ -го працівника

Сума прибуткового податку працівників подана у вигляді наступної таблиці:

<i>Січень</i>	<i>Лютий</i>	...	...	
$P_{11}$	$P_{12}$	...	$P_{1n}$	- прибутковий податок 1-го працівника
$P_{21}$	$P_{22}$	...	$P_{2n}$	- прибутковий податок 2-го працівника
...	...	...	...	...
$P_{m1}$	$P_{m2}$	...	$P_{mn}$	- прибутковий податок $m$ -го працівника

Визначити найменші прибутки кожної фірми протягом  $N$  місяців і вивести їх, вказавши номер фірми та номер місяця.

12. Визначити заробітні плати кожного працівника за кожен місяць, які він отримав після вирахування прибуткового податку. Серед них знайти найбільшу та вивести її на екран, вказавши номер працівника та номер місяця.
13. Визначити заробітні плати кожного працівника за кожен місяць, які він отримав після вирахування прибуткового податку. Серед них знайти найменшу та вивести її на екран, вказавши номер працівника та номер місяця.
14. Визначити заробітні плати кожного працівника за кожен місяць, які він отримав після вирахування прибуткового податку. Серед них знайти



найбільшу за кожен місяць і вивести її на екран, вказавши номер місяця.

- Визначити заробітні плати кожного працівника за кожен місяць, які він отримав після вирахування прибуткового податку. Серед них знайти найменшу за кожен місяць і вивести її на екран, вказавши номер місяця.

Інформація про місячні сальдо  $M$  рахунків (дебетове сальдо, виражене додатньою величиною, кредитове – від’ємною) задана у вигляді таблиці



<i>Січень</i>	<i>Лютий</i>	...	...	
$S_{11}$	$S_{12}$	...	$S_{1n}$	- місячне сальдо по 1-рахунку
$S_{21}$	$S_{22}$	...	$S_{2n}$	-місячне сальдо по 2-рахунку
...	...	...	...	...
$S_{m1}$	$S_{m2}$	...	$S_{mn}$	- місячне сальдо по $m$ -рахунку

Скласти алгоритм та відповідну програму згідно нижче наведеного варіанта.

- Визначити всі дебетові сальдо, записати їх в одновимірний масив та вивести на екран.
- Визначити всі кредитові сальдо, записати їх в одновимірний масив та вивести на екран.
- Визначити всі дебетові сальдо, записати їх, знайти найбільше та вивести його на екран.
- Визначити всі кредитове сальдо, запам’ятати їх, знайти найменше та вивести його на екран.
- Визначити всі дебетові сальдо, записати їх, знайти найменше та вивести його на екран.
- Визначити всі кредитове сальдо, запам’ятати їх, знайти найбільше та вивести його на екран.
- Визначити, на скільки дебетове сальдо по всіх рахунках відрізняється від кредитового.
- Визначити, в якому місяці дебетове сальдо було найбільше.
- Визначити, в якому місяці дебетове сальдо було найменше.
- Визначити різницю між максимальним та мінімальним кредитовим сальдо.

## Завдання 8.5

Для завдання, наведеного нижче, розробити алгоритм і написати відповідну програму.

### Приклад 8.5

Дано тензор  $T(2 \times 3 \times 2) = \left\{ \begin{pmatrix} 1 & 0 & 2 \\ 3 & -1 & 1 \end{pmatrix}; \begin{pmatrix} 3 & 0 & 4 \\ 5 & 2 & 3 \end{pmatrix} \right\}$ . Знайти суму його

елементів, індекси яких парні.

### Розв'язання:



**Program l8z4;**

**Uses crt;**

**const n=3; m=2; z=2;**

**var a:array[1..n,1..m,1..z] of real;**

**i,j,k,c:integer; s:real;**

**begin**

**clrscr;**

**s:=0;**

*{Введення елементів тензора}*

**writeln('vvedit elementi masivu');**

**for i:=1 to n do begin**

**for j:=1 to m do begin**

**for k:=1 to z do begin**

**readln(a[i,j,k]);**

*{Визначення парних індексів}*

**if (i mod 2=0) and (j mod 2=0)**

**and (k mod 2=0) then**

*{Обчислення суми елементів}*

**s:=s+a[i,j,k];**

**end; end; end;**

**writeln('suma= ',s:3:2);**

**readln; end.**

### Варіанти завдань для самостійної роботи:

1. В тензорі  $T(2 \times 2 \times 3)$  знайти найбільший спільний дільник елементів тензора.

2. Дано інформацію про готель (скільки чоловік  $i$  в якому номері проживають), де  $i$  - номер корпусу,  $j$  – номер поверху,  $k$  – номер кімнати. Визначити, скільки порожніх номерів в  $n$  корпусі на  $z$  поверсі.
3. Дано  $3 - x$  вимірний векторний простір. Знайти скалярний добуток двох векторів.

4. Дано тензор  $T(2 \times 3 \times 2) = \left\{ \begin{pmatrix} 1 & 0 & 2 \\ 3 & -1 & 1 \end{pmatrix}, \begin{pmatrix} 3 & 0 & 4 \\ 5 & 2 & 3 \end{pmatrix} \right\}$ . Знайти квадрат найбільшого елемента.

5. Дано інформацію про готель (скільки чоловік  $i$  в якому номері проживають), де  $i$  - номер корпусу,  $j$  – номер поверху,  $k$  – номер кімнати. Визначити кількість проживаючих у номерах в  $n$  корпусі на  $z$  поверсі.

6. В тензорі  $T$  (див. приклад 4) знайти добуток елементів його діагоналей.
7. У матриці  $A(m, n)$  ( $m \leq 5$ ,  $n \leq 7$ ) знайти добуток елементів стовпця, в якому знаходиться максимальний елемент.
8. Дано інформацію про готель (скільки чоловік  $i$  в якому номері проживають), де  $i$  - номер корпусу,  $j$  – номер поверху,  $k$  – номер кімнати. Визначити різницю між кількістю зайнятих та порожніх номерів в  $n$ -му корпусі на  $z$ -му поверсі.
9. У тензорі  $T$  (див. завдання 4) знайти різницю між максимальним та мінімальним елементом.

10. Дано тензор  $T(2 \times 3 \times 2) = \left\{ \begin{pmatrix} -1 & 0 & 8 \\ 2 & -1 & 1 \end{pmatrix}, \begin{pmatrix} 3 & -5 & 1 \\ 7 & 2 & -3 \end{pmatrix} \right\}$ . Вивести окремо суму додатних та від'ємних елементів, підрахувавши при цьому їх кількість.

11. Дано тензор  $T(2 \times 3 \times 2) = \left\{ \begin{pmatrix} 5 & 0 & 6 \\ 3 & -1 & 1 \end{pmatrix}, \begin{pmatrix} 3 & 3 \\ -7 & 1 & 3 \end{pmatrix} \right\}$ . Знайти суму добутоків елементів кожного рядка тензора.

12. Дано тензор  $T(2 \times 3 \times 2) = \left\{ \begin{pmatrix} -1 & 0 & -8 \\ -5 & 3 & -2 \end{pmatrix}, \begin{pmatrix} 8 & 5 & 1 \\ 3 & 0 & -3 \end{pmatrix} \right\}$ . Віднести до квадрату всі елементи тензора.

13. Дано тензор  $T(2 \times 3 \times 2) = \left\{ \begin{pmatrix} 6 & 1 & 3 \\ 0 & -2 & -1 \end{pmatrix}, \begin{pmatrix} 9 & 5 & 3 \\ -5 & 0 & 3 \end{pmatrix} \right\}$ . Віднести до кубу елементи тензора, значення яких більші за 3.

14. Дано тензор  $T(2 \times 3 \times 2)$ . Знайти найбільший елемент та піднести його до кубу.

15. Провести таке перетворення тензора  $T(2 \times 3 \times 2) = \left\{ \begin{pmatrix} 5 & 1 & 0 \\ 8 & 2 & -2 \end{pmatrix}, \begin{pmatrix} 2 & 5 & 1 \\ -4 & 1 & -3 \end{pmatrix} \right\}$ , при якому останній стовпець займе місце першого, а всі інші змістяться на один стовпець вправо.


16. Дано тензор  $T(2 \times 3 \times 2)$ . Отримати новий тензор шляхом ділення всіх його елементів на найбільший по модулю елемент.

17. Знайти мінімальний елемент тензора

$$T(2 \times 3 \times 2) = \left\{ \begin{pmatrix} 10 & 1 & 0 \\ 6 & 2 & -2 \end{pmatrix}, \begin{pmatrix} 2 & 5 & 1 \\ -4 & 3 & -3 \end{pmatrix} \right\} \text{ та вивести його. Елементи}$$

тензора, що мають від'ємні значення, замінити максимальним елементом.

18. Знайти максимальний елемент тензора


$$T(2 \times 3 \times 2) = \left\{ \begin{pmatrix} 9 & 1 & 2 \\ 3 & 0 & -2 \end{pmatrix}, \begin{pmatrix} 2 & 5 & 1 \\ 4 & 3 & -5 \end{pmatrix} \right\} \text{ та вивести його.}$$

19. Елементи тензора, що мають додатні значення, замінити середнім арифметичним значенням елементів тензора.

20. Дано тензор  $T(3 \times 3 \times 2)$ . Визначити, скільки разів зустрічається найбільший елемент в тензорі.

21. Дано тензор  $T(3 \times 3 \times 2)$ . Визначити суму та добуток елементів тензора.

22. Дано тензор  $T(2 \times 3 \times 2) = \left\{ \begin{pmatrix} 8 & 5 & 3 \\ -2 & 2 & -2 \end{pmatrix}, \begin{pmatrix} 2 & -4 & 1 \\ 0 & 3 & -7 \end{pmatrix} \right\}$  Знайти найменший

елемент та замінити ним всі додатні елементи.

23. Дано інформацію про готель (скільки чоловік  $i$  в якому номері проживають), де  $i$  – номер корпусу,  $j$  – номер поверху,  $k$  – номер кімнати. Визначити різницю між кількістю зайнятих та порожніх номерів у готелі.

24. Провести перетворення тензора  $T(3 \times 3 \times 3)$  за формулами:

$$T_{ijk} = \frac{T_{ijk} \cdot z_{\max}^2}{\sqrt{z_{\min}}}, \text{ де } z_{\min} = \min\{T_{ijk}\}, z_{\max} = \max\{T_{ijk}\}, i = \overline{1,3}, j = \overline{1,3}, k = \overline{1,3}.$$

25. Дано тензор  $T(3 \times 3 \times 2)$ . Визначити добуток елементів, індекси яких парні.

### Контрольні запитання:

1. Дайте означення масиву.
2. Як здійснюється опис двовимірного масиву?
3. Чим характеризується кожен масив?
4. Як розміщуються елементи двовимірного масиву в пам'яті ПК?
5. Наведіть приклад введення, виведення двовимірного масиву.
6. Що розуміють під сортуванням масиву?
7. Які методи сортування масивів Ви знаєте?
8. Проаналізуйте алгоритм сортування прямим вибором та сортування прямим обміном.

9. Наведіть алгоритм сортування обміном на великих відстанях.
10. Який метод сортування називається стійким?

## Лабораторна робота №9

### Тема: Програми з використанням підпрограм



**Мета роботи:** Ознайомитися з особливостями роботи підпрограм, навчитися створювати підпрограми-процедури та процедури-функції, зробити висновки щодо доцільності використання підпрограм.

#### Хід роботи:

1. Розробити алгоритм обчислення деяких величин з використанням підпрограм згідно варіантів завдань для самостійної роботи.
2. Написати відповідні програми на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.

#### Завдання 9.1

За умовою завдання виконати обчислення, які доцільно оформити у вигляді нестандартних функцій, реалізувати їх засобами алгоритмічної мови та мови програмування.

#### Приклад 9.1

Обчислити величину  $S = \frac{\sqrt{\sum_{i=1}^5 t_i^2 + A^2}}{\sqrt{A^2 + B^2}} + \frac{\sqrt{B^2 + \sum_{i=1}^7 t_i^2}}{\sqrt{\sum_{i=1}^6 t_i^2 + C^2}}$ , де  $A=7,6$ ,

$B=-8,9$ ,  $C=3,65$ ;  $\{t_i\} = (5,2; -7,1; 8,3; -3,4; 7,5; -8,4; 6,0; 5,2; 1,9; 4,3)$ .

#### Розв'язання:

**Program** l9z1;

**Uses** crt;

**Const** t:array[1..10] of real =(5.2,-7.1,8.3,-3.4,7.5,-8.4,6.0,5.2,1.9,4.3);

    a=7.6;b=-8.9;c=3.65;

**Var** s,s1,s2,s3:real;

{Функція обчислення суми }

**Function** suma(p,k:integer):real;

**Var** s:real;  
i:integer;

**Begin**

**For** i:=p **to** k **do**

s:=s+t[i];

suma:=s;

**End;**

*{Функція обчислення кореня}*

**Function** korin (x,y:real):real;

**Begin**

korin:=sqrt(sqr(x)+sqr(y));

**End;**

**Begin**

**Clrscr;**

s1:=suma(1,5); s2:=suma(1,7); s1:=suma(1,6);

*{Обчислення S, в ролі ідентифікаторів виступають виклики функцій}*

s:=korin(s1,a)/korin(a,b)+korin(s2,b)/korin(s3,c);

**Writeln**('s= ',s:8:2);

**End.**

**Результат: s=7,53.**

**Варіанти завдань для самостійної роботи:**

$$1. B = \frac{\prod_{k=1}^5 (b_k - a_k^2)}{\cos^2(0,7-x)} - \frac{\cos^3(0,7-xy)}{\prod_{k=2}^5 (b_k^3 - a_k)} - \frac{\prod_{k=1}^5 (a_k - b_k)}{\cos^2(0,7-1,2y)}, \text{ де } x = 15,03 \cdot 10^{-2},$$

$y = 4,4$ , масиви  $\{a_k\}$ ,  $\{b_k\}$  із варіанта 21.

$$2. Q = \frac{\prod_{m=2}^7 (\sqrt[3]{a_m} + b^3_m)}{\ln|1 + xtg y|} + \frac{\ln|(1 + 2tg y)|}{\prod_{m=1}^5 (\sqrt[3]{a_m} + b^2_m)} + \frac{\prod_{m=4}^{10} (\sqrt[3]{a_m} + b^3_m)}{\lg|1 + tg 1,3|}, \text{ де}$$

$\{a_m\} = (1; 2,3; -1; -6,4; -2,3; 8,1; 3,3; 0,5; -0,1; 4,4)$ ,

$\{b_m\} = (0,5; -2,3; 3,4; 1,0; 9; -3,2; 1; 0,2; 1; -5,1)$ ,  $y=0,5$ ;  $x=1,5$ ;

$$3. B = \frac{\sqrt[3]{1 + \operatorname{tg} x^2}}{\sum_{m=2}^8 (3a_m + b^3 m)} - \frac{\sqrt[7]{1 + \operatorname{tg} y^2}}{\sum_{m=1}^5 (5a_m + b^5 m)} - \frac{\sum_{m=7}^{10} (a_m + b_m)}{\sqrt[5]{1 + \operatorname{tg} 1,44}},$$

де  $x=2,4$ ;  $y=-3,3$ ; масиви  $\{a_m\}, \{b_m\}$  - із варіанта 2

$$4. P = \frac{\prod_{k=1}^7 (\cos b_k - a^2 k)}{\ln|1,3 - x \operatorname{tg} z|} + \frac{\ln|x - 1,9 \operatorname{tg} x|}{\prod_{k=2}^5 (\cos a_k - b^2 k)} - \frac{\prod_{k=3}^7 (\cos a_k - a^2 k)}{\ln|zx - 3 \operatorname{tg} 1,2|},$$

де  $x=-3,5$ ,  $z=0,5$ ;  $\{a_k\} = (0,5; -3,3; 4,1; -1; -6,2; 2,2; 0,8)$ ,

$\{b_k\} = (-0,5; 2,4; 2,2; 1; 4,4; 2,1; -0,4)$ ;

$$5. Q = \frac{\sum_{i=1}^5 (a_i - b^3 i)}{\cos(xz - 1,7)} - \frac{\cos(3,6 - x^2)}{\sum_{i=2}^4 (b^2 i - a^3 i)} + \frac{\sum_{i=2}^5 (b^3 i - a^3 i)}{\cos(x - 0,3z)},$$

де  $x=6,4$ ,  $z=-0,9$ ;  $\{a_i\} = (-1,1; 2,4; 5,6; -2,4; 0,5)$ ,

$\{b_i\} = (1,4; 3,6; 1,3; -2,1; -3,3)$ ;

$$6. G = \frac{\log_3 |z - x|}{\prod_{k=2}^6 (a_k - \sin b_k)} + \frac{\prod_{k=1}^4 (b_k - \sin^2 a_k)}{\log_5 |1,3 - xz|} - \frac{\log_k |z - 1,3x|}{\prod_{k=3}^7 (a_k - \sin^3 a_k)},$$

де  $x=-4,3$ ;  $z=2,8$ ; масиви  $\{a_k\}, \{b_k\}$  - із варіанта 4.

$$7. S = \frac{\sqrt[3]{2 + \cos x^2}}{\sum_{i=1}^{10} (a_i + b_i)} - \frac{\sum_{i=2}^6 (a_i^2 + b_i)}{\sqrt[5]{3 + \cos y^2}} + \frac{\sqrt{x + \cos(yx)^2}}{\sum_{i=4}^9 (a_i^2 + b_i^2)},$$

де  $x=5,04$ ,  $y=-3,18$ ;  $\{a_i\} = (1,1; 2,4; 3,3; 5,4; -2,2; 2,2; -1; 0,5; -2; 0,4)$ ,

$\{b_i\} = (1,2; -3,4; -5; 0,1; 0,2; 6,1; 2; -3)$ .

$$8. Q = \frac{\cos(xz - 1,44)}{\sum_{m=1}^5 (a_m^2 + \cos^2 b_m)} + \frac{\sum_{m=3}^8 (a_m + \cos b_m)}{\cos(x - z^2)} - \frac{\cos(z - 0,81)}{\sum_{m=6}^{10} (a_m^3 + \cos^3 b_m)},$$

де  $x=0,017$ ;  $z=4,51$ ; масиви  $\{a_m\}, \{b_m\}$  - із варіанта 2.

$$9. R = \frac{\ln|1 + \cos^2 2^2|}{\prod_{k=6}^{10} (2b_k - \sqrt{a_k})} - \frac{\prod_{k=1}^3 (4b_k - 4\sqrt{a_k})}{\ln|1 + \cos^2 0,8|} - \frac{\ln|1 + \cos^2 49|}{\prod_{k=5}^9 (3a_k - \sqrt{b_k})},$$

де  $x = -0,49$ ;  $\{a_k\} = (0,5; 1,3; 4,4; 2,5; 0,2; 4,3; 3,4,8)$ ;  
 $\{b_k\} = (1,1; 2,3; 4,2; 1,6; 4,0; 1,1; 2,7; 5,4)$ .



Національний університет  
водного господарства  
та природокористування

$$10. Q = \frac{\sqrt[3]{2 + \cos x^2}}{\sum_{i=1}^7 (a_i^2 - b_i^3)} + \frac{\sum_{i=2}^5 (b_i^3 - a_i^3)}{\sqrt[5]{3 + \cos y^2}} + \frac{\sqrt[3]{x + \cos(yz)^2}}{\sum_{i=3}^8 (a_i^2 - b_i)}$$

де  $x = -0,94$ ;  $y = 2,61$ ;  $z = 1,8$ ;  $\{a_i\} = (0; 1; 1,1; 5,4; 6; 7,7; 1,8,8)$ ;  
 $\{b_i\} = (4,4; -3; 2,7; 0,9; 4,4; -5,3; -1,4; 2,2)$ .

$$11. D = \frac{\ln|\sin x + y|}{\prod_{i=1}^{10} (tga_i - b_i^2)} + \frac{\prod_{i=3}^7 (tgb_i - a_i^2)}{\ln|3 + \sin xy|} + \frac{\ln|x^2 y + \sin 3,1|}{\prod_{i=4}^8 (tgc_i - b_i^2)},$$

$\{a_i\} = (0,2; 0,4; 2,2; 1,2; 3,4; -2,4; -4,1; 1,1; -5,1)$ ,

$\{b_i\} = (1,5; -4,4; 5,4; 1,2; 0,0,5; 7,1; 7,4; 1,1)$ ;  $c_i = a_i + b_i$ ;  $i = \overline{1,10}$ ;  $x = 0,64$ ;  $y = 2,26$ .

$$12. S = \frac{\prod_{i=1}^{10} (\sin a_i - b_i)}{\ln|1 + xtgy|} + \frac{\prod_{i=1}^7 (\sin b_i - a_i)}{\ln|1 + tgx|} + \frac{\prod_{i=3}^9 (\sin a_i - b_i)}{\ln|1 + 2tg0,3|},$$

де  $x = 0,05$ ;  $y = 16,55$ ;

масиви  $\{a_i\}, \{b_i\}$  – із варіанта 7.

$$13. Q = \frac{\sum_{i=1}^7 (a_i^2 - b_i)}{\sqrt[3]{|1 + x \cos y^2|}} + \frac{\sqrt[5]{|1 + \cos x^2|}}{\sum_{i=2}^5 (a_i - b_i^2)} + \frac{\sum_{i=3}^9 (a_i^2 - b_i^2)}{\sqrt{|1 + 2 \cos y|}},$$

де  $x = 0,13$ ;  $y = 4,55$ ; масиви  $\{a_i\}, \{b_i\}$  – із варіанта 7.

$$14. Q = \frac{\sqrt[3]{2 + \cos x^2}}{\sum_{i=1}^{10} (a_i + b_i)} + \frac{\sum_{i=11}^{20} (a_i^2 + b_i^3)}{\sqrt[5]{3 + \cos y^2}} + \sqrt[3]{x + \cos(yz)^2},$$

де  $x = 14,51$ ;  $y = -6,2$ ;  $z = 2,15$ ;

$a_i = 0,4; b_i = 1,1$ ;  $a_i = (-1)^{i+1} (a_1 + i\Delta a)$ ;  $\Delta a = 0,2$ ,

$b_i = b_1 + i\Delta a; i = \overline{1,20}$ .



$$15. P = \frac{|a + b^2 \operatorname{tg} x|}{\prod_{i=1}^5 c_i^2 d_i} - |p - 1 + \operatorname{tg} y| \cdot \prod_{i=3}^7 c_i d_i^2 + \frac{\prod_{i=4}^{10} c_i^2 d_i^2}{|q^2 + \operatorname{tg} 0,5|},$$

де  $a=6,4; b=-1,45; p=4,03; q=2,55; c_i = \frac{i0,5}{3}; d_i = \frac{i-0,9}{4}; i = \overline{1,15}$ .

$$16. A = \frac{\sqrt[3]{|x-1,3y^3|}}{\prod_{m=2}^5 (a_m - b_m^2)} + \frac{\prod_{m=3}^6 (a_m^3 - b_m^2)}{\sqrt[3]{|y-3,6x|}} + \frac{\sqrt[4]{|xy-1,5|}}{\prod_{m=2}^7 (a_m^2 - b_m^3)},$$

де  $x=0,51 \cdot 10^{-3}; y=2,56 \cdot 10^{-2}$ ; масиви  $\{a_m\}, \{b_m\}$  - із варіанта 4.

$$17. R = \frac{\sin^2(x-y)}{\sum_{m=1}^5 (a_m - \sin|b_m|)} - \frac{\sum_{m=2}^7 (3a_m - \sin|b_m|)}{\sin^2(1,3-xy)} - \frac{\sin^2(1,3x-0,6)}{\sum_{m=1}^3 (b_m - \sin|a_m|)},$$

де  $x=-6,08; y=2,24$ ; масиви  $\{a_m\}, \{b_m\}$  - із варіанта 2 (перших 7 елементів).

$$18. S = \frac{\sum_{m=1}^5 (a_m + b_m^2)}{\cos(xz-1,7)} - \frac{\cos(3,6-x^2)}{\sum_{m=6}^9 (a_m + b_m)} + \frac{\sum_{m=3}^9 (a_m + b_m^2)}{\cos(x-0,37)},$$

де  $x=5,4 \cdot 10^{-2}; z=1,44$ ; масиви  $\{a_m\}, \{b_m\}$  - із варіанта 2 (перших 9 елементів).

$$19. G = \frac{\ln|1 + \cos^2 x^2|}{\prod_{k=1}^5 (a^2_k + b_k)} - \frac{\ln|1 + \cos^2 0,21|}{\prod_{k=6}^8 (a^2_k + b^2_k)} - \frac{\prod_{k=1}^8 (a_k + b^2_k)}{\ln|1 + \cos^2 z|},$$

де  $x=-0,89; z=-0,77; \{a_k\} = (0;1;-2;4;2;-3,5;2,1;4,4),$

$\{b_k\} = (0,4;-2,0;0;-4,4;3;1;6;-0,2).$

$$20. A = \frac{\log_3|a-x|}{\sum_{m=1}^5 (a_m^2 - b_m^3)} + \frac{\sum_{n=2}^4 (a_m^3 - b_m^4)}{\log_{3,7}|1,3-xa|} + \frac{\log_7|a-1,3x|}{\sum_{m=2}^5 (a_m - a_m^2)},$$

де  $a=4,28 \cdot 10^{-2}; x=12,06;$

$\{a_m\} = (-0,4;2,7;5;6,4;0,75), \{b_m\} = (2,8;-3,1;2,1;4,4;0,97).$

$$21. S = \frac{\sin^2(x-y)}{\prod_{k=2}^6 (a_k - \sin b_k)} + \frac{\prod_{k=1}^6 (b_k - \sin a_k)}{\sin^2(1,3-xy)} + \frac{\sin^2(1,3x-0,6)}{\prod_{k=3}^5 (a_k - \sin a_k)},$$

де  $x=4,44$ ;  $y=0,54$ ;  $\{a_k\} = (0,4; 8,8; 7,4; 12,5; 5; 0)$ ;

$\{b_k\} = (-4,1; 3,2; 0,7; -4,1; 2,2; 4,8)$ .

$$22. P = \frac{\sqrt[3]{0,3-xz}}{\sum_{i=1}^5 \sqrt[3]{a_i + b_i}} - \frac{\sum_{i=6}^{10} \sqrt{a_i + b_i}}{\sqrt[9]{23,1 - \sin x}} + \frac{\sqrt{|0,92-z|}}{\sum_{i=2}^7 \sqrt[3]{a_i + b_i}}, \text{ де } x=12,47; z=0,18;$$

масиви  $\{a_i\}, \{b_i\}$  – із варіанта 11.

$$23. S = \frac{\prod_{i=1}^3 (2a_i + b_i)}{\ln|1 + x \operatorname{tg} y|} + \frac{\ln|1 + \operatorname{tg} x|}{\prod_{i=3}^7 (a_i + 2b_i)} + \frac{\ln|1 + 2 \operatorname{tg} 0,3|}{\prod_{i=6}^{10} (3a_i + 2b_i)},$$

де  $x=-6,13$ ;  $y=25,4 \cdot 10^{-4}$ ; масиви  $\{a_i\}, \{b_i\}$  – із варіанта 11.

$$24. S = \frac{\prod_{i=1}^4 (\cos a_i + 2b_i)}{\operatorname{tg}|x^2 - y^2|} - \frac{\operatorname{tg}|xz - yx|}{\prod_{i=5}^{10} (\cos a_i + b_i)} - \frac{\prod_{i=3}^6 \cos a_i}{\operatorname{tg}|z^2 - yx|},$$

де  $x=-0,16 \cdot 10^2$ ;  $y=2,47$ ;  $z=1,05 \cdot 10^{-2}$ , масиви  $\{a_i\}, \{b_i\}$  – із варіанта 11.

$$25. P = \frac{\sum_{i=3}^7 (\cos 2a_i + b_i)}{\ln|1,3 - x \operatorname{tg} z|} + \frac{\ln|x - 1,9 \operatorname{tg} x|}{\sum_{i=6}^{10} (\cos 2a_i + 3b_i)} - \frac{\sum_{i=5}^8 (\cos a_i + \frac{b_i}{3})}{\ln|zx - 3 \operatorname{tg} 1,2|},$$

де  $x=24,41$ ;  $z=-12,46 \cdot 10^{-4}$ ; масиви  $\{a_i\}, \{b_i\}$  – з варіанта 11.

## Завдання 9.2

За умовою завдання виконати обчислення, які оформити у вигляді підпрограми, реалізувати їх засобами алгоритмічної мови програмування.

### Приклад 9.2

Користуючись формулами

$$x_c = \frac{\sum_{i=1}^n m_i x_i}{\sum_{i=1}^n m_i}, y_c = \frac{\sum_{i=1}^n m_i y_i}{\sum_{i=1}^n m_i}, z_c = \frac{\sum_{i=1}^n m_i z_i}{\sum_{i=1}^n m_i}, \text{ визначити і вивести на}$$

друк координати центра мас системи матеріальних точок

$M_i(x_i, y_i, z_i), i = \overline{1,6}$ , які задані масивом мас  $m(6) = (3; 6,3; 8,5; 5,2; 2; 1,5)$  і

масивами координат  $X(6) = (2; 3; 4; 1; 0; 5)$ ,  $Y(6) = (3; -4; 2; 7; -2; 0)$ ,  $Z(6) = (4; 3; 0; 2; -1; 7)$ .

**Розв'язання:**

**Program** 19z2;

**Uses** crt;

**Const** m:array[1..6] of real=(3,6.3,8.5,5.2,2,1.5);

  x:array[1..6] of real=(2,3,4,1,0,5);

  y:array[1..6] of real=(3,-4,2,7,-2,0);

  z:array[1..6] of real=(4,3,0,2,-1,7);

**Var** xc,yc,zc:real;

  {Процедура обчислення суми добутків}

**Procedure** suma(a:array of real;var ac:real);

**Var** i:integer;

      s1,s2:real;

      xc,yc,zc:real;

**Begin**

    s1:=0; s2:=0;

**For** i:=1 to 6 **do**

**Begin**

        s1:=s1+m[i]\*a[i];

        s2:=s2+m[i];

**End;**

    ac:=s1/s2; **End;**

**Begin**

**Clrscr;**

    suma(x,xc);

    suma(y,yc);

    suma(z,zc);

**Writeln**('M('xc:4:1,',',yc:4:1,',',zc:4:1,')');

**End.**

**Варіанти завдань для самостійної роботи:**

1. Обчислити значення виразу 
$$P = \frac{m!}{(m+n)!} - \frac{a^2 \sum_{i=1}^8 b_i^2}{(m-1)! - (\sum_{i=1}^6 b_i)^2}$$
, де  $a=3,68$ ,

$m=9, n=3, (b_i)_{i=1,8} = (3.5; 4.0; 2.5; 3.3; 2.1; 1.5; 7.3; 8.5)$ .



2. Обчислити імпульс  $P = \sum_{i=1}^n m_i v_i$  і кінетичну енергію  $T = \frac{1}{2} \sum_{i=1}^n m_i v_i^2$

системи  $n=10$  матеріальних точок, заданими масивами своїх мас та швидкостей  $M(10)=(3; 2; 5; 8.5; 4; 0.5; 7; 2; 9; 1)$ ,  $U(10)=(1.1; 2.5; 0.3; 8; 4; 0.5; 5; 3.2; 9; 1.5)$ .

3. Користуючись формулами

$$P_{kp} = \frac{\sum_{i=1}^n y_i P_i^{kp}}{\sum_{i=1}^n y_i}, \quad T_{kp} = \frac{\sum_{i=1}^n y_i T_i^{kp}}{\sum_{i=1}^n y_i},$$

обчислити і надрукувати критичну температуру  $T_{кр}$  і тиск  $P_{кр}$  газової суміші. Тут  $y_i, T_i^{kp}, P_i^{kp}$  - відповідно вміст, критична температура і критичний тиск  $i$ -того компонента вуглекислого газу в суміші, задані своїми масивами

$Y(9)=(4; 1; 8; 5; 6; 3; 7; 5; 4)$ ,  $T_i^{kp}(9)=(70; 60; 80; 65; 50; 45; 85; 44; 68)$ ;  
 $P_i^{kp}(9)=(0; 12; 15; 11; 14; 10; 13; 16; 19)$ .

4. Обчислити і надрукувати значення виразу

$$Z = \frac{(5n-4)!+8}{(2n+1)!} + \frac{n! \cdot \prod_{i=1}^7 x_i}{\prod_{i=1}^7 \sqrt[3]{x_i}}, \text{ де } n=3, \quad X(7)=(5; 6; 7; 3; 9; 8; 4).$$

5. Обчислити загальний опір 10 опорів у колі постійного струму з'єднаних: а) послідовно; б) паралельно користуючись формулами

$$R_{\text{посл}} = \sum_{i=1}^n R_i, \quad R_{\text{пар}} = \frac{1}{\sum_{i=1}^n \frac{1}{R_i}}, \text{ де } R(10)=(5; 6; 7; 8; 4; 3; 2; 5; 9; 12) \text{ (Ом)}.$$

6. Умова, що і в прикладі 5,  $R_2 = 2$  Ом, а решта утворюють арифметичну прогресію з різницею  $\square \square \square \square \square \square$  м.

7. Обчислити та надрукувати значення виразу

$$y = \sum_{k=1}^{10} \cos kx + \frac{\sqrt{(n!)^2 + \cos^2 x}}{\sqrt{\left(\sum_{k=1}^6 \cos kx\right)^2 + ((2n)!)^2}} + \sqrt[4]{((2n-1)!)^2 + 4}, \text{ де } x=8.31; n=4.$$

8. Обчислити ємність конденсаторів при:  
а) послідовному; б) паралельному з'єднаннях, користуючись формулами

$$C_{\text{посл}} = \frac{1}{\sum_{i=1}^n \frac{1}{C_i}}, \quad C_{\text{пар}} = \sum_{i=1}^n C_i, \quad \text{якщо } C(6)=(1.5; 1.1; 3.6; 5.4; 4.4; 2.1)$$

(Пф).

9. Обчислити та надрукувати значення виразу

$$W = \frac{ax + dy}{bx + cy}, \quad \text{де } X = \sum_{i=1}^7 x_i; \quad a = \sum_{i=1}^7 x_i y_i; \quad Y = \sum_{i=1}^7 y_i; \quad b = \prod_{i=1}^7 \frac{1}{x_i}; \quad d = \prod_{i=1}^7 \frac{x_i}{y_i};$$

с- довільне число.  $(X_i)=(1.1; 2.3; 4.5; 2.0; 4.2; 3.1; 1.8)$ ,  $(Y_i)=(-2.1; 3.0; 4.1; 2.5; 4.7; 6.8; 7.3)$ .

10. Обчислити та надрукувати середнє значення  $\bar{x}$  і середньоквадратичне відхилення результатів експерименту, користуючись формулами

$$\bar{x}_{\text{ср}} = \frac{1}{n} \sum_{i=1}^n x_i, \quad y = \sqrt{\frac{1}{n-1} \sum_{i=1}^n x_i^2 - \frac{n}{n-1} \bar{x}_{\text{ср}}^2}, \quad n=10, \quad X(10)=(20; 11; 10; 19; 22; 23; 11; 18; 14; 25).$$

11. Обчислити і надрукувати значення виразу

$$y = 1 - \frac{\sqrt[3]{\sum_{i=1}^7 t_i + a^2 + e^{\sqrt[3]{a+b^2}}}}{\sqrt[3]{b + \left(\sum_{i=1}^9 t_i\right)^2 + \sum_{i=1}^4 t_i}}, \quad \text{де } a=6.54; \quad b=4.85; \quad t_i - \text{ елементи масиву}$$

$$T(9) = (3; 4; 15; 2; 3; 4; 5; 4; 1).$$

12. Обчислити значення виразу

$$Q = \sum_{i=1}^6 a_i + \sum_{i=1}^6 (a_i^2 - b_i) + \sum_{i=1}^6 b_i a_i, \quad \text{де } (a_i)_{i=1,6} = (1.5; 1.3; 2.0; 3.6; 4.1; -1.6), \quad (b_i)_{i=1,6} = (2.1; 1.5; -1.0; 2.3; 4.3; 5.6).$$

13. Обчислити та надрукувати значення виразу

$$z = \frac{3}{4} \left( \sum_{i=1}^{10} \frac{\sqrt{a_i}}{a_{\min}^2} + \sqrt{\sum_{i=1}^{12} \frac{|b_i|}{b_{\min}^2}} \right), \text{ де } a_{\min}, b_{\min} - \text{ найменші значення}$$

елементів масивів  $(a_i)_{i=1,10} = (6; 5.6; 8.3; 7.4; 3.1; 4.6; 3.5; 1.4; 5.2; 8.3)$ ,  $(b_i)_{i=1,10} = (-8.5; 1.3; 4.9; -5.6; 6.7; -1.2; 7.8; 9.3; 14.5; 12.1)$ .

14. Визначити, які з пао векторів ортогональні між собою  $(a_i)_{i=1,5} = (1; 0; 3; 2; -1)$ ,  $(b_i)_{i=1,5} = (2; 3; 0; 1; 4)$ ,  
 $(c_i)_{i=1,5} = (-1; 2; 0; 0; 1)$ .

Вказівка. Два вектори  $x=(x_1, x_2, \dots, x_n)$ ,  $y=(y_1, y_2, \dots, y_n)$  називаються **ортогональними**, якщо їх скалярний добуток  $= 0$

$$x \perp y \iff (xy) = \sum_{i=1}^n x_i y_i = 0$$

15. Вершини п'ятикутника задано своїми координатами  $M_1(0;0)$ ,  $M_2(1;2)$ ,  $M_3(3;2)$ ,  $M_4(5;0)$ ,  $M_5(4;-2)$ . Обчислити площу п'ятикутника, скориставшись при цьому формулою Герона для обчислення площі трикутника:  $S\Delta = \sqrt{p(p-a)(p-b)(p-c)}$ , де  $p = \frac{a+b+c}{2}$ ; a, b, c- довжини сторін.

16. Обчислити та надрукувати значення виразу

$$y = \sqrt{i^2 + j^2} \left( \frac{i!}{\sqrt{i^2 + 1.3^2}} + \frac{\cos i}{\sqrt{(j!)^2 + i^2}} \right), \text{ де } i=6, j=5.$$

17. Обчислити і вивести на друк значення виразу  $z = x_1 + x_2 + x_3$ ,

$$\text{де } x_1 = \frac{\sum_{i=1}^8 a_i}{10!}; \quad x_2 = \frac{\sum_{i=1}^7 b_i}{8!}; \quad x_3 = \frac{\sum_{i=3}^8 c_i}{7!};$$

$$(a_i)_{i=1,8} = (5.0; 8.3; -4.5; 6.8; 3.4; 0; -4.2; 5.9),$$

$$(b_i)_{i=1,8} = (3.5; -5.4; 6.2; 7.0; -5.6; 4.9; 2.8; 7.4),$$

$$(c_i)_{i=1,8} = (4.2; -8.6; 5.5; -7.8; 2.7; 0.3; -3.0; 2.8).$$

18. Вектори задані своїми координатами

$$(a_i)_{i=1,3} = (1.5; 2.1; 0.3), \quad (b_i)_{i=1,3} = (3; 0.2; 0.7),$$

$$(c_i)_{i=1,3} = (0.7; -2.5; 1.2), \quad (d_i)_{i=1,3} = (0.4; 8.5; -1.2),$$

$$(f_i)_{i=1,3} = (0.1; -0.3; 1.3), \quad (e_i)_{i=1,3} = (0.4; 1.3; 0.6).$$

Знайти вектор з найменшою довжиною. Надрукувати його координати і довжину.

19. Обчислити та надрукувати значення виразу

$$A = \sqrt{\frac{(2m)!}{(2m-n)!}} + 3\sqrt[3]{\frac{(5m-2n)!}{(n+m)!}}, \text{ де } m=4, n=4.$$

20. Населені пункти задані своїми координатами M1(-1;1), M2(1.5; 2), M3(6; 4), M4(3;4), M5(1;5). Знайти, який населений пункт найближче розміщений до залізничної колії, що задається рівнянням  $3x-4y+5=0$ .

Вказівка. При обчисленні відстані від заданого пункту до залізниці користатись формулою  $\alpha = \frac{|ax+by+c|}{\sqrt{a^2+b^2}}$ .

21. Обчислити та надрукувати значення виразу

$$y = \left( \frac{ab + \sqrt{\left(\prod_{i=1}^5 x_i\right)^2 + a^2}}{\sqrt[4]{b^2 + \left(\prod_{i=1}^7 x_i\right)^2}} + \sin \sqrt{a^2 + b^2} \right) e^{-\sqrt{1+b^2}}, \text{ де } a=7.63;$$

$$b=-4.85, x_i=(-3.6; 7.8; -4.2; 3.5; -0.6; 2.8; 4.2).$$

22. Обчислити значення виразу  $u = t^{x_1+y_1} - e^{x_2-y_2}$ , де  $x_1, x_2, y_1, y_2$  корені рівнянь:  $3x^2 - 6x + 1 = 0$ ,  $2y^2 - y + 4 = 0$ . Причому  $x_1 \leq x_2, y_1 \leq y_2$ , якщо корені дійсні. У випадку комплексних коренів - рахувати їх рівними нулю.

23. Обчислити та надрукувати значення виразу  $W = \begin{cases} aX + Y, & p > 0 \\ bX - Y, & p \leq 0 \end{cases}$ , якщо

$$W = \begin{cases} aX + Y, & p > 0 \\ bX - Y, & p \leq 0 \end{cases}, \text{ де } p = \frac{a+b}{X-Y}; a = \prod_{i=1}^6 \frac{1}{b_i}; b = \prod_{i=1}^6 \frac{1}{b_i}; X = \sum_{i=1}^6 |a_i|;$$

$$Y = \sum_{i=1}^6 |b_i|; (a_i)_{i=1,6} = (-2.3; 6.2; 5.8; -3.4; 7.1; 0.05),$$

$$(b_i)_{i=1,6} = (3.0; -2.3; 4.1; 2.5; 6.8; 4.5).$$

24. Обчислити та надрукувати добове число  $n_{\text{аіа}}$  ввімкнень насоса підкачки, використовуючи формулу

$$n_{\text{доб}} = \left[ \frac{Q_{c.i.}}{W_p} \left( \sum_{i=1}^n R_i - \sum_{i=1}^n R_i^2 \right) \right], \text{ де } Q_{c.i.} - \text{ середня продуктивність насоса}$$

підкачки,  $\frac{M^3}{\text{год}}$ ;  $W_p$  - регулюючий об'єм водоповітряного бака;  $(R_i)_{i=1,n}$  масив чисел, при чому  $0.3 \leq R_i \leq 3$ ;  $n$  - число годин, на протязі яких вода поступає в сітку.  $Q_{c.n.} = 5.4$ ;  $W_p = 180$ ;  $n = 8$ ;  $(R_i)_{i=1,8} = (0.4; 0.5; 2.1; 3.0; 2.6; 1.4; 1.6; 1.8)$ .

25. Обчислити та надрукувати значення виразу  $y = \frac{Ax - B}{Cx + D}$ ,

$$\text{де } A = \sum_{i=1}^7 a_i; C = \sum_{i=1}^8 c_i; B = \prod_{i=1}^4 (a_i^2 - a_i) \quad D = \prod_{i=1}^5 c_i; x = 5.83;$$

$$(a_i)_{i=1,7} = (3; 1.5; 4.2; 3.4; 6.8; 2; 1.3), (c_i)_{i=1,7} = (6.5; 3.0; 4.1; 2.3; 1.1; 0.5; 7.3).$$



Національний університет  
водного господарства  
та природокористування

### Контрольні запитання:

1. Дайте означення підпрограми.
2. Як класифікують підпрограми в ТР?
3. Дайте означення підпрограми-процедури та підпрограми-функції.
4. Зобразіть структуру опису підпрограми-процедури та підпрограми-функції
5. Що розуміють під поняттями формальні та фактичні параметри?
6. Що таке глобальні та локальні змінні?
7. Що розуміють під вхідними та вихідними параметрами?
8. Скільки значень повертає підпрограма-процедура та підпрограма функція?
9. З якою метою використовують рекурсію в підпрограмах?
10. Яка доцільність використання підпрограм у програмах?



Національний університет  
водного господарства  
та природокористування

### Лабораторна робота №10

#### Тема: Робота з текстовими змінними

**Мета роботи:** Ознайомитися з основними функціями та стандартними процедурами обробки стрічок та навчитися застосовувати їх для написання програм.

#### Хід роботи:

1. Розробити алгоритми обробки текстового масиву згідно варіантів завдань для самостійної роботи.
2. Написати відповідні програми на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.



## Завдання 10.1

### Приклад 10.1

Дано текстовий масив A(10). Знайти і надрукувати елементи найменшої довжини. Вивести на друк даний елемент, його порядковий номер і довжину.

#### Розв'язання:

```
Program l10z1;  
Uses crt;  
Var a:array [1..10] of string;  
    i,n,d:integer;  
  
Begin  
    Clrscr;  
    Writeln('Введіть 10 слів');  
    For i:=1 to 10 do          {Введення елементів масиву A}  
        Readln(a[i]);  
    n:=1;  
    d:=length(a[1]);  
    For i:=2 to 10 do        {Відшукування елемента мінімальної довжини}  
        If length(a[i])<d then begin  
            n:=i;  
            d:=length(a[i]);  
        End;  
    Writeln(a[n], '-- елемент мінімальної довжини, його довжина ',d,  
    порядковий номер ',n);  
End.
```

#### Варіанти завдань для самостійної роботи:

1. В текстовому масиві A(13) поміняти місцями: 1-й елемент з 13-м, 2-й з 12-м, і т. д. Вивести на друк початковий та перетворений масиви.
2. Дано текстовий масив: 'папір', 'вода', 'башта', 'канал', 'висота', 'об'єм'. Злити 2-й і 4-й елементи масиву і отриману текстову змінну поставити на друге місце. 4-й елемент масиву знищити.
3. Дано текстовий масив A(12). Відсортувати його в порядку зростання довжин його елементів і записати отриманий масив у масив B(12).
4. З елементів текстового масиву B(20) сформувати масиви, елементи яких мають однакову довжину.
5. В текстовому масиві C(15) знайти суму довжин елементів з найменшою та найбільшою довжиною.

6. В текстовому масиві A(15) поміняти місцями елементи з найменшою та найбільшою довжинами.
7. В текстовому масиві B(12) відшукати елемент з найбільшою довжиною, вивести його на друк разом з номером і довжиною.
8. Дано текстовий масив B(12). Відсортувати його в порядку спадання довжин його елементів.
9. Дано текстовий масив A(10). Знайти і надрукувати елемент найменшої довжини. Вивести на друк даний елемент, його порядковий номер і довжину (кількість символів).
10. Дано масив A(10). Вивести на друк елементи в порядку зростання їх довжини.
11. Масив B(10) містить прізвища студентів. Впорядкувати його в алфавітному порядку.
12. Дано числовий масив оцінок: 3, 4, 4, 5, 2, 3, 3, 4. Сформувати текстовий масив оцінок, замінивши: 3 – на задовільно, 4 – на добре, і т. д. Надрукувати отриманий масив.
13. Дано результати екзаменаційної сесії:

№	П. І. Б.	Вища математика	Фізика	Чисельні методи
1.	Іваненко А.М.	задовільно	добре	незадовільно
2.	Петренко О.О.	добре	незадовільно	добре
3.	Сидорук К.С.	відмінно	задовільно	добре
4.	Крук П.В.	відмінно	відмінно	відмінно
5.	Голуб О.В.	задовільно	добре	добре

Підрахувати середній бал студента по кожному предмету.

14. Умова та ж, що і в 13. Підрахувати середній бал кожного студента.
15. За результатами екзаменаційної сесії (завдання 13) визначити кількість 3, 4, 5.
16. Впорядкувати елементи текстового масиву A(12) в алфавітному порядку.
17. Дано текстовий масив A(10). Вивести на друк його елементи в спадаючому порядку їх довжин.
18. В текстовому масиві, що містить 8 слів, підрахувати суму довжин елементів, що стоять на парних місцях.
19. В текстовому масиві F(10) підрахувати суму довжин перших 7-ми елементів.
20. В текстовому масиві з 9-ти елементів знайти суму довжин елементів з 2-го по 6-й.

21. Дано текстовий масив В(12). Відсортувати його в порядку спадання довжин його елементів і записати отриманий масив у масив А(12).
22. Дано масив текстових змінних В(10). Створити масив С(10), що містить елементи масиву В(10), записані у зворотному порядку.
23. В текстовому масиві А(10) підрахувати суму довжин останніх 5-ти елементів.
24. В текстовому масиві С(15) знайти різницю довжин елементів з найбільшою та найменшою довжиною.
25. З текстового масиву А(10) видалити елементи найбільшої та найменшої довжин. Результат записати у новий масив.

## Завдання 10.2

Скласти програму обробки текстових величин згідно вказаної умови. Передбачити друк початкового та результуючого текстів.

### Приклад 10.2

Дано текстові величини: “дуб”, “кок”, “тет”, “піп”, “рід”.  
Надрукувати ті слова, які є симетричними.

**Розв'язання:**

```

Program l10z2;
Uses crt;
Label m1;
Const a:array[1..5] of string=('дуб','кок','тет','піп','рід');
Var i,j,k:integer;
  Begin
    Clrscr;
    For j:=1 to 5 do
      Begin
        k:=length(a[j]); {Визначення довжини слова}
        For i:=1 to k do
          Begin {Умова перевірки симетричності слова}
            If a[j][k]<>a[j][i] then goto m1;
            k:=k-1; End;
          Writeln(a[j]);
        m1:end;
      End.
  
```

**Результат:** кок, тет, піп.

### **Варіанти завдань для самостійної роботи:**

1. Є текст F, що містить літери, цифри. Замінити цифри пробілами, в кінці поставити крапку.
2. Є текст, що містить літери, цифри, пробіли. Необхідно виділити слова, викинути з них цифри і сформувати з них масив C.
3. Є текст, до складу якого входять літери, цифри, пробіли та коми. Порахувати кількість слів (набір символів розділених комами, або пробілами) та роздрукувати в стовпчик.
4. Є текстова змінна F, до складу якої входять літери, цифри, пробіли. Необхідно утворити нову змінну, до складу якої входять лише букви та пробіли (викинути всі цифри). Роздрукувати початкову та результуючу змінну.
5. Є текст F, який містить слова, розділені комами. Необхідно коми замінити пробілами і роздрукувати отриманий текст.
6. У тексті F, який містить слова, розділені комами або пробілами, поміняти місцями перше слово з останнім, друге - з передостаннім, і т.д.
7. З тексту A вибрати і роздрукувати лише ті слова, які починаються з букви K.
8. У тексті G елементи розбити на склади і записати зі знаками тире у новий масив (складом вважати елемент слова, що має кількість букв > 1 і закінчується голосною, наприклад: абракадабра => абра-ка-да-бра).
9. Дана деяка текстова змінна T. Переставити її букви в алфавітному порядку.
10. Дано слово A. Прочитати його у прямому і зворотному порядку. Вивести "ТАК" або "НІ", залежно від того, співпадає чи ні результат читання символічної константи A.
11. В символічній змінній A розмістити букви в порядку, зворотньому до алфавітного.
12. Заданий текстовий рядок T ="я студент першого курсу". Створити рядок N, який буде включати останні два слова рядка T.
13. З рядка T ="ой на, ой на горі там жінці жнуть" вилучити слово "там" і всі пробіли. Результат записати в N.
14. Змінна T ="а по-під горою\_яром\_долиною\_козаки\_йдуть" поміняти символ "\_" на " " (пробіл). Результат занести в змінну P і підрахувати кількість пробілів у новій змінній P.
15. Дана текстова змінна: T= "Ой в лісі, лісі темному, де ходить хитрий лис". Підрахувати кількість "," (ком) у змінній та замінити всі слова "лісі" на "полі". Результати заміни записати в масив N.
16. Дана текстова змінна: T ="ой тяжко в світі жити". Замінити слово "тяжко" на "важко" і після 4 слова вставити слово "білім". Результат записати в K.

17. Дана деяка текстова змінна K. Переставити її перші чотири літери в алфавітному порядку.
18. З тексту E вибрати і роздрукувати лише ті слова, які закінчуються на літеру "а".
19. Є текст, що містить літери, цифри, пробіли. Необхідно виділити слова на основі кожного другого слова та сформувати масив C.
20. Є текст F, який містить слова, розділені комами. Необхідно коми замінити набором символів "====" і роздрукувати отриманий текст.
21. З тексту E вивести слова з подвоєнням літери „н”, визначити кількість слів з подвоєннями.
22. Є текст G. Підрахувати кількість літер "а", „о".
23. В реченні F всі слова починаються з великої літери. Замінити всі великі літери на малі, крім першої літери в слові.
24. Дана текстова змінна: T= "Ой в лісі, лісі темному, де ходить хитрий лис". Підрахувати кількість голосних звуків.
25. Записати деяку текстову змінну K у зворотному порядку.

### **Завдання 10.3**

Скласти програму обробки текстових величин згідно вказаної умови. Передбачити друк початкового та результуючого текстів.

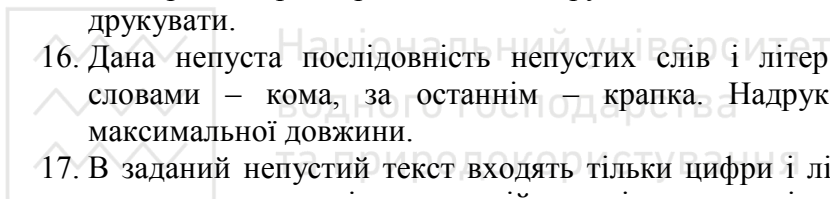
#### **Варіанти завдань для самостійної роботи:**

1. Надрукувати заданий текст, видаливши з нього зайві пробіли, тобто із декілька пробілів, що йдуть підряд, залишити один.
2. Надрукувати в алфавітному порядку всі різні маленькі літери кирилиці, що входять в даний текст із 200 літер.
3. В заданий непорожній текст входять тільки цифри, літери і коми. Визначити, чи задовольняє він наступну умову: текст складається тільки із цифр, при чому їх числові значення утворюють арифметичну прогресію (наприклад, 11, 9, 7).
4. В заданий непустий текст входять лише цифри і літери. Визначити, чи задовольняє даний текст такій вимозі: текст починається з  $k$  літер ( $1 \leq k \leq 9$ ), за яким слідує лише одна літера – цифра з числовим значенням  $k$ .
5. Дано послідовність, яка містить від 1 до 30 слів, в кожному з яких від 1 до 5 латинських літер, між словами – коми, за останнім словом – крапка. Надрукувати слова, в яких більше букв 'а'.
6. Використовуючи лише літерний ввід, тобто процедуру Read(c), де c – літерна змінна, ввести непусту послідовність цифр, перед якою може знаходитись знак „+” або „-”, за якою слідує пропуск, і, отримавши відповідне ціле число, присвоїти його цілій змінній k.

7. Дано непусту послідовність слів, що містять від 1 до 8 літер, між сусідніми словами – кома, після останнього слова – крапка. Надрукувати всі слова найменшої довжини.
8. Надрукувати даний текст, видаливши з нього всі голосні.
9. Заданий текст роздрукувати по рядках, розуміючи під рядком або чергові 60 літер, якщо серед них немає коми, або частину тексту до коми включно.
10. В даний непорожній текст входять тільки цифри і букви. Визначити, чи задовольняє він наступній умові: текст є десятковим записом числа, кратного 9.
11. Дано два різних слова, в кожному з яких від 1 до 8 малих латинських букв, за кожним з яких – пробіл. Надрукувати ці слова, впорядкувавши літери слів в алфавітному порядку.
12. Надрукувати TRUE, якщо в заданому тексті буква *a* зустрічається частіше, ніж буква *b*, і надрукувати FALSE в іншому випадку.
13. В заданий непорожній текст входять тільки цифри і літери. Визначити, чи задовольняє він наступну властивість: текст є шістнадцятковим записом, кратного 5.
14. В заданий непорожній текст входять тільки цифри і літери. Визначити, чи задовольняє він слідує властивість: текст є записом парного числа в сімковій системі числення.
15. Використовуючи тільки літерний вивід, тобто процедуру Write(c), де *c* – літерний параметр. Вивести на друк значення цілої змінної *k*, „+” не друкувати.
16. Дана непуста послідовність непустих слів і літер. Між сусідніми словами – кома, за останнім – крапка. Надрукувати всі слова максимальної довжини.
17. В заданий непустий текст входять тільки цифри і літери. Визначити, чи задовольняє він наступній умові: текст співпадає з кінцевим відрізком ряду 0123456789 (наприклад, 9, 89, 789).
18. В заданий непустий текст входять тільки цифри і літери. Визначити, чи задовольняє він таку умову: текст містить, крім літер, тільки одну цифру, причому її числове значення рівне довжині тексту.
19. З’ясувати, чи симетричний, заданий у вхідному файлі текст (за ним слідує крапка).
20. Задано текст з малих латинських літер, за яким слідує крапка. Надрукувати в алфавітному порядку всі букви, які входять в даний текст по одному разу.
21. З заданий непустий текст входять тільки цифри і букви. Визначити, чи задовольняє він наступній умові: текст починається з деякої ненулевої цифри, за якою слідує тільки літери, і їх кількість рівна числовому значенню цієї цифри.



Національний університет  
водного транспорту  
та морської інженерії



22. Дано непорожню послідовність слів, які містять від 1 до 8 букв, між сусідніми словами – кома, за останнім словом – крапка. Надрукувати всі слова найменшої довжини.
23. Дано непусту послідовність слів із латинських літер; сусідні слова розділені один від одного комою, за останнім – крапка. Визначити кількість слів, які закінчуються літерою *w*.
24. Дана послідовність, яка містить від 1 до 30 слів, в кожному з яких від 1 до 5 латинських літер; між сусідніми словами – кома, за останнім словом – крапка. Надрукувати цю ж послідовність слів, але в оберненому порядку.
25. Дано послідовність, яка складається від 1 до 30 слів, в кожному із яких від 1 до 5 малих латинських літер, між сусідніми словами – кома, за останнім – крапка. Надрукувати всі слова, які зустрічаються в послідовності більше одного разу.

### **Контрольні запитання:**

1. Які типи даних призначені для збереження символної інформації?
2. Як здійснюється опис стрічкового типу в ТР?
3. Наведіть відомі Вам функції перетворення типів?
4. Які дії визначено над стрічками?
5. Дайте означення символного масиву. Як він описується в ТР?
6. Наведіть стандартні функції для роботи зі стрічками.
7. Наведіть стандартні процедури для роботи зі стрічками.
8. Які значення може приймати символна змінна та які операції можна над нею виконувати?
9. Яка максимальна довжина змінної стрічкового типу?
10. Як визначити довжину змінної стрічкового типу?

## **Лабораторна робота №11**

### **Тема: Використання записів**

**Мета роботи:** Ознайомитися із записами, вкладеними записами. Навчитися працювати з записами, масивами записів.

### **Хід роботи:**

1. Розробити алгоритми обробки даних про сесію з використанням записів згідно варіантів для самостійної роботи.

2. Написати відповідні програми на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.

### Завдання 11.1

Інформацію про результати екзаменаційної сесії групи студентів 1-ого курсу спеціальності «Прикладна математика» подано у вигляді таблиці:



Національний університет  
водного господарства  
та природокористування

№	Прізвище	Математичний аналіз	Програмування	Операційні системи	Алгебра та геометрія
1.	Берестяк Г.С.	4	3	3	4
2.	Заєць О.М.	5	4	4	5
3.	Зоренко П.М.	3	4	4	4
4.	Іванчук С.О.	4	3	3	3
5.	Кравченко З.І.	2	3	3	2
6.	Панченко І.А.	3	5	4	5
7.	Сидорук В.Р.	5	4	4	3
8.	Снопко П.О.	4	2	3	3
9.	Якимець Н.С.	4	5	5	5
10	Якубів Р.Н.	5	5	4	4

Інформацію про кожного студента оформити в програмі у вигляді запису. Сукупність записів об'єднати в масив.

Скласти програму, яка забезпечує введення отриманої інформації, виведення її у вигляді таблиці, а також виведення інформації згідно конкретного варіанта.

### Приклад 11.1

Надрукувати таблицю, що містить номери, прізвища та кількість оцінок "3", "2", "4", "5" у кожного студента та підрахувати загальну кількість.

**Розв'язання:**

**Program** l11z1;

**Uses**

Crt;

**Type**

ff=record



```
pr:string;  
m,p,o,a:integer;
```

**End;**

*{Вигляд масиву типу запис у типізованій константі}*

```
Const t:array[1..10] of ff=((pr:'Берестяк';m:4;p:3;o:3;a:4),  
(pr:'Заець';m:3;p:4;o:4;a:4),(pr:'Зоренко';m:3;p:4;o:4;a:4),  
(pr:'Іванчук';m:4;p:3;o:3;a:3),(pr:'Кравченко';m:2;p:2;o:3;a:2),  
(pr:'Панченко';m:5;p:5;o:4;a:4), (pr:'Сидорук';m:3;p:5;o:4;a:5),  
(pr:'Снопок';m:5;p:4;o:4;a:3),(pr:'Якимець';m:4;p:2;o:3;a:3),  
(pr:'Якубів';m:4;p:5;o:5;a:5));
```

**Var**

```
ch,o,h,z,p,l,v,g,q,tr,kl:integer;  
w:string;
```

*{Функція обчислення кількості оцінки у i-го студента}*

```
Function k(pr,o:integer):integer;
```

**Var**

```
j:integer;
```

**Begin**

```
j:=0;  
if t[pr].m=0 then j:=j+1;  
if t[pr].p=0 then j:=j+1;  
if t[pr].o=0 then j:=j+1;  
if t[pr].a=0 then j:=j+1;
```

```
k:=j;
```

**End;**

**Begin**

```
Clrscr;
```

```
Textcolor(13);
```

```
Writeln (' Студент мат.ан прог. ОС алгебра');
```

*{Вивід таблиці-умови}*

```
For q:=1 to 10 do
```

```
Writeln(t[q].pr:12,t[q].m:8,t[q].p:8,t[q].o:8,t[q].a:8);
```

```
Writeln; Writeln;
```

```
Write (' Студент к-ть 2 к-ть 3 к-ть 4 к-ть 5');
```

*{Створення таблиці-відповіді}*

```
For q:=1 to 10 do
```

```
Begin
```

```
Writeln; Write(t[q].pr:12);
```

```
For kl:=2 to 5 do
```

```
Begin
```

```
h:=k(q,kl);
```

```
Write (h:8);
```

```
Case kl of {Визначення загальної кількості оцінок}
```

```
2:z:=z+h;  
3:tr:=tr+h;  
4:ch:=ch+h;  
5:p:=p+h;
```

**End;**

**End;**

**End;**

**Writeln; Textcolor(22);**

**w:='Всього';**

**Write(w:12,z:8,tr:8,ch:8,p:8);**

**Readln;**

**End.**

### ***Варіанти завдань для самостійної роботи:***

1. Надрукувати таблицю, що містить номери, прізвища та кількість “4”, “5”, “3”, “2” у кожного студента групи, а також підрахувати середній бал всіх студентів групи.
2. Надрукувати таблицю, що містить номери, прізвища, оцінки та середній бал тих студентів групи, середній бал яких більше 4, а також підрахувати кількість таких студентів у групі.
3. Надрукувати таблицю, що містить номери, прізвища та оцінки студентів, які мають хоча б одну “3”, а також підрахувати кількість таких студентів у групі.
4. Надрукувати таблицю, що містить номери, прізвища та оцінки студентів, які не мають жодної “5”. Підрахувати кількість таких студентів.
5. Надрукувати таблицю, що містить номери, прізвища та оцінки кожного студента, а в кінці вказати середній бал групи з кожної дисципліни.
6. Надрукувати таблицю, що містить номери, прізвища, оцінки та середній бал кожного студента групи.
7. Надрукувати таблицю, що містить номери, прізвища та оцінку студентів з математичного аналізу, а також підрахувати середній бал групи з цього предмета.
8. Надрукувати таблицю, що містить прізвища та оцінки тих студентів, які мають найбільший та найменший середній бал у групі.
9. Надрукувати таблицю, що містить номери, прізвища, оцінки та середній бал студентів групи, середній бал яких менше 4.
10. Надрукувати таблицю, що містить номери, прізвища, оцінки студентів, які мають тільки добрі та відмінні оцінки.
11. Надрукувати таблицю, що містить номери, прізвища, оцінки та кількість “3” в оцінках кожного студента.

12. Надрукувати таблицю, що містить номери, прізвища та оцінки тих студентів, які отримали з програмування добрі та відмінні оцінки, а також підрахувати кількість таких студентів.
13. Надрукувати таблицю, що містить номери, прізвища та оцінки тих студентів, які отримали з алгебри та геометрії задовільну або незадовільну оцінку, а також підрахувати кількість таких студентів.
14. Надрукувати таблицю, що містить номери, прізвища та екзаменаційні оцінки студентів. В кінці вказати дисципліну, середній бал якої максимальний.
15. Надрукувати таблицю, що містить номери, прізвища та оцінки студентів, які отримали хоча б одну незадовільну оцінку.
16. Надрукувати кількість “2”, “3”, “4”, “5” з кожної дисципліни.
17. Надрукувати таблицю, що містить номери, прізвища і кількість “2”, “3”, “4”, “5” в оцінках кожного студента.
18. Надрукувати таблицю, що містить номери, прізвища і оцінки студентів з предметів “Математичний аналіз” і “ОС”.
19. Надрукувати таблицю, яка містить середні екзаменаційні бали кожного студента.
20. Надрукувати таблицю, що містить номери, прізвища, оцінки та кількість позитивних оцінок кожного студента.
21. Надрукувати таблицю, яка містить середні екзаменаційні бали групи по кожному предмету.
22. Надрукувати таблицю, яка містить прізвища студентів, які не склали жодного іспиту.
23. Надрукувати таблицю, що містить номери, прізвища та екзаменаційні оцінки студентів. В кінці вказати дисципліну, середній бал якої мінімальний.
24. Надрукувати таблицю, що містить номери, прізвища, оцінки та середній бал студентів групи, середній бал яких не менший 4,5.
25. Надрукувати таблицю, що містить номери, прізвища, оцінки та середній бал кожного студента групи. Середній бал впорядкувати за спаданням.

### Завдання 11.2

Задано масив даних наступної структури:

Прізвище; Рік народження; Адреса;  
 Чол.: 1) військовий,           2) заробітна плата;  
 Жін.: 1) кількість дітей,   2) чи заміжня,  
           3) заробітна плата.

Скласти програму з використанням записів з варіантами (вкладених записів), яка забезпечує введення отриманої інформації, виведення її у вигляді таблиці, а також виведення інформації згідно конкретного варіанту.

## Приклад 11.2

Знайти середню заробітню плату заміжньої жінки з кількістю дітей менше 3.

**Розв'язання:**

**Program** l11z2;

**Uses** crt;

**Const** n=5;

**Type**

pol=(wom,men);

rr=record

fam,im,adres:string;

pik:integer;

stat:string;

**Case** stat:pol of *{Запис з варіантом}*

wom:(dit,zap:integer;zam:string);

men:(zar:integer;viysk:string);

**End;**

**Var** xap:array [1..n] of rr;

i:integer;

s,min,max,riz:real;

**Begin**

**Clrscr;**

**For** i:=1 to n **do begin** *{Створення анкети}*

**Clrscr;**

**Writeln** (' Введіть прізвище ',i,' особи '); **Readln**(xap[i].fam);

**Writeln** (' Ім'я '); **Readln** (xap[i].im);

**Writeln**(' Рік народження '); **Readln**(xap[i].pik);

**Writeln**(' Адреса '); **Readln** (xap[i].adres);

**Writeln**(' Стать (чол/жін) ; **Readln**(xap[i].stat);

**If** xap[i].stat='чол' **then**

**Begin**

**Writeln** (' Військовий (так/ні) '); **Readln** (xap[i].viysk);

**Writeln** (' Його зарплата '); **Readln** (xap[i].zar);

**End;**

**If** xap[i].stat='жін' **then begin**

**Writeln** (' Кількість дітей'); **Readln**(xap[i].dit);

**Writeln** ('Чи заміжня (так/ні)') **Readln**(xap[i].zam);

**Writeln** ('І зарплата'); **Readln** (xap[i].zap);

**End;**

**End;**

```

For i:=1 to n do begin           {Вивід за умовою}
  If (xap[i].stat='жін') and (xap[i].dit>3) then begin
    Writeln(xap[i].fam); Writeln(xap[i].im);
    Writeln(xap[i].pik); Writeln(xap[i].adres);
    Writeln(xap[i].dit); Writeln(xap[i].stat);
    Writeln(xap[i].zap); Writeln(xap[i].zap);
    Writeln;
  End; End;
Readln;
End.

```



ий університет  
 го господарства  
 родокористування

### *Варіанти завдань для самотійної роботи:*

1. Знайти найменшу заробітну плату заміжньої жінки з кількістю дітей більше трьох.
2. Знайти заробітну плату військовозобов'язаних чоловіків.
3. Знайти заробітну плату тих чоловіків та жінок, в яких вона однакова і результат вивести в вигляді таблиці.
4. Знайти заробітну плату незаміжньої жінки, яка має хоча б одну дитину.
5. Знайти сумарну заробітну плату звільнених від військового обов'язку чоловіків.
6. Знайти різницю в сумарній заробітній платі чоловіків та жінок.
7. Знайти на скільки сумарна заробітна плата чоловіків більша від сумарної заробітної плати жінок.
8. Знайти максимальну заробітну плату заміжньої жінки.
9. Знайти максимальну заробітну плату серед військовозобов'язаних чоловіків.
10. Знайти мінімальну заробітну плату серед незаміжніх жінок.
11. З'ясувати, чи заробітні плати менші за g1 або більші за g2 ( $g1 < g2$ ).
12. Знайти різницю між максимальною та мінімальною заробітною платою чоловіків та жінок.
13. Знайти середню заробітну плату звільнених від військового обов'язку чоловіків.
14. Знайти різницю між максимальною та мінімальною заробітною платою чоловіків.
15. Визначити, на скільки середня заробітна плата звільнених від військового обов'язку чоловіків відрізняється від мінімальної заробітної плати жінок.
16. З'ясувати, чи заробітні плати не перевищують середню.

17. Знайти різницю між максимальною та мінімальною заробітною платою жінок.
18. З'ясувати, чиї заробітні плати перевищують середню.
19. З'ясувати, чиї заробітні плати не перевищують сумарну заробітну плату військовозобов'язаних чоловіків.
20. З'ясувати, чиї заробітні плати не перевищують сумарну заробітну плату жінок.
21. Знайти різницю між максимальною та мінімальною заробітною платою чоловіків, які звільнені від службових обов'язків.
22. Знайти мінімальну заробітну плату серед заміжніх жінок.
23. Визначити, на скільки середня заробітна плата заміжніх жінок відрізняється від найбільшої.
24. Визначити, на скільки середня заробітна плата чоловіків відрізняється від найменшої.
25. Знайти мінімальну заробітну плату незаміжньої жінки і порівняти її з середньою.

#### ***Контрольні запитання:***

1. Дайте означення запису.
2. Що таке поле запису?
3. Наведіть способи опису записів в ТР.
4. Що називають складеним іменем? Наведіть приклад.
5. Як відбувається звернення до елементів запису?
6. Яку обробку можна виконувати над елементами запису?
7. Запишіть оператор приєднання і поясніть його роботу.
8. Що розуміють під записом з варіантами?
9. Дайте означення ієрархічних записів.
10. Як описуються вкладені записи? Наведіть приклади.

#### **Лабораторна робота №12**

##### **Тема: Файли даних. Робота з текстовими та типізованими файлами**

**Мета роботи:** Ознайомитися з типізованими, текстовими файлами та файлами без типу. Розглянути способи створення текстових файлів. Вивчити основні процедури та функції для роботи з текстовими файлами.

##### **Хід роботи:**

1. Розробити алгоритми створення текстових, типізованих файлів та файлів без типу згідно варіантів завдань для самостійної роботи.
2. Написати відповідні програми на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.

### Завдання 12.1



#### Приклад 12.1

Компоненти файла є натуральні числа від 96 до 158. Знайти кількість парних та непарних чисел.

**Розв'язання:**

*Приклад створення типізованого файлу:*

```

Program l12v1z1;
  Uses crt;
  Var f1:file of integer;
      i,p,np,k:integer;
  {Процедура створення файла з числами від 96 до 158}
  Procedure stvor;
    Begin
      Assign(f1,'a:ch.txt');
      Rewrite(f1);
      For i:=96 to 158 do
        Write(f1,i);
      Close(f1);
    End;
  Begin
    Clrscr;
    Write('створювати новий файл?(1-так)');
    Readln(k);
    If k=1 then stvor;
    Reset(f1);
    {Обрахування кількості парних та непарних чисел}
    While not eof(f1) do begin
      Read(f1,i);
      If (i mod 2=0) then p:=p+1 else np:=np+1;
    End;
    Close(f1);
    Writeln(' кількість непарних ',np);
    Writeln(' кількість парних ',p);
    Readln;
  End;

```

**End.**

**Результат:** Кількість непарних - 31, кількість парних - 32.

**Приклад створення текстового файлу:**

**Program** l12v1z1;

**Uses** crt;

**Var** fl:text; {Файл тину текст}

i,p,np,k:integer;

**Procedure** stvor;

**Begin**

**Assign**(fl,'a:cht.txt');

**Rewrite**(fl);

**For** i:=96 to 158 **do**

**Writeln**(fl,i);

**Close**(fl);

**End;**

**Begin**

**Clrscr**;

**Write**('створювати новий файл?(1-так)');

**Readln**(k);

**If** k=1 **then** stvor;

**Reset**(fl);

**While** not eof(fl) **do begin**

**Readln**(fl,i);

**If** (i mod 2=0) **then** p:=p+1 **else** np:=np+1;

**End;**

**Close**(fl);

**Writeln**(' кількість непарних ',np);

**Writeln**(' кількість парних ',p);

**Readln**;

**End.**

**Результати:** Кількість непарних - 31, кількість парних - 32.

**Варіанти завдань для самостійної роботи:**

1. Компонентами файла g є натуральні числа від 76 до 138. Знайти кількість парних чисел серед його компонент.
2. Дано файл f, компоненти якого є натуральні числа від 1 до 150. Записати у файл g всі компоненти файла f, які є парними числами.
3. Дано файл f, компоненти якого є натуральні числа від 1 до 150. Записати у файл g всі компоненти файла f, які діляться на 3 і не діляться на 7.



4. Дано файл f, компоненти якого є натуральні числа від 1 до 150. Записати у файл g всі компоненти файла f, які є повними квадратами.
5. Дано файл f, компоненти якого  $A_1, \dots, A_n$  ( $n=9$ ) обчислюються за формулою:  $A_i = 1.5 * i$ , ( $i = \overline{1,9}$ ). Створити файл f.
6. Дано файл даних f: (1,3,-4,-21,5,9,-3,11,-20). Переписати компоненти файла f у файл g так, щоб у файлі g спочатку йшли додатні елементи, а потім від'ємні.
7. Дано натуральне число  $n=10$ . Записати в файл g послідовність чисел  $b_1, b_2, \dots, b_n$ , визначених за формулою  $b_i = 2^i / i!$ ,  $i=1,2,3,\dots,n$ .
8. Послідовність  $x_1, x_2, \dots, x_n$  утворена за законом  $x_i = (i - 0.1) / i^3 + tg 2i$ , ( $i = \overline{1,9}$ ). Дано дійсне число  $\varepsilon = 0.05$ . Записати у файл f ті члени послідовності, для яких виконується умова  $x_i < \varepsilon$ .
9. У файлі даних f: (1, 2, 3, 8, 10, 12, 17.7, 14, 5.6) підрахувати кількість елементів та обчислити суму їх квадратів.
10. У файлі даних f: (7,6,4,3,2,1,0,14,8,19,41,105) підрахувати кількість елементів та знайти їх середнє значення.
11. Дано числовий файл f: (7,6,5,4,-8,5,3,2,1). Записати в числовий файл g компоненти файла f у зворотному порядку.
12. Дано текстовий файл f: ('смородина', 'яблука', 'груші', 'сливи', 'абрикоси', 'порічки'). Записати його компоненти в файл g у зворотному порядку.
13. Дано числовий файл f: (3,4,5,2,4,3,4,3,7). З'ясувати, чи є перші два компоненти файла парними чи непарними.
14. Записати у файл f послідовність чисел Фібоначі:  $U_1, U_2, \dots, U_n$  ( $n=12$ ). Послідовність чисел Фібоначі:  $U_1, U_2, \dots, U_n$  утворюється за законом  $U_0 = 0, U_1 = 1, \dots, U_i = U_{i-1} + U_{i-2}; (i = 2, 3, \dots)$ .
15. Дано числа 7, -5, 3, 4, 6, 1, 8, 100. Записати їх у файл f. Поміняти місцями найбільший і найменший елементи в даному файлі. Записати ці дані в новий файл g.
16. Записати наступні дані у файл послідовного доступу: (7.8, -5.23, 1.2, 0.71, 43, 7.8, 8, 9.76). Знайти: а) суму компонент файла; б) добуток компонент файла; в) останню компоненту файла.
17. Записати наступні дані у файл послідовного доступу:
18. (-55, 14.2, 70.7, 3.3, -8.9, 17.05, -0.6, -10). Знайти: а) найменше значення з парними індексами; б) різницю першої та останньої компоненти файла.
19. Записати наступні дані у файл послідовного доступу:
20. (-5, 51.2, 30.1, 4.3, 7.8, 5611.03, -0.65, 610). Знайти найбільше із значень модулів компонент з парними індексами.
21. Записати наступні дані у файл послідовного доступу:

22. (-5.5,3.2,12.1,-4.3, 7.9,14.03,5.5,34). Знайти: а) кількість чисел файла, які знаходяться в інтервалі [0.2; 0.9]; б) максимальне значення компонент файла, які стоять на парних місцях.
23. Записати символічні дані у файл g: ('золото', 'срібло', 'платина', 'цинк', 'алюміній'). Отримати копію даного файла g.
24. Наступні числові дані записати в два файли: 1) - 4.7 ,11.8, 0.44, 45.6, 23.6, 1.3 → файл f1; 2) 4.5, 56.7, 3.6, 4.6, 47.5, -5.4 → файл f2. Записати дані файла f2 в кінець файла f1.
25. Дано два списки: 1) 'каучук', 'нейлон', 'паролон', 'капрон'; 2) 'залізо', 'олово', 'свинець', 'мідь'. Записати їх в текстові файли f1, f2. Дописати компоненти файла f1 у кінець файла f2.
26. Наступні дані записати у файл f: (1,3,4,2,5,7,8,16,44,77,10). Записати у файл g1 всі парні числа файла f, а у файл g2 - всі непарні числа файла f. При цьому зберегти порядок слідування чисел.
27. Дано файл даних f: (4,7,9,44,18,21,43,72,108,14,66). Записати у масив A ті дані, які кратні 3, та знайти їх суму.
28. Наступні дані записати у два файли:  
 1) 7,11,0.6,1.33,40,13,17,20 → файл f1;  
 2) 0.4,33,1.2,0.5,43,78,90 → файл f2.  
 Записати елементи файла f2 у файл f1 після 3-го елемента.

## Завдання 12.2

### Приклад 12.2

Сформувати файл "BOOK", який містить інформацію про книги вашої бібліотеки. Інформація повинна містити прізвище автора, назву книги, видавництво та рік видання. Видати інформацію про книги видані "Просвітою".

### Розв'язання:

*Приклад створення типізованого файлу:*

**Program** l12z2;

**Uses** crt;

**Type** rr=record

naz,avt,vid:string;

rik:integer;

end;

**Var** f2:file of rr;

rrs:rr;

n,p,no,k:integer;

**Procedure** stvor; {Процедура створення типізованого файлу}

**Begin**

**Assign**(f2,'a:kn.txt');

**Rewrite**(f2);

**Repeat**

**Clrscr**;

**Write**(' назва книги: ');**Readln**(rrs.naz);

**Write**(' автор: ');**Readln**(rrs.avt);

**Write**(' видавництво ');**Readln**(rrs.vid);

**Write**(' рік видання: ');**Readln**(rrs.rik);

**Write**(f2,rrs);

**Write**(' Продовжити створення каталогу книг?(так-1) ');

**Readln**(p);

**Until** p>1 ;

**Close**(f2);

**End**;

**Begin**

**Clrscr**;

**Write**('створити файл ?(1-так) ');

**Readln**(k);

**If** k=1 **then** stvor;

**Clrscr**;

**Assign**(f2,'a:kn.txt');

**Reset**(f2);

**While** not eof(f2) **do begin** {Вивід даних з файла}

**Read**(f2,rrs);

**With** rrs **do**

**Begin**

**Write**(naz:14);**Write**(avt:13);

**Write**(vid:13);**Write**(rik:6);

**End**;

**Writeln**;

**End**;

**Writeln**;

**Reset**(f2);

**Writeln**(' книги видані „Просвітою” ');

**While** not eof(f2) **do begin** {Вивід даних за умовою}

**Read**(f2,rrs);

**If** rrs.vid='просвіта' **then with** rrs **do begin**

**Write**(naz:14);**Write**(avt:13);

**Write**(vid:13);**Write**(rik:6);

**End**;

**End**;

**Close**(f2);

```
Readln;  
End.
```

*Приклад створення текстового файлу:*

```
Program l12v1z2;  
Uses crt;  
Var f2,f3:text;  
    rrs,rrs1,rrs2,rrs3,rrs4,rrs5:string;  
    n,p,no,k,i,h:integer;  
Procedure stvor;    {Процедура створення текстового файлу}  
Begin  
    Assign(f2,'a:knt.txt');  
    Rewrite(f2);  
    Repeat  
        Clrscr;  
        Write(' назва книги: ');  
        Readln(rrs);          Writeln(f2,rrs);  
        Write(' автор: ');    Readln(rrs);  
        Writeln(f2,rrs);  
        Write(' видавництво '); Readln(rrs);  
        Writeln(f2,rrs);      Write(' рік видання: ');  
        Readln(rrs);          Writeln(f2,rrs);  
        rrs:="";  
        Writeln(f2,rrs);  
        Write(' Продовжити створення каталогу книг?(так-1) ');  
        Readln(p);  
    Until p>1 ;  
    Close(f2);  
End;  
Begin  
    Clrscr;  
    Write('створити файл?(1-так) ');  
    Readln(k);  
    If k=1 then stvor;  
    Clrscr;  
    Assign(f2,'a:knt.txt');  
    Reset(f2);  
    While not eof(f2) do begin    {Вивід даних з файлу}  
        For i:=1 to 5 do begin  
            Readln(f2,rrs);  
            Write(rrs:15);  
        End;  
    End;  
    Writeln;
```

```

End;
Reset(f2);
Writeln('  книги  видані просвітою ');
While not eof(f2) do begin      {Вивід даних за умовою}
  Readln(f2,rrs); Readln(f2,rrs2);
  Readln(f2,rrs3); Readln(f2,rrs4);
  Readln(f2,rrs5);
  If rrs3='просвіта' then begin
    Write(rrs1,' ');Write(rrs2,' ');
    Write(rrs3,' ');Write(rrs4,' ');
    Writeln(rrs5);
  End;
End;
Close(f2);
Readln;
End.

```



Національний університет  
водного господарства  
та природокористування

### ***Варіанти завдань для самостійної роботи:***

1. Сформувати файл “BOOK”, який містить інформацію про книги вашої бібліотеки. Інформація повинна містити прізвище автора, назву книги, видавництво та рік видання. Використовуючи сформований файл, роздрукувати інформацію про прочитані книги.
2. Сформувати файл, який містить інформацію про побутові холодильники: назва холодильника, вартість, об'єм холодильної камери, завод-виготовлювач. Використовуючи сформований файл, роздрукувати інформацію про холодильники, вартість яких понад 55000 гривень.
3. Під час футбольної гри формується файл, який включає прізвище гравця та кількість набраних за гру балів. Використовуючи сформований файл, роздрукувати прізвища трьох найрезультативніших гравців команди.
4. Сформувати файл, який містить інформацію про потяги, які слідують до Києва (номер потяга, повна назва, час в дорозі). Використовуючи сформований файл, роздрукувати інформацію про потяги, час перебування в дорозі яких не перевищує 6 годин.
5. Сформувати файл, який містить інформацію про потяги, які відправляються зі станції Здолбунів (враховуючи транзитні): номер потяга, станцію призначення, час відправки, час в дорозі. Використовуючи сформований файл, роздрукувати інформацію про потяги, які йдуть до Львова.

6. Сформувати файл, що містить інформацію про студентів: прізвище, група, дата народження. Вивести дані про тих студентів, які народилися влітку.
7. Сформувати файл “stud”, який має наступну структуру: прізвище студента, рік народження, стать. Роздрукувати, використовуючи файл, список студентів чоловічого роду та вказати їх вік. В кінці списку надрукувати середній вік студентів.
8. Сформувати файл “EXAM” за результатами екзаменаційної сесії (три екзамени). Інформація про студентів вводиться у символічному вигляді в наступному порядку: прізвище **\*\*N1\*\*N2\*\*N3**, де Ni-оцінка. Використовуючи файл, роздрукувати результати сесії у вигляді таблиці. Передбачити друк шапки таблиці з назвами дисциплін.
9. Протокол лижних гонок записати у файл “SCI”. Для кожного учасника ввести: прізвище, час старту та фінішу (година, хвилина, секунда). Використовуючи сформований файл, роздрукувати прізвища учасників, котрі виконали норми ГПО.
10. Сформувати файл “CAR”, який містить інформацію про власників автомобілів: прізвище, марка, колір. Використовуючи сформований файл, роздрукувати відомості про власників, які мають автомобілі “Жигулі” зеленого кольору.
11. Сформувати файл, який містить дані про книги вашої особистої бібліотеки: прізвище автора, назва книги, видавництво, рік видання, кількість сторінок. Використовуючи сформований файл, роздрукувати інформацію про книги, випущені у видавництві “Мир”, а також підрахувати загальну кількість таких книг.
12. Сформувати файл, який міститиме інформацію про магнітофони: марка, його вартість, клас. Використовуючи сформований файл, роздрукувати інформацію про магнітофони першого класу.
13. Сформувати файл - телефонний довідник. Інформація повинна містити прізвище, ім'я та по батькові абонента, номер телефону. Роздрукувати весь довідник.
14. Сформувати файл “FRIEND” із прізвищ і дат народження ваших друзів. Використовуючи сформований файл, роздрукувати прізвища тих, хто народився взимку.
15. У шаховому турнірі беруть участь 10 шахістів. Сформувати файл, який включає прізвища та результати ігор ( перемога – 1, нічия – -1, програш – 0). Використовуючи сформований файл, обробити результати чемпіонату і роздрукувати назви команд, які зайняли призові місця, а також кількість перемог кожної команди.
16. В чемпіонаті з футболу бере участь 16 команд. Сформувати файл команд і результатів матчу (виграш - 2 очка, нічия -1, програш - 0). Використовуючи сформований файл, обробити результати чемпіонату і

роздрукувати назви команд, які зайняли призові місця, а також кількість перемог кожної команди.

17. В журналі обліку відвідувань щодня по кожному з предметів відмічають пропуск заняття студентами. Сформувати файл, який включає прізвище, дату відвідування занять з одного предмета та відмітку про присутність (1 - присутній, 0 - відсутній) кожним студентом групи. Використовуючи сформований файл, сформувати список тих студентів, які мають більше 5 пропусків, і вивести дати пропущених занять.
18. Сформувати файл, який включає прізвища та оцінки студентів впродовж семестру з дисципліни “Обчислювальна техніка”. Використовуючи сформований файл, роздрукувати прізвища тих студентів, в котрих середній бал з дисципліни - 4.
19. До 20 спортивних журналістів звернулись із проханням назвати 3 кращих футболістів сезону. Сформувати файл, який включає прізвища футболістів, кількість набраних очків від кожного журналіста (3 - перше місце, 2 - друге місце, 1 - третє місце). Використовуючи сформований файл, визначити 3 кращих гравців.
20. Сформувати файл, який включає прізвища та посади викладачів, які викладають дисципліну “Обчислювальна техніка” на всіх факультетах університету. Використовуючи сформований файл, роздрукувати дані про тих викладачів, прізвища яких починаються на літеру “В”.
21. Скласти програму створення текстового файлу з іменем T, що містить інформацію про „Продтовари”. Вивести відповідну повну таблицю даних файлу і ту її частину, яка містить відомості про асортимент продтоварів у магазині, найменування, ціну, кількість, дату випуску, термін зберігання. Використовуючи сформований файл, вивести інформацію про список товарів, ціна яких вища вказаної. Передбачити в програмі вилучення та додавання даних.
22. Скласти програму створення текстового файлу з іменем T, що містить інформацію про „Магнітофони”. Вивести відповідну повну таблицю даних файлу і ту її частину, яка містить відомості про магнітофони у магазині: назва, завод виробника, місто, рік випуску, клас, ціна, кількість. Використовуючи сформований файл, вивести інформацію про магнітофони певного класу. Передбачити в програмі вилучення та додавання даних.
23. Скласти програму створення текстового файлу „Журнал відвідування”. Вивести відповідну повну таблицю даних файлу і ту її частину, яка містить відомості про відвідування студентами групи занять за тиждень: прізвище та ініціали, 1 день, 2 день, 3 день, 4 день, 5 день. Використовуючи сформований файл, вивести інформацію про студентів, які не мають жодного пропуску. Передбачити в програмі вилучення та додавання даних.

24. Скласти програму створення текстового файлу „Вироби”. Вивести відповідну повну таблицю даних файла і ту її частину, яка містить відомості про кількість виробів у складальному цеху заводу: прізвище та ініціали робітника, 1 день, 2 день, ..., 5 день. Використовуючи сформований файл, вивести інформацію про робітників, які випустили не менше вказаної кількості виробів. Передбачити в програмі вилучення та добавлення даних.

25. Скласти програму створення текстового файлу з іменем Т „Співробітники університету”. Вивести відповідну повну таблицю даних файла і ту її частину, яка містить інформацію про співробітників університету: прізвище та ініціали, рік народження, підрозділ, посада, стаж роботи, місячний оклад. Використовуючи сформований файл, вивести інформацію про співробітників вказаного підрозділу. Передбачити в програмі вилучення та добавлення даних.

### Завдання 12.3

Результати екзаменаційної сесії групи студентів 1-го курсу подані у вигляді таблиці, наведені в п. 11.1. Попередньо створивши файл(и), які містять дані цієї таблиці, та використовуючи створений(і) файл(и), розробити програму згідно варіанта.

#### Приклад 12.3

Надрукувати таблицю, що містить номери, прізвища та кількість “5”, “4”, “3”, “2” у кожного студента групи, а також підрахувати загальну кількість “5”, “4”, “3”, “2” в групі.

#### Розв’язання:

#### *Приклад створення типізованого файлу:*

```
Program l12v1z3;
```

```
Uses crt;
```

```
Label m1,m2;
```

```
Const a:array [1..10] of string=('Берестяк','Памченко','Заєць',  
    'Іванчук','Кравченко','Панченко','Сидорук',  
    'Снопок','Якимець','Якубів');
```

```
    b:array [1..10] of integer =(4,5,3,4,2,3,5,4,4,5);
```

```
    c:array [1..10] of integer =(3,4,4,3,3,5,4,2,5,5);
```

```
    d:array [1..10] of integer =(3,4,4,3,3,4,4,3,5,4);
```

```
    f:array [1..10] of integer =(4,5,4,3,2,5,3,3,5,4);
```

```
Var i,j,ch,o,h,t,z,p,l,v,q,po,m,pr,osys,al:integer;
```

```
    fl:file of integer;
```



```

    a1:file of string;
    w,pri:string;
    {Процедура обчислення кількості оцінок}
    Function k(i,o:integer):integer;
    Begin
        j:=0;
        If m=o then j:=j+1;
        If pr=o then j:=j+1;
        If osys=o then j:=j+1;
        If al=o then j:=j+1;
        k:=j;
    End;
    Procedure stvor;          {Процедура створення файлу}
    Begin
        Assign(a1,'d:/pri.dat');
        Assign(f1,'d:/alg.dat');
        Rewrite(a1);Rewrite(f1);
        For po:=1 to 10 do begin
            Write(a1,a[po]);
            Write(f1,b[po],c[po],d[po],f[po]);
        End;
        Close(a1);Close(f1);
    End;

```

```

    Begin
    Clrscr;
    Write(' створити файл, чи він уже існує (1-так) ');
    Readln(i);
    If i=1 then stvor;
        Assign(a1,'d:/pri.dat');    Assign(f1,'d:/alg.dat');
        Reset(a1);                Reset(f1);
    Writeln (' Студент мат.ан прог. ОС алгебра');
    For q:=1 to 10 do begin
        Read(a1,pri);
        Read(f1,m,pr,osys,al);
    {Виведення прізвищ студентів та їх оцінок}
    Writeln(pri:12,m:8,pr:8,osys:8,al:8);
    End;
    Writeln; Writeln;
    Reset(a1);Reset(f1);
    Write (' Студент к-ть 2 к-ть 3 к-ть 4 к-ть 5');
    For q:=1 to 10 do begin      {Введення оцінок}
        Read(a1,pri);

```

```

Read(fl,m,pr,osys,al);
Writeln(pri:12);
Write(pri:12);
o:=2;h:=k(q,o);
Write (h:8);
z:=z+h;o:=3;h:=k(q,o);
Write (h:8);
t:=t+h;o:=4;h:=k(q,o);
Write (h:8);
ch:=ch+h;o:=5;h:=k(q,o);
Write (h:8);
p:=p+h;
m2:end;
Writeln;
Textcolor(22);
w:='Всього';
Write(w:12,z:8,t:8,ch:8,p:8);
Readln;
End.

```

**Приклад створення текстового файлу:**

```

Program l12v1z3;
Uses crt;
Label m1,m2;
Const a:array [1..10] of string=('Берестяк', 'Памченко', 'Заєць', 'Іванчук',
'Кравченко', 'Панченко', 'Сидорук', 'Снопок', 'Якимець', 'Якубів');
b:array [1..10] of integer =(4,5,3,4,2,3,5,4,4,5);
c:array [1..10] of integer=(3,4,4,3,3,5,4,2,5,5);
d:array [1..10] of integer=(3,4,4,3,3,4,4,3,5,4);
f:array [1..10] of integer=(4,5,4,3,2,5,3,3,5,4);
Var i,j,ch,o,h,t,z,p,l,v,q,po,m,pr,osys,al:integer;
fl:text;
al:text;
w,pri:string;
Function k(i,o:integer):integer; {Функція обчислення кількості оцінок}
Begin
j:=0;
If m=o then j:=j+1;
If pr=o then j:=j+1;
If osys=o then j:=j+1;
If al=o then j:=j+1;
k:=j;
End;

```

**Procedure** stvor;

**Begin**

**Assign**(a1,'d:/pri.txt');**Assign**(f1,'d:/alg.txt');

**Rewrite**(a1); **Rewrite**(f1);

**For** po:=1 **to** 10 **do begin**

**Writeln**(a1,a[po]);**Writeln**(f1,b[po]);

**Writeln**(f1,c[po]);**Writeln**(f1,d[po]);

**Writeln**(f1,f[po]);

**End**;

**Close**(a1);**Close**(f1);

**End**;

**Begin**

**Clrscr**;

**Write**(' створити файл чи він уже існує (1-так) ');

**Readln**(i);

**If** i=1 **then** stvor;

**Assign**(a1,'d:/pri.txt');

**Assign**(f1,'d:/alg.txt');

**Reset**(a1);**Reset**(f1);

**Writeln** (' Студент мат.ан прог. ОС алгебра');

**For** q:=1 **to** 10 **do begin** *{Введення оцінок}*

**Readln**(a1,pri);**Readln**(f1,m);

**Readln**(f1,pr);**Readln**(f1,osys);

**Readln**(f1,al);

*{Виведення прізвищ студентів та їх оцінок}*

**Writeln**(pri:12,m:8,pr:8,osys:8,al:8); **End**;

**Writeln**; **Writeln**;

**Reset**(a1);**Reset**(f1);

**Write** (' Студент к-ть 2 к-ть 3 к-ть 4 к-ть 5');

**For** q:=1 **to** 10 **do begin**

**Readln**(a1,pri);**Readln**(f1,m,pr,osys,al);

**Writeln** (pri:12);

**Write**(pri:12);

*{Визначення кількості оцінок кожного студента групи}*

o:=2; h:=k(q,o); write (h:8);

z:=z+h; o:=3;h:=k(q,o);

**Write** (h:8);

t:=t+h; o:=4; h:=k(q,o);

**Write** (h:8);

ch:=ch+h; o:=5; h:=k(q,o);

**Write** (h:8);

p:=p+h;

```

m2:end;
Writeln;
Textcolor(22);
w:='Всього';
Write(w:12,z:8,t:8,ch:8,p:8);
Readln;
End.

```

#### Завдання 12.4

Скласти програму створення текстового табличного файлу з іменем Т, з певною інформацією. Вивести відповідну повну таблицю даних файла і ту її частину, яка містить вибрані дані згідно варіанту. Передбачити в програмі вилучення та додавання даних.

#### Приклад 12.4

Сформувати файл Т, який містить інформацію про книги вашої бібліотеки. Інформація повинна містити прізвище автора, назву книги, видавництво та рік видання. Вивести інформацію про книги видані 1991 року.

#### Розв'язання:

*Приклад створення типізованого файлу прямого доступу:*

**Program** l12v1z4;

```

Uses
  crt;
Label m1,m2;
Type
  rr=record
    naz,avt,vid:string;
    rik:integer;
  end;

```

**Var**

```

f2,f3:file of rr;
rrs:rr;
n,p,no,k,i:integer;

```

**Procedure** stvor; *{Процедура створення файлу з даними}*

**Begin**

```

  Clrscr;
  Assign(f2,'a:kn1.txt');
  Rewrite(f2);
  Repeat
    Write('  назва: ');    Readln(rrs.naz);
    Write('  автор: ');   Readln(rrs.avt);

```

```

Write(' видавництво: '); Readln(rrs.vid);
Write(' рік: '); Readln(rrs.rik);
Write(f2,rrs);
Write(' створювати ще каталог (1-так)?');
Readln(k);
Until k<>1;
Close(f2);
End;

```

**Procedure** dob; *{Процедура добавлення даних у файл}*

```

Begin
  Clrscr;
  Reset(f2);
  Seek(f2,filesize(f2));
  Repeat
    Write(' назва: '); Readln(rrs.naz);
    Write(' автор: '); Readln(rrs.avt);
    Write(' видавництво: '); Readln(rrs.vid);
    Write(' рік: '); Readln(rrs.rik);
    Write(f2,rrs);
    Write(' створювати ще каталог (1-так)?');
    Readln(k);
  Until k<>1;
  Close(f2);
End;

```

**Procedure** vil; *{Процедура вилучення даних з файла}*

```

Begin
  Clrscr;
  Reset(f2); Assign(f3,'a:num');
  Rewrite(f3);
  Write(' номер книги, яку потрібно видалити:');
  Readln(k);
  For i:=1 to k-1 do
    Begin
      Read(f2,rrs); Write(f3,rrs);
    End;
  Read(f2,rrs);
  While not eof(f2) do {Перехід в кінець файла}
    Begin
      Read(f2,rrs); Write(f3,rrs);
    End;
  Close(f2); Close(f3);
  Erase(f2); Rename(f3,'a:kn1.txt');
End;

```

```

Procedure show;           {Процедура перегляду вмісту файла}
Begin
  Reset(f2);
  Clrscr;
  While not eof(f2) do           {Перехід в кінець файла}
    Begin
      Read(f2,rrs);
      Writeln(rrs.naz:13,rrs.avt:12,rrs.vid:15,rrs.rik:5);

```

```

    End;

```

```

End;

```

```

Procedure umov;           {Вивід даних за умовою }

```

```

Begin

```

```

  Clrscr;

```

```

  Reset(f2);

```

```

  Textcolor(3);

```

```

  Writeln(' Книги 1991 року ');

```

```

  While not eof(f2) do

```

```

    Begin

```

```

      Read(f2,rrs);

```

```

If rrs.vid='просвіта' then

```

```

  writeln(rrs.naz:13,rrs.avt:12,rrs.vid:15,rrs.rik:5);

```

```

    End;

```

```

  Close(f2);

```

```

End;

```

```

Procedure menu;           {Процедура створення меню}

```

```

Var

```

```

  cc:char;

```

```

Begin

```

```

  While cc<>'6' do

```

```

    Begin

```

```

      Clrscr;

```

```

      Writeln('1----Створити файл');

```

```

      Writeln('2----Показати файл');

```

```

      Writeln('3----Добавити дані в файл');

```

```

      Writeln('4----Вилучити запис');

```

```

      Writeln('5----Вивід за умовою');

```

```

      Writeln('6----Вихід');

```

```

      cc:=readkey;

```

```

      Case cc of

```

```

        '1':stvor;

```

```

        '2':show;

```

```

        '3':dob;

```

```

        '4':vil;

```

```

        '5':umov;
        '6':exit;
    End;
End;
End;
Begin
    Assign(f2,'a:kn1.txt');
    Clrscr;
    Menu;
End.

```

**Приклад створення текстового файлу:**

```

Program l12v1z4;
Uses
    crt;
Label m1,m2;
Var
    f2,f3:text;
    n,p,no,k,i:integer;
    naz,avt,vid:string;
    rik:integer;
Procedure stvor;                                {Процедура створення файла з даними}
Begin
    Clrscr;
    Assign(f2,'c:kn1.txt');
    Rewrite(f2);
    Repeat
        Write('    назва: ');
        Readln(naz);      Writeln(f2,naz);
        Write('    автор: ');Readln(avt);
        Writeln(f2,avt);
        Write('    видавництво: ');Readln(vid);
        Writeln(f2,vid);  Write('    рік: ');
        Readln(rik);Writeln(f2,rik);
        Write('    створювати ще каталог (1-так)?');
        Readln(k);
    Until k<>1;
    Close(f2);
End;
Procedure dob;                                {Процедура додавання даних у файл}
Begin
    Clrscr;
    Reset(f2);

```

```

Append(f2);
Repeat
    Write(' назва: ');      Readln(naz);
    Writeln(f2,naz);       Write(' автор: ');
    Readln(avt);           Writeln(f2,avt);
    Write(' видавництво: '); Readln(vid);
    Writeln(f2,vid);       Write(' рік: ');
    Readln(rik);           Writeln(f2,rik);
    Write(' створювати ще каталог (1-так)?');
    Readln(k);
Until k<>'1';
Close(f2);

```

```

End;
Procedure vil;                {Процедура вилучення даних з файла}

```

```

Begin
    Clrscr;
    Reset(f2);
    Assign(f3,'a:num');
    Rewrite(f3);
    Write(' номер книги,яку потрібно видалити:');
    Readln(k);
    For i:=1 to k-1 do
        Begin
            Readln(f2,naz);Readln(f2,avt);
            Readln(f2,vil);Readln(f2,rik);
            Writeln(f3,naz);Writeln(f3,avt);
            Writeln(f3,vid);Writeln(f3,rik);
        End;
        Readln(f2,naz);Readln(f2,avt);
        Readln(f2,vil);Readln(f2,rik);
    While not eof(f2) do      {Перехід в кінець файла}
        Begin
            Readln(f2,naz);Readln(f2,avt);
            Readln(f2,vid);Readln(f2,rik);
            Writeln(f3,naz);Writeln(f3,avt);
            Writeln(f3,vid);Writeln(f3,rik);
        End;
        Close(f2); Close(f3);
        Erase(f2); Rename(f3,'a:kn1.txt');
    End;
Procedure show;              {Процедура перегляду вмісту файла}
Begin

```



```

Reset(f2);
Clrscr;
While not eof(f2) do      {Перехід в кінець файла}
  Begin
    Readln(f2,naz);Readln(f2,avt);
    Readln(f2,vid);Readln(f2,rik);
    Write(naz,' ');Write(avt,' ');
    Write(vid,' ');Writeln(rik,' ');

```

```

End;
End;

```

```

Procedure umov;          {Вивід даних за умовою }

```

```

Begin
  Clrscr;
  Reset(f2);
  Textcolor(3);
  Writeln(' Книги 1991 року ');
  While not eof(f2) do
    Begin
      Readln(f2,naz);Readln(f2,avt);
      Readln(f2,vid);Readln(f2,rik);
      If rik=1991 then writeln(naz:13,avt:12,vid:15,rik:5);
    End;
  Close(f2);

```

```

End;
Procedure menu;          {Процедура створення меню}

```

```

Var
  cc:char;

```

```

Begin
  While cc<>'6' do
    Begin
      Clrscr;
      Writeln('1----Створити файл');
      Writeln('2----Показати файл');
      Writeln('3----Добавити дані в файл');
      Writeln('4----Вилучити запис');
      Writeln('5----Вивід за умовою');
      Writeln('6----Вихід');
      cc:=readkey;
      Case cc of
        '1':stvor;
        '2':show;
        '3':dob;
        '4':vil;

```

```

        '5':umov;
        '6':exit;
    End;
End;
End;
Begin
    Assign(f2,'a:kn1.txt');
    Clrscr;
    Menu;
End.

```

**Приклад створення типізованого файлу послідовного доступу:**

```

Program l12v1z4;

```

```

Uses

```

```

    crt;

```

```

Label

```

```

    m1,m2;

```

```

Type

```

```

    rr=record

```

```

        naz,avt,vid:string;

```

```

        rik:integer;

```

```

    End;

```

```

Var

```

```

    f2,f3:file of rr;

```

```

    rrs:rr;

```

```

    n,p,no,k,i:integer;

```

```

Procedure stvor;

```

*{Процедура створення файлу з даними}*

```

Begin

```

```

    Clrscr;

```

```

    Assign(f2,'a:kn1.txt');

```

```

    Rewrite(f2);

```

```

    Repeat

```

```

        Write('    назва: ');    Readln(rrs.naz);

```

```

        Write('    автор: ');    Readln(rrs.avt);

```

```

        Write('    видавництво: ');Readln(rrs.vid);

```

```

        Write('    рік: ');      Readln(rrs.rik);

```

```

        Write(f2,rrs);

```

```

        Write('    потрібно створювати ще (1-так)?');

```

```

        Readln(k);

```

```

    Until k<>1;

```

```

    Close(f2);

```

```

End;

```

```

Procedure dob;    {Процедура додавання даних у файл}

```

**Begin**

**Clrscr;**

**Reset(f2);**

**While not eof(f2) do**     *{Перехід в кінець файла}*

**Read(f2,rrs);**

**Repeat**

**Write(' назва: ');**         **Readln(rrs.naz);**

**Write(' автор: ');**         **Readln(rrs.avt);**

**Write(' видавництво: ');** **Readln(rrs.vid);**

**Write(' рік: ');**             **Readln(rrs.rik);**

**Write(f2,rrs);**

**Write(' потрібно добавляти ще (1-так)?');**

**Readln(k);**

**Until k<>1;**

**Close(f2);**

**End;**

**Procedure vil;**                     *{Процедура вилучення даних з файла}*

**Begin**

**Clrscr;**

**Reset(f2);**

**Assign(f3,'a:num');**

**Rewrite(f3);**

**Write(' номер книги, яку потрібно вилучити:');**

**Readln(k);**

**For i:=1 to k-1 do**

**Begin**

**Read(f2,rrs);Write(f3,rrs);**

**End;**

**Read(f2,rrs);**

**While not eof(f2) do**

**Begin**

**Read(f2,rrs);Write(f3,rrs);**

**End;**

**Close(f2);**     **Close(f3);**

**Erase(f2);**     **Rename(f3,'a:kn1.txt');**

**End;**

**Procedure show;**                 *{Процедура перегляду вмісту файла}*

**Begin**

**Reset(f2);**

**Clrscr;**

**While not eof(f2) do**         *{Перехід в кінець файла}*

**Begin**

```

        Read(f2,rrs);
        Writeln(rrs.naz:13,rrs.avt:12,rrs.vid:15,rrs.rik:5);
    End;
End;
Procedure umov;      {Вивід даних за умовою }
Begin
    Clrscr;
    Reset(f2);
    Textcolor(3);
    Writeln('-----Книги1991 року ');
    While not eof(f2) do
        Begin
            Read(f2,rrs);
            If rrs.vid='1991' then writeln(rrs.naz:13,rrs.avt:12,rrs.vid:15,rrs.rik:5);
        End;
    Close(f2);
End;
Procedure menu;      {Процедура створення меню }
Var
    cc:char;
Begin
    While cc<>'6' do
        Begin
            Clrscr;
            Writeln('1----Створити файл');
            Writeln('2----Показати файл');
            Writeln('3----Добавити дані в файл');
            Writeln('4----Вилучити запис');
            Writeln('5----Вивід за умовою');
            Writeln('6----Вихід');
            cc:=readkey;
            Case cc of
                '1':stvor;
                '2':show;
                '3':dob;
                '4':vil;
                '5':umov;
                '6':exit;
            End;
        End;
    End;
End;
Begin
    Assign(f2,'a:kn1.txt');

```

**Clrscr;**  
**Menu;**  
**End.**

**Результати:**

*Створення файлу:*

Назва: *Програмування*

Автор: *Фаронов*

Видавництво: *Либідь*

Рік: *1991*

*Створювати ще каталог (1-так)? 1*

Назва: *Дискретна математика*

Автор: *Семенюк*

Видавництво: *РДТУ*

Рік: *1990*

*Створювати ще каталог (1-так)? 2*

*Додавання даних::*

Назва: *Basic*

Автор: *Соботюк*

Видавництво: *РДТУ*

Рік: *1991*

*Створювати ще каталог (1-так)?2*

*Після добавлення:*

Програмування	Фаронов	Либідь	1991
Дискретна математика	Семенюк	РДТУ	1990
Basic	Соботюк	РДТУ	1991

*Вилучення:*

Номер книги, Яку потрібно видалити:3

*Після вилучення:*

Програмування	Фаронов	Либідь	1991
Дискретна математика	Семенюк	РДТУ	1990

*Вивід за умовою:*

Програмування	Фаронов	Либідь	1991
---------------	---------	--------	------

### **Варіанти завдань для самостійної роботи:**

1. Т="книги"; ... книги особистої бібліотеки: автор, назва книги, видавництво, рік видання, кількість сторінок, ціна. Вивести книгу, видану у видавництві „Просвіта” 2000 року.
2. Т="телевізори"; ...телевізори в магазині: місце виробництва, назва телевізора, розмір діагоналі, рік та місяць випуску, ціна, кількість; ...телевізори даної марки.
3. Т="холодильники"; ...холодильники на складі: місто виробництва, назва холодильника, обсяг холодильної камери, рік та місяць випуску, ціна, кількість; ...холодильники 2009 року випуску.
4. Т="чемпіонат з футболу"; ...команди чемпіонату: назва, місто, кількість ігор, кількість очок, кількість вигравів, кількість програвів, місце; ...команди, які мають 3 програти.
5. Т="студенти групи"; ...студентів групи: прізвища та ініціали, область проживання батьків, стать, рік народження, кількість балів екзаменаційної сесії; ...студентів вказаного року народження.
6. Т="поверх гуртожитку"; ...студентів, які проживають на вказаному поверсі гуртожитку: прізвище та ініціали, кімната, стать, рік народження, факультет, курс, група; ...студентів вказаної кімнати.
7. Т="потяги на Київ"; ...потяги, які слідуєть на Київ: № потягу, назва, тип, час відправлення із Рівного, час прибуття в Київ, ціна квитка; ...прибуття потягів до 12 год. дня в Київ.
8. Т="автобуси на Львів"; ...автобуси, які слідуєть на Львів: № рейсу, назва, тип, час відправлення з Рівного, час прибуття до Львова, ціна квитка; ...автобуси, які відправляються з Рівного після 12 год. дня.
9. Т="літаки на південь"; ...літаки, які вилітають на південь: № рейсу, назва, тип, час вильоту з Рівного, час прильоту в пункт призначення, ціна квитка; ...літаки вказаного типу.
10. Т="потяги із Рівного"; ...потяги, які слідуєть через Рівне: № потяга, назва, тип, час відправлення із Рівного, час прибуття в кінцевий пункт, ціна квитка; ...потяги, які слідуєть на Луцьк.
11. Т="автобуси на захід"; ...автобуси, які слідуєть через Рівне на Захід: № рейсу, назва, тип, час відправлення з Рівного, час прибуття в кінцевий пункт, ціна квитка; ...автобуси, які слідуєть на Дубно.
12. Т="літаки через Рівне"; ...літаки, які слідуєть через Рівне: № рейсу, місто відправлення, місто прибуття, тип, час вильоту із Рівного, час прильоту в пункт призначення, ціна квитка; літаки, які слідуєть на Харків.
13. Т="Друзі"; ...мої друзі: прізвище та ініціали, дата народження, вулиця, № будинку, № квартири, телефон; ...друзі, які проживають на даній вулиці.

14. Т=“Знайомі”; ...мої знайомі: прізвище та ініціали, дата народження, пункт проживання, вулиця, № будинку, № квартири, телефон; ...знайомі, які проживають в даному пункті.
15. Т=“Ансамблі”; ...ансамблі: назва, місто, кількість людей, вид мистецтва, рік створення; ...ансамблі вказаного виду мистецтва.
16. Т=“Успішність з о/т”; ...успішність студентів групи з обчислювальної техніки: прізвище та ініціали, оцінка по лаб. роб.№1, оцінка по лаб. роб. №2, оцінка по лаб. роб. №3, оцінка по лаб. роб. №4; ...успішність студентів, у яких середній бал <4.
17. Т=“Магнітофони”; ...магнітофони в магазині: назва, завод-виробник, місто, рік випуску, клас, ціна, кількість; ...магнітофони даного класу.
18. Т=“Шаховий чемпіонат”; ...шахісти: прізвище, та ініціали, місто, рейтинг, число перемог, число нічиїх, число поразок, місце; учасників, які не мають поразок.
19. Т=“Журнал відвідування”; ...відвідування студентами групи занять за тиждень: прізвище та ініціали, 1 день, 2 день, 3 день, 4 день, 5 день; ...студенти, які не мають пропусків.
20. Т=“Зарплата”; ...місячна зарплата робітників цеху: прізвище та ініціали, рік народження, посада, розряд, стаж роботи, розмір зарплати; ...робітників вказаної посади.
21. Т=“Вироби”; ...кількість виробів в складальному цеху заводу: прізвища та ініціали робітника, 1 день, 2 день... 5 день; ...робітників, які випустили не менше вказаної кількості виробів.
22. Т=“Телефонний абонент”; ... відомість про телефонного абонента: прізвище та ініціали, адреса, рік встановлення телефону, номер телефону, ...список абонентів з вказаним роком встановлення телефону.
23. Т=“Продтовари” ... асортимент продтоварів в магазині: найменування, ціна, кількість, дата випуску, термін зберігання; ...список товарів, ціна яких вище вказаної.
24. Т=“Продтовари” ... асортимент продтоварів в магазині: найменування, ціна, кількість, дата випуску, термін зберігання; ...список товарів, ціна яких вище вказаної.
25. Т=“Співробітники інституту”; ... співробітників інституту: прізвище та ініціали, рік народження, підрозділ, посада, стаж роботи, місячний оклад; список співробітників вказаного підрозділу.

#### **Контрольні запитання:**

1. Обґрунтуйте доцільність використання файлів даних.
2. Наведіть схему класифікації файлів в ТР.
3. Дайте означення фізичного та логічного файлів в ТР, зобразіть їх структуру.
4. Наведіть схему класифікації файлів в ТР.

5. Дайте означення типізованого та текстового файлів в TP та наведіть способи їх опису в програмі.
6. Які додаткові процедури та функції використовуються в текстових файлах?
7. Які способи створення текстових файлів Ви знаєте?
8. Як організовується прямий доступ до компонент файла?
9. Яке призначення безтипкових файлів?
10. Як організовується прямий доступ у типізованих файлах?



Національний університет  
водного господарства  
та природокористування

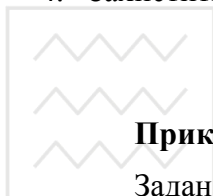
## Лабораторна робота №13

### Тема: Множини

**Мета роботи:** Ознайомитися з основними операціями та прийомами роботи з множинами.

#### Хід роботи:

1. Розробити алгоритм розв'язку поставленої задачі з використанням множин.
2. Написати відповідну програму на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.



Національний університет  
водного господарства  
та природокористування

### Завдання 13.1

#### Приклад 13.1

Задана множина типу стрічка, що завершується символом ' '.

Описати функцію, яка підраховує загальну кількість цифр і знаків в даній стрічці.

#### Розв'язання:

**Uses** crt;

**Label** m1;

**Type** ff=set of char;

{Опис множини}

**Var** n,i:char; s:ff; f,v:integer;

**Procedure** stvor;

{Створення множини}

**Begin**

**Writeln**('введіть множину');

    s:=[ ];



```

While n <> '' do
    Begin
        Read(n);
        S:=s+[n];
    End;
End;
Procedure obchis;
    {Процедура обчислення кількості цифр і знаків
даной множини}

```

```

Begin
    For i:='0' to '9' do
        If i in s then v:=v+1;
    For i:='!' to 'я' do
        If i in s then f:=f+1;
    Writeln(' Кількість цифр в множині ',v);
    Writeln(' Кількість знаків ',f);
End;
Begin
    Clrscr;
    stvor; obchis;
    Readln;
End.

```

**Результат:** Множина ['1', ',', '4', '~']

Кількість цифр: 2;  
Кількість знаків: 4.

### **Варіанти завдань для самостійної роботи:**

1. Дано речення. Створити множини літер, які повторюються та які не повторюються в реченні.
2. З множини цілих чисел 1..20 виділити: множину чисел, що діляться на 6 без остачі; множину чисел, які діляться без остачі або на 2, або на 3.
3. Скласти програму, яка аналізує множину типу текст, який закінчується символом “крапка” і надрукувати знайдені приголосні букви.
4. Дано три множини символного типу  $y_1 = \{ 'A', 'B', 'D', 'R', 'M' \}$ ;  $y_2 = \{ 'R', 'A', 'H', 'D' \}$ ;  $y_3 = \{ 'R', 'A' \}$ . Сформувати нову множину:  $x = (y_1 \cap y_2) \cup (y_1 \setminus y_2)$ .
5. Дана множина назв фігур. Перевірити, чи входить в дану множину фігура “Коло”, та в разі її відсутності видати відповідне повідомлення.
6. Дано три множини символного типу  $y_1 = \{ 'A', 'B', 'D', 'R', 'M' \}$ ;  $y_2 = \{ 'R', 'A', 'H', 'D' \}$ ;  $y_3 = \{ 'R', 'A' \}$ . Сформувати нову множину:  $z = (y_1 \cup y_3) \cup (y_2 \setminus y_1)$ .

7. Дана множина цілих чисел  $-20..20$ . Виділити: множину чисел, що діляться на 4 без остачі; множину чисел, які діляться без остачі на 5, або на 12.
8. Дано три множини символного типу  $y_1 = ['A', 'B', 'D', 'R', 'M']; y_2 = ['R', 'A', 'H', 'D']; y_3 = ['R', 'A']$ . Сформувати нову множину:  $g = y_3 \cup (y_1 \cap y_2) \cup (y_1 \setminus y_2)$ .
9. Дано три множини символного типу  $A, B$  і  $C$ . Сформувати нову множину  $X$ , яка є об'єднанням множин  $A$  і  $B$  та перевірити, чи включена множина  $C$  в множину  $X$ .
10. Дана множина  $K$  довільних чисел. Сформувати множину чисел, що діляться без остачі на 3 та множину всіх дійсних чисел від  $-20..20$ , які не містить множина  $K$ .
11. Дано множини  $A=(1,7,3)$  та  $B=(3,7,1)$ . Перевірити чи множина  $A$  включена в множину  $B$ . Вивести на екран відповідне повідомлення.
12. Дано множину типу текст. Утворити множину символів, відмінних від букв та пропуску в даному тексті.
13. Дано три множини символного типу  $y_1 = ['A', 'B', 'D', 'R', 'M']; y_2 = ['R', 'A', 'H', 'D']; y_3 = ['R', 'A']$ . Сформувати нову множину:  $t = y_3 \setminus (y_1 \cup y_2)$ .
14. Дана множина натуральних чисел від 0 до 100. Сформувати множини чисел, різниця між якими рівна десяти та які діляться на 2 без остачі.
15. Дано речення. Створити множину літер, яких немає в реченні.
16. Дана множина типу текст. Надрукувати ті літери, які зустрічаються в кожному слові тексту тільки один раз.
17. Дано дві множини  $A$  та  $B$ , елементами яких є розділові знаки. Перевірити, чи дані множини еквівалентні, в протилежному випадку видати відповідне повідомлення.
18. Дано множину типу текст. Знайти найбільше число цифр, які розташовані підряд.
19. Дана множина раціональних чисел від  $-25$  до  $55$ . Сформувати множину цілих чисел із заданого діапазону.
20. Дано чотири множини символного типу  $A = ['c', '+', 'k', '-', 'v', '*']; B = ['x', '#', 'g']; C = ['o', '7', '&']; D = ['s', 'x', '+']$ . Сформувати множину:  $N = D \cup (A \setminus C) \cap (B \setminus D)$ .
21. Дано множину типу текст, який закінчується символом "крапка". Надрукувати знайдені голосні букви, які входять більше, ніж в одне слово.
22. Дано три множини символного типу  $A, B$  і  $C$ . Сформувати нову множину  $Z$ , яка складається із тих елементів множини  $B$ , які не входять в множину  $A$ , проте є елементами множини  $C$ .
23. Дана цілочислова квадратна матриця розмірності  $n$ . Елементи матриці знаходяться в діапазоні від 1 до 100. Надрукувати всі цифри із заданого діапазону, яких немає ні в одному рядку заданої матриці.

24. Дано два речення, слова в яких розділені пробілами або комами. Кожне речення закінчується крапкою. Сформуванати множину слів, які є в обох реченнях.
25. Дано послідовність символів, яка закінчується крапкою. Надрукувати всі латинські букви, які є в даній послідовності.

### Завдання 13.2

Сконструювати множину елементами якої є члени вищенаведеної послідовності, вивести її на екран, передбачити введення з клавіатури будь-якого нового елемента та визначення належності даного елемента новоутвореній множині.

#### Приклад 13.2

Цифри від "0" до "9".

#### Розв'язання:

Uses crt;

Var se :set of char;  
i,n:char;

{Опис множини}

Begin

Writeln(' створення множини з цифр ');

se:=[];

While n<>'!' do

Begin

Readln(n);

se:=se+[n];

End;

Write(' вивід множини ');

For i:='0' to '9' do

If i in se then write(i,' ');

{Визначення належності елемента множини}

Readln;

End.

#### Варіанти завдань для самостійної роботи:

- Літери від 'a' до 't'.
- Множина знаків арифметичних операцій.
- Літери від 'в' до 'к' та від 'р' до 'я'.
- Множина чисел Армстронга, що складаються не більше, ніж із чотирьох цифр. (Число Армстронга – це число із  $k$  цифр, для якого

сума  $k$ -х степенів його цифр рівна самому числу, наприклад:  
 $153 = 1^3 + 5^3 + 3^3$ ).

5. Літери від 'а' до 'з' і цифри від '0' до '5'.
6. Знаки арифметичних операцій і букви від 'в' до 'р'.
7. Множина кольорів райдуги.
8. Множина натуральних чисел від 1 до 1000, які співпадають з останніми розрядами своїх квадратів, наприклад:  $25^2 = 625$ ;  $76^2 = 5676$ .
9. Цифри від '0' до '3' і знаки арифметичних операцій.
10. Знаки арифметичних операцій та розділові знаки.
11. Задано натуральне число  $n$ . Серед чисел  $n$ ,  $n+1$ , ...,  $2n$  утворити множину “близнюків” – простих чисел, різниця між якими рівна двом.
12. Знаки операцій відношення та знаки арифметичних операцій.
13. Множина натуральних чисел.
14. Розділові знаки і знаки операцій відношення.
15. Множина всіх чисел від 0 до 9999, сума цифр яких дорівнює 10.
16. Дано натуральні числа  $k$ ,  $m$ ,  $n$ , символи  $s_1, \dots, s_k$ ,  $t_1, \dots, t_m$ ,  $u_1, \dots, u_n$ .  
Утворити множину тих символів, які входять одночасно у всі три послідовності.
17. Літери від 'л' до 'с' та знаки арифметичних операцій.
18. Множина римських цифр.
19. Множина досконалих чисел, що не перевищують заданого  $n$ .  
(Натуральне число називається досконалим, якщо воно рівне сумі всіх своїх простих дільників, наприклад:  $6 = 1 + 2 + 3$ ).
20. Цифри від '0' до '3' і букви від 'а' до 'к'.
21. Множина  $n$  перших членів послідовності Хеммінга. (Послідовністю Хеммінга називається послідовність чисел, які не мають інших дільників, крім 2, 3 і 5, і розташовані в порядку зростання).
22. Знаки арифметичних операцій і цифри від '0' до '7'.
23. Множина назв математичних функцій.
24. Знаки операцій відношення та літери від 'd' до 'z'.
25. Дано натуральне число  $n$ . Утворити множину перших  $n$  рядків трикутника Паскаля. (Трикутником Паскаля називається числовий трикутник, в якому по краях стоять одиниці, а кожне число всередині рівне сумі двох чисел, що стоять над ним в найближчому рядку зверху).

### Завдання 13.3

#### Приклад 13.3

З клавіатури вводиться слово. Надрукувати в порядку, зворотному до алфавітного, всі приголосні літери, що не входять в задане слово.

## Розв'язання:

Uses crt;

Label m1;

Const y: array[1..6] of char=('a', 'y', 'e', 'u', 'i', 'o');

Var q, S, G: set of char;

*{Опис множини}*

n, I: char;

j: integer;

**BEGIN**

**Writeln**(' Введіть слово латинськими літерами. ');

**Writeln**(' Для закінчення вводу введіть символ крапку. ');

S:=[];

*{Введення множини}*

**While true do begin**

**Read**(n);

**If** n = '.' **then goto** m1;

S:=S+[n];

**End**;

**m1: writeln**;

q:=[];

*{Знаходження всіх букв, що не входять в задане слово}*

**For** I='a' **to** 'z' **do begin**

**If not** (I in S) **then include**(q, I);

**End**;

*{Видалення всіх голосних букв}*

**For** j:=1 **to** 6 **do exclude**(q, y[j]);

*{Вивід всіх приголосних букв, що не входять в задане слово}*

**Writeln**('Результат:');

**For** I='z' **downto** 'a' **do**

**If** I in q **then write**(I);

**Readln**;

**END.**

### *Варіанти завдань для самостійної роботи:*

1. Користуючись методом "решето Ератосфена", надрукувати всі прості числа від 1 до 255.
2. З клавіатури вводиться послідовність символів, що закінчується крапкою. Підрахувати загальну кількість голосних букв.
3. Дано дві довільні множини. Написати програму для виконання операцій над множинами (об'єднання, переріз, різниця) в діалоговому режимі.
4. З клавіатури вводиться довільний символ. Програма повинна видавати повідомлення, яке вказує, до якого діапазону належить символ (малі

латинські літери, великі латинські літери, літери кирилиці, цифри, розділові знаки, спеціальні символи) або не належить до жодного.

5. Дано рядок з довільних символів. Визначити, яких літер більше – голосних чи приголосних.
6. Дано множину  $S$ , в яку входять цілі числа від 1 до 100. Визначити, скільки елементів довільної множини  $D$  належить множині  $S$ .
7. Дано множину  $V$ , в яку входять малі латинські літери від 'a' до 'z'. Вивести в алфавітному порядку всі елементи довільної множини  $A$ , що входять в множину  $V$ .
8. Дано довільне п'ятизначне число. Вивести в спадаючому порядку всі цифри, що не входять в запис даного числа.
9. Утворити множину всіх чотиризначних чисел, сума цифр яких дорівнює їхньому добутку.
10. Дано масив довільних дійсних чисел, пронумерованих від 1 до 10. Визначити суму елементів масиву, індекси яких належать множині  $Mn1$  і добуток елементів, індекси яких належать множині  $Mn2$ , де  $Mn1, Mn2: \text{set of } 1..10$ ;
11. Дано множину перших ста натуральних чисел. Утворити множину  $P$  – простих чисел та множину  $S$  – складних чисел з цього діапазону.
12. Підрахувати кількість слів у тексті, якщо між словами є лише один розділовий знак (',', '.', ':', ';', ' ' і т. д.)
13. З клавіатури вводиться текст, слова в якому розділені одним або декількома розділовими знаками. Скласти програму, яка б друкувала всі слова.
14. Дано множину 20 – ти довільних символів. Визначити, скільки серед них є латинських літер, цифр, розділових знаків та інших символів і надрукувати їх групами.
15. Дано масив `St:array[1..10] of string[25]`. Вивести слово, в якому найбільше голосних букв.
16. В класі навчається 25 учнів. Кожному учню були виставлені оцінки за семестр з 15 предметів. Визначити, скільки в класі відмінників.
17. Дано масив `St:array[1..10] of string[25]`. Надрукувати всі слова в порядку зростання в них кількості голосних букв.
18. Нехай є описання:  
Type  $Mova$ =(російська, англійська, німецька, французька, іспанська, італійська, китайська);  
 $Znannya$ =set of  $Mova$ ;  
 $Students$ =array[1..5] of  $Znannya$ ;  
Вивести на екран іноземну мову, якою: а) володіють всі студенти; б) не володіє жоден студент.
19. В зростаючому порядку надрукувати всі цілі числа в діапазоні 1..10000, які можна подати у вигляді  $n^2 + m^2$ , де  $n, m \geq 0$ .
20. Нехай є описання:

Type Tovar=(телевізори, холодильники, електроплити, пралмашини, відеомагнітофони, відеокамери, аудіоплеєри);

Asortiment=set of Tovar;

magazin=array[1..4] of Asortiment;

Утворити та вивести на екран множину назв товарів, які є на полицях:

а) всіх магазинів; б) хоча б одного магазину.

21. Нехай є описання:

Type Predmet=(вища\_мат, програмування, історія, чис\_методи);

Nezdav=set of Predmet;

Student=array[1..6] of Nezdav;

В групі є шість студентів, які не здали іспити зимової сесії. Вияснити, чи є серед них такі студенти, які не здали жодного іспиту.

22. Дано множину цілих чисел від 1 до 100. Визначити, скільки серед них чисел Фібоначчі і скільки чисел, перша значуща цифра в десятковому записі яких 1 або 2.

23. В порядку спадання надрукувати всі цілі числа в діапазоні 1..4900, які можна представити у вигляді  $n^2 + 2k^2$ , але не можна представити у вигляді  $7ij + j + 3$  ( $n, k, i, j \geq 0$ ).

24. Нехай є описання:

Type Prizv=(Петрук, Кірик, Тарасюк, Степанюк, Литвинов, Пожарська);

Gosti=set of Prizv;

Grupa=array[Prizv] of Gosti;

Визначити, чи є в групі людина, яка побувала в гостях у всіх інших осіб групи.

25. В ресторані наявні окремі меню на сніданок, обід та вечерю. Відомо, що в кожному з цих меню не більше 10 видів страв. Визначити, які види страв присутні в меню і на сніданок, і на обід, і на вечерю, якщо такі взагалі є. Визначити види страв, які є тільки на сніданок, тільки на обід, тільки на вечерю.

### Контрольні запитання:

1. Що розуміють під множиною в ТР?
2. Як здійснюється опис множини в ТР?
3. Дайте визначення елемента множини.
4. Чи може множина містити елементи різних типів?
5. Який об'єм пам'яті займає один елемент множини?
6. Як позначається порожня множина?
7. Що таке потужність множини?
8. Які операції можна здійснювати над множинами?
9. Яка потужність множини, базовий тип якої *boolean*?
10. Як працює операція *in*?

## Лабораторна робота №14

### Тема: Робота в графічному режимі

**Мета роботи:** Навчитися працювати в графічному режимі, ознайомитися з основними графічними процедурами та функціями.



Національний університет  
водного господарства  
та природокористування

#### Хід роботи:

1. Розробити алгоритм розв'язку поставленої задачі для побудови графічних зображень.
2. Написати відповідну програму на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.

#### Завдання 14.1

Побудувати графік функції  $y = f(x)$ , заданої на відрізку  $[a, b]$  (лабораторна робота №4, завд. 4.1).

#### Приклад 14.1

Побудувати графік функції  $y = \frac{\cos^2 x}{x^2 + 1}$ ,  $0,5 \leq x \leq 7,6$ ;  $\Delta x = 0,02$ .

#### Розв'язання:

Uses Graph, crt;

{Підключення графічної бібліотеки}

Var Gd, Gm: Integer;

x, y, x2, x1, y2, y1: real;

s: string;

Function f(x: real): real;

{Обчислення аргумента}

Begin

f:=sqr(cos(x))/(x\*x+1);

End;

Begin

Gd := Detect;

InitGraph(Gd, Gm, 'd:\bp\bgi\');

{Ініціалізація графічного режиму}

If GraphResult <> grOk then

Halt(1);

Cleardevice;

{Очистка графічного екрана}

SetColor(3);

{Встановлення кольору зображення}

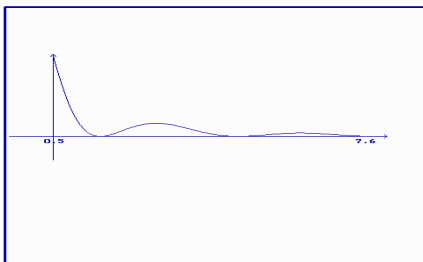


```

Settextstyle(1,0,4);                               {Встановлення стилю тексту}
Outtextxy(5,getmaxy div 2-50,'Grafic function
f(x)=cos(x*x)/(x*x+1)');
Delay(10000);
Delay(50000);
Cleardevice;
Setcolor(red);                                     {Побудова осей координат}
Setlinestyle(0,0,3);                               {Встановлення стилю ліній}
Rectangle(20,20,500,getmaxy-20);
Setcolor(13);
Setlinestyle(0,0,1);
Line(25,getmaxy div 2,455,getmaxy div 2);
Line(75,100,75,getmaxy-200);
Moveto(75,100);
Lineto(72,104);
Moveto(75,100);
Lineto(78,104);
Moveto(455,getmaxy div 2);
Lineto(452,getmaxy div 2+3);
Moveto(455,getmaxy div 2);
Lineto(452,getmaxy div 2-3);
Settextstyle(0,0,0);
Setcolor(green);
x2:=0.5;
y2:=f(x2);
Moveto(round(x2*50+50),round(getmaxy div 2-(y2*220)));
While x2<=7.6 do begin                               {Побудова графіка}
  x1:=x2+0.02;
  y1:=f(x1);
  Lineto(round(x1*50+50),round(getmaxy div 2-(y1*220)));
  x2:=x1;
End;
Settextstyle(0,0,0);
Setcolor(13);
Outtextxy(65,getmaxy div 2+5,'0.5');
Outtextxy(420,getmaxy div 2+5,'7.6');
Readln;
CloseGraph;                                       {Завершення роботи з графічним режимом}
End.

```

Результат:





## Завдання 14.2

Скласти програму побудови на екрані монітора елементарних фігур. Малюнок підписати, вказавши його назву та прізвище виконавця.

### Приклад 14.2

Побудувати куб (сторону задати довільно).

#### Розв'язання:

```
Uses Crt, Graph;                                {Підключення графічної бібліотеки}
Var Gd, Gm, r, xm, ym: Integer;
Begin
  Clrscr;
  Write(' line ? ');
  Readln(r);
  Gd := Detect;
  InitGraph(Gd, Gm, 'd:\bp\bgi\');               {Ініціалізація графічного режиму}
  If GraphResult <> grOk then
    Halt(1);
  Cleardevice;                                   {Очистка графічного екрана}
  Setcolor(2);                                   {Встановлення кольору зображення}
  Settextstyle(1,0,6);                           {Встановлення стилю тексту}
  Outtextxy(15,25,'kubic zav_1 ');
  Setcolor(4);
  ym:=getmaxy;                                   {Встановлення значення максимальної координати
                                              по вертикалі}
  Moveto(0,ym);                                  {Переміщення вказівника в задану точку}
  Rectangle(20,ym-(20+r),20+r,ym-20);           {Побудова куба}
  Moveto(20,ym-(20+r));
  Lineto(20+r div 2,ym-(20+r)-r div 2);
  Lineto(20+r div 2+r,ym-(20+r)-r div 2);
  Lineto(20+r,ym-(20+r));
  Moveto(20+r,ym-20);
  Lineto(20+r div 2+r,ym-20-r div 2);
  Lineto(20+r div 2+r,ym-(20+r)-r div 2);
```

```

Moveto(20,ym-20);
Setlinestyle(3,0,1);
Lineto(20+r div 2,ym-20-r div 2);
Lineto(20+r div 2+r,ym-20-r div 2);
Moveto(20+r div 2,ym-20-r div 2);
Lineto(20+r div 2,ym-20-r div 2-r);
CloseGraph;

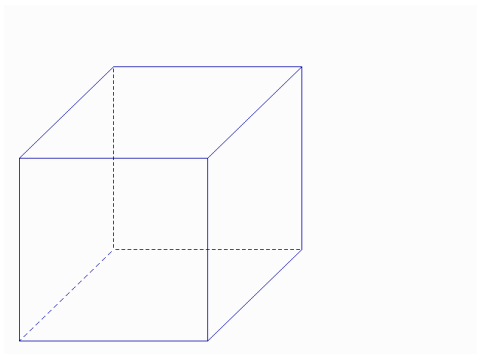
```

*{Завершення роботи з графічним режимом}*

**End.**



**Результат:** верситет  
водного господарства  
та природокористування



### ***Варіанти завдань для самостійної роботи:***

1. Побудувати тетраедр.
2. Побудувати ромб і вписати в нього коло.
3. Побудувати правильну шестикутну піраміду.
4. Побудувати конус з осьовим перерізом.
5. В циліндр вписати конус.
6. Побудувати фужер.
7. Побудувати кільця Сатурна.
8. Побудувати секторну діаграму.
9. Побудувати об'ємну стовпчикову діаграму.
10. Зобразити шахову дошку для 100-клітинкових шашок.
11. Зобразити систему координат з нанесеним масштабом.
12. Зобразити ракету.
13. Зобразити профіль легкового автомобіля.
14. Зобразити сузір'я Великої Ведмедиці.
15. Зобразити дифракційну решітку проходження світла через збірну лінзу.
16. Зобразити план студентського містечка.
17. Зобразити водонапірну башту.
18. Зобразити електричне поле між двома різнойменними точковими зарядами.
19. Зобразити силові лінії магнітного поля соленоїда.
20. Намалуйте схему електричного поля, в яке ввімкнені електрична лампочка-перемикач і батарея.

21. Намалуйте хід променя в трикутній призмі.
22. Намалуйте кільця Ньютона.
23. Намалуйте хід променя через збірну і розсіювальну лінзу.
24. На площині зобразити сітку криволінійних координат, сполучивши сусідні точки відрізками (рис. 1).
25. Зобразити павутину (рис 2).



Національний університет  
водного господарства  
та природокористування

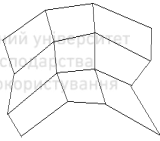


Рис. 1.

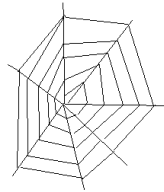


Рис.2.

### Завдання 14.3

Скласти програму зображення рухомого об'єкта. Передбачити звукові ефекти під час виконання програми.

#### Приклад 14.3

Постріл з луку по мішені.

#### Розв'язання:

```

Uses Crt, Graph; {Підключення графічної бібліотеки}
Var Gd, Gm,r,xm,ym,m,size,i: Integer;
    p:pointer;
Begin
  Clrscr;
  Gd := Detect;
  InitGraph(Gd, Gm, 'd:\bp\bgi\'); {Ініціалізація графічного режиму}
  If GraphResult <> grOk then
    Halt(1);
  Cleardevice; {Очистка графічного екрана}
  Setbkcolor(3); {Встановлення кольору фону}
  m:=-1;
  For r:=5 downto 1 do begin
    m:=-m;
    Ellipse(getmaxx-90,getmaxy div 2,0,360,8*r,10*r); {Побудова еліпса}
    If m<0 then setfillstyle(1,1) else setfillstyle(1,4);
    Floodfill(getmaxx-90,getmaxy div 2,15);
  End;
  Setcolor(1); {Встановлення кольору зображення}
  Setlinestyle(0,0,3); {Встановлення стилю ліній}

```

```

Line(getmaxx-90,getmaxy div 2-50,getmaxx-90,0);
Setlinestyle(0,0,3);
Ellipse(0,getmaxy div 2,335,25,90,90);
Moveto(80,200);Setlinestyle(0,0,1);
Lineto(80,280);Setcolor(1);
Moveto(75,240);Lineto(120,240);
Delay(26000);

```



```

Setcolor(3);
Moveto(80,200);Lineto(80,280);
Setcolor(1);
Ellipse(170,getmaxy div 2,163,197,95,120);
Sound(1207);
Delay(15000);
Setcolor(3);
Ellipse(170,getmaxy div 2,163,197,95,120);
Setcolor(1);Nosound;
Moveto(80,200);Lineto(80,280);
Sound(8800);
Size:=imagesize(75,240,120,240);

```

```

Getmem(p,size);
Getimage(75,240,120,240,p^);

```

```

For i:=75 to 500 do begin
Delay(140);

```

```

Putimage(i,240,p^,xorput);
End;

```

```

Nosound; Sound(500);
Delay(9000); Nosound;

```

```

Settextstyle(1,0,25);

```

```

Outtextxy(10,20,'!#good#!');
Setcolor(0);

```

```

Circle(getmaxx-90,getmaxy div 2,2);
Sound(550);Sound(500);Delay(9000);

```

```

Nosound;
Readln;

```

```

Nosound;
CloseGraph;

```

*{Завершення роботи з графічним режимом}*

**End.**

### ***Варіанти завдань для самостійної роботи:***

1. Броунівський рух кольорових частинок.
2. Робота вітряка (клавіші → та ← повинні керувати напрямком вітру).

3. Рух годинникових стрілок.
4. Рух сонячної системи.
5. Авторалі.
6. Гру “кішки-мишки”.
7. Пісочний годинник.
8. Рух рибок в акваріумі.
9. Коливання математичного маятника.
10. Рух людини по парку (клавіші →, ←, ↑, ↓ - повинні керувати рухом людини).
11. Рух пливки в кінопроекторі.
12. Гру “хрестики-нулики”.
13. Рух Землі навколо Сонця.
14. Спуск лижника з гори.
15. Рух автомобіля по гірській дорозі.
16. Запис інформації на диск.
17. Пішохідний перехід.
18. Хмарку, з якої йде дощ, а потім з’являється райдуга.
19. Рух трамваю по колу.
20. Рух фізичного маятника.
21. Політ метелика.
22. Гру “баскетбол”.
23. Рух потяга від станції А до станції В (передбачити рух крізь тунель).
24. Рух черепашки по лабіринту.
25. Корабель, що тоне.

#### Завдання 14.4.

Побудувати рухоме графічне зображення.

#### Приклад 14.4

Перемикання світлофора.

#### Розв’язання:

```
Uses Graph, Crt;
Var grDriver: Integer;
    grMode: Integer;
    ErrCode: Integer;
    res: Integer;
    i, j: Integer;
```

*{Підключення графічної бібліотеки}*

*{Лічильники циклів}*

*{Процедура, що малює коло заданого кольору}*  
*{x, y, r – координати центра та радіус кола}*

*{fc, bc – параметри, що задають колір кола}*

**Procedure Krug** (x, y, r: Integer; fc, bc: Integer);

**Begin**

**SetFillStyle**(SolidFill, fc);

**SetColor**(fc);

**PieSlice**(x, y, 0, 360, r);

**SetColor**(bc);

**Circle**(x, y, r);

**End;**

**Begin**

*{Основна програма}*

grDriver:=detect;

**InitGraph**(grDriver, grMode, 'e:\tp\bgi'); *{Ініціалізація графічного режиму}*

ErrCode:=GraphResult;

**if** ErrCode = grOk **then**

**begin**

**OutTextXY**(10, 10, 'Дотримуйтесь правил дорожнього руху!');

**Rectangle**(88, 88, 112, 152);

*{Ввімкнено червоний колір}*

**Krug**(100, 100, 10, Red, White);

**Krug**(100, 120, 10, LightGray, White);

**Krug**(100, 140, 10, LightGray, White);

**For** i:=1 **to** 3 **do**

**Begin**

*{Тут горить червоний}*

**Delay**(3000);

*{Затримка 3 секунди}*

**Krug**(100, 120, 10, Yellow, White);

*{Ввімкнути жовтий}*

**Delay**(1000);

**Krug**(100, 100, 10, LightGray, White);

*{Вимкнути червоний}*

**Krug**(100, 120, 10, LightGray, White);

*{Вимкнути жовтий}*

**Krug**(100, 140, 10, Green, White);

*{Ввімкнути зелений}*

**Delay**(2000);

*{Мигаючий зелений сигнал}*

**For** j:=1 **to** 3 **do**

*{Мигає 5 разів}*

**Begin**

**Delay**(500);

**Krug**(100, 140, 10, Green, White);

*{Ввімкнути зелений}*

**Delay**(500);

**Krug**(100, 140, 10, LightGray, White);

*{Вимкнути зелений}*

**End;**

**Krug**(100, 120, 10, Yellow, White);

*{Ввімкнути жовтий}*

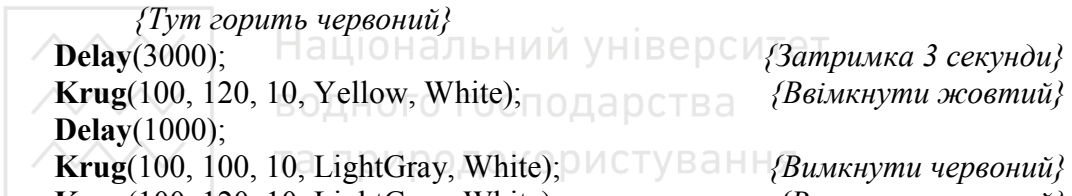
**Delay**(1500);

**Krug**(100, 120, 10, LightGray, White);

*{Вимкнути жовтий}*

**Krug**(100, 100, 10, Red, White);

*{Ввімкнути червоний}*



**End; End;**  
**CloseGraph;**  
**End.**

{Завершення роботи з графічним режимом}

### **Варіанти завдань для самостійної роботи:**

1. Стрілочний (чи електронний) годинник, що показує реальний час. Використати системний годинник комп'ютера.
2. Рух снаряда, який вилетів з гармати.
3. Пружний удар двох кульок.
4. Непружний удар двох кульок.
5. Калюжу (вид зверху) і краплі дощу на воді.
6. Новорічна ялинка з гірляндами. Місце розташування та колір кульок генерувати випадковим чином.
7. Падіння яблука з дерева.
8. Екран радіолокатора кругового огляду. Поява цілі повинна супроводжуватись звуковим сигналом.
9. Екран осцилографа, на якому відображаються фігури, задані наступними рівняннями:  $X=a*\sin(b*t)$ ;  $Y=c*\sin(d*t)$ , де  $a,c$  – амплітуди коливань;  $b,d$  – частоти;  $t$  – час.
10. Двосторонній рух автомобілів по вулиці.
11. Літак, що піднімається в повітря.
12. Рух човна, що пливе по морю в шторм.
13. Настільна лампа, яка вмикається по натисканню клавіші і освітлює кімнату.
14. Біжучий по горизонталі рядок тексту.
15. Рух супутника по орбіті Землі.
16. Секундомір.
17. Захід сонця.
18. Молоток, що забиває цвях в дошку.
19. Візок, що спускається по похилій площині.
20. Зорепад.
21. Біжучі по вертикалі рядки символів.
22. Феєрверк.
23. Привітальна листівка до свята Нового року з динамічними елементами (мигаючою гірляндою, надписами, салютом).
24. Проростання колоска із зернини.
25. Мурашник.

### **Завдання 14.5**



Побудувати графік функції в полярних системах координат  $\rho = f(x)$ .

### Приклад 14.5

Побудувати графік функції, заданої в полярних системах координат  $\rho = 2r \cos \varphi + 2$ .

#### Розв'язання:

Національний університет  
водного господарства  
та природокористування

**Uses** Graph,crt;

*{Підключення графічної бібліотеки}*

**Const** p=pi\*pi/360;

**Var** Gd, Gm,x,x1,x2: Integer;

y,y2,y1,r:real;

s:string;

**Function** f(x:real):real;

*{Обчислення аргумента}*

**Begin**

f:=(2\*r\*cos(x)+2)\*cos(x);

**End;**

**Begin**

**Clrscr;**

**Writeln**(' введіть r :');

**Readln**(r);

Gd := Detect;

**InitGraph**(Gd, Gm, 'd:\program\bp\bgi');

*{Ініціалізація графічного режиму}*

**If** GraphResult <> grOk **then**

**Halt**(1);

**Cleardevice;**

*{Очистка графічного екрана}*

**Setcolor**(3);

*{Встановлення кольору зображення}*

**Settextstyle**(1,0,4);

*{Встановлення стилю ліній}*

**Outtextxy**(5,getmaxy div 2-50,' Grafic function f(x)=2rcosx+2');

**Delay**(20000);

**Delay**(50000);

**Cleardevice;**

**Setcolor**(red);

**Setlinestyle**(0,0,3);

**Rectangle**(20,20,getmaxx-20,getmaxy-20);

*{Побудова прямокутника}*

**Setcolor**(13);

**Setlinestyle**(0,0,1);

**Line**(25,getmaxy div 2,getmaxx-25,getmaxy div 2);

**Line**(getmaxx div 2,150,getmaxx div 2,getmaxy-150);

**Moveto**(getmaxx div 2,150);

**Lineto**(getmaxx div 2-3,154);

```

Moveto(getmaxx div 2,150);
Lineto(getmaxx div 2+3,154);
Moveto(getmaxx-25,getmaxy div 2);
Lineto(getmaxx-28,getmaxy div 2+3);
Moveto(getmaxx-25,getmaxy div 2);
Lineto(getmaxx-28,getmaxy div 2-3);
Settextstyle(0,0,0);
Outtextxy(getmaxx div 2-10,getmaxy div 2-10,'0');

```



```

H:
вог
та
SetColor(green);
For x:=-280 to 280 do begin
    x1:=round((x+318)*sin(x));
    y1:=f(x*p);
    x2:=round((x+319)*sin(x));
    y2:=f(p*(x+1));
    Line(x1,round(getmaxy div 2-(y1*15)),x2,round(getmaxy div 2-
(y2*15)));
    Delay(50);
    End;
Readln;
CloseGraph;
End.

```

*{Побудова графіка}*

*{Завершення роботи з графічним режимом}*

### **Варіанти завдань для самостійної роботи:**

1.  $\rho = a \sin 2\varphi$ ;
2.  $\rho^2 = a^2 \cos 2\varphi$ ;
3.  $\rho = atg\varphi$ ;
4.  $\rho = a\varphi + 2$ ;
5.  $\rho = a^\varphi$ ;
6.  $\rho = a(1 + \cos \varphi)$ ;
7.  $\rho = a \sin \varphi$ ;
8.  $\rho = 2r \sin \varphi$ ;
9.  $\rho = \frac{a}{\varphi}$ ;
10.  $\rho = a(\varphi - \sin \varphi)$ ;
11.  $\rho = a^2 - tg\varphi$ ;
12.  $\rho = (r \cos \varphi + r \sin \varphi)$ ;
13.  $\rho = 1 + \ln \varphi$ ;
14.  $\rho = 2a \cos \varphi$ ;

$$15. \rho = \frac{a}{\cos(\varphi - \pi/3)};$$

$$16. \rho = \frac{a^3}{3} \cos \varphi;$$

$$17. \rho = 4a(1 - \cos \varphi);$$

$$18. \rho = \cos \frac{\pi}{3} \varphi;$$

$$19. \rho = \sin \varphi \cos \varphi.$$

$$20. \rho = \operatorname{ctg} \varphi + \operatorname{cosec} \varphi.$$

$$21. \rho = a \sec^2 \frac{\varphi}{2}.$$

$$22. \rho = 2a \cos 3\varphi.$$

$$23. \rho = a \sin^3 \frac{\varphi}{3}.$$

$$24. \rho = 5 + \cos \varphi.$$

$$25. \rho^2 = a^2 \cos^2 \frac{\varphi}{2}.$$

### Завдання 14.6

Побудувати графік функції, заданої в параметричній формі  $x = x(t)$ ,  $y = y(t)$  ( $a = \text{const}$ ,  $a > 0$ ).

#### Приклад 14.6

Побудувати графік функції, заданої в параметричній формі

$$\begin{cases} x = a(2 \cos t - \cos 2t) \\ y = a(2 \sin t - \sin 2t) \end{cases}$$

#### Розв'язання:

Uses Graph,crt;

{Підключення графічної бібліотеки}

Const a=2;

p=pi\*pi/360;

Var Gd, Gm,t: Integer;

x,y,x2,x1,y2,y1:real;

Function fx(t:real):real;

Begin

fx:=a\*(2\*cos(t)-cos(2\*t));

End;

**Function** fy(t:real):real;

**Begin**

fy:=a\*(2\*sin(t)-sin(2\*t));

**End;**

**Begin**

Gd := Detect; *{Автоматичне розпізнавання апаратури}*

**InitGraph**(Gd, Gm, 'd:\bp\bgi\'); *{Ініціалізація графічного режиму}*

**If GraphResult** <> grOk **then**

**Halt**(1); *{Зупинка використання}*

**Cleardevice**; *{Очистка графічного екрану}*

**Setcolor**(3); *{Встановлення кольору зображення}*

**Settextstyle**(1,0,4); *{Встановлення стилю ліній}*

**Outtextxy**(5,getmaxy div 2-50,' Grafic parametrition function ');

**Delay**(20000);**Delay**(50000);

**Cleardevice**;**Setcolor**(red);

**Setlinestyle**(0,0,3);

**Rectangle**(20,20,getmaxx-20,getmaxy-20); *{Побудова області зображення}*

**Setcolor**(13);

**Setlinestyle**(0,0,1);

**Line**(25,getmaxy div 2,getmaxx-25,getmaxy div 2); *{Зображення ліній}*

**Line**(getmaxx div 2,150,getmaxx div 2,getmaxy-150);

**Moveto**(getmaxx div 2,150); *{Переміщення вказівника в задану точку}*

**Lineto**(getmaxx div 2-3,154);

**Moveto**(getmaxx div 2,150);

**Lineto**(getmaxx div 2+3,154);

**Moveto**(getmaxx-25,getmaxy div 2);

**Lineto**(getmaxx-28,getmaxy div 2+3);

**Moveto**(getmaxx-25,getmaxy div 2);

**Lineto**(getmaxx-28,getmaxy div 2-3);

**Settextstyle**(0,0,0);

**Outtextxy**(getmaxx div 2-10,getmaxy div 2-10,'0');

**Moveto**(round(x2\*50+50),round(getmaxy div 2-(y2\*220)));

**Setcolor**(green);

**For** t:=-0 **to** 360 **do begin** *{Побудова графіка}*

x1:=fx(t\*p);

y1:=fy(t\*p);

x2:=fx((t+1)\*p);

y2:=fy((t+1)\*p);

**Line**(round(x1\*20+320),round(getmaxy div 2-(y1\*20)),round(x2\*20+320),round(getmaxy div 2-(y2\*20)));

**Delay**(300);

**End;**

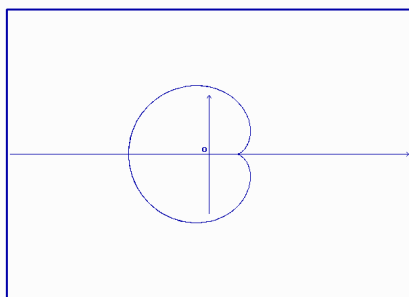
ReadIn;  
CloseGraph;  
End.

{Завершення роботи з графічним режимом}

Результат:



Національний університет  
водного господарства  
та природокористування



**Варіанти завдань для самостійної роботи:**

1. 
$$\begin{cases} x = 3t^2, \\ y = 3t - t^3. \end{cases}$$

2. 
$$\begin{cases} x = a \cos^3 t, \\ y = a \sin^3 t. \end{cases}$$

3. 
$$\begin{cases} x = t^2 - 1, \\ y = t^3 - t. \end{cases}$$

4. 
$$\begin{cases} x = r(\cos t + t \sin t), \\ y = r(\sin t - t \cos t). \end{cases}$$

5. 
$$\begin{cases} x = t^2, \\ y = t - \frac{t^3}{3}. \end{cases}$$

6. 
$$\begin{cases} x = a \cos t, \\ y = a \sin t. \end{cases}$$

7. 
$$\begin{cases} x = a(t - \sin t), \\ y = a(1 - \cos t). \end{cases}$$

8. 
$$\begin{cases} x = 2a \sin t, \\ y = 2a \cos t. \end{cases}$$

9. 
$$\begin{cases} x = 4(t - 2 \sin t), \\ y = 4(1 - 2 \cos t). \end{cases}$$

10. 
$$\begin{cases} x = \cos t - 2 \cos 2t, \\ y = \sin t - 2 \sin 2t. \end{cases}$$

11. 
$$\begin{cases} x = 3t^2, \\ y = 6t - 2t^3. \end{cases}$$

12. 
$$\begin{cases} x = 2 \cos^3 t, \\ y = 2 \sin^3 t. \end{cases}$$

13. 
$$\begin{cases} x = 2(t^2 - 1), \\ y = 2(t^3 - t). \end{cases}$$

14. 
$$\begin{cases} x = 7(\cos t + t \sin t), \\ y = 7(\sin t - t \cos t). \end{cases}$$

15. 
$$\begin{cases} x = t^2 + 4, \\ y = t - \frac{t^3}{3} + 2. \end{cases}$$

16. 
$$\begin{cases} x = 5a \cos t, \\ y = 5a \sin t. \end{cases}$$

17. 
$$\begin{cases} x = 2a(t - \sin t), \\ y = 2a(1 - \cos t). \end{cases}$$

18. 
$$\begin{cases} x = 2a \sin t + 4, \\ y = 2a \cos t + 6. \end{cases}$$

$$19. \begin{cases} x = 4a(t - 2 \sin t), \\ y = 4a(1 - 2 \cos t). \end{cases}$$

$$20. \begin{cases} x = e^t \cos t, \\ y = e^t \sin t. \end{cases}$$

$$21. \begin{cases} x = \frac{3at}{1+t^3}, \\ y = \frac{3at^2}{1+t^3}. \end{cases}$$

$$22. \begin{cases} x = \sqrt{t}, \\ y = \sqrt[3]{t}. \end{cases}$$

$$23. \begin{cases} x = a(\operatorname{sh} t - t), \\ y = a(\operatorname{ch} t - 1). \end{cases}$$

$$24. \begin{cases} x = 2t + 3t^2, \\ y = t^2 + 2t^3. \end{cases}$$

$$25. \begin{cases} x = t \cdot e^t, \\ y = t \cdot e^{-t}. \end{cases}$$



Національний університет  
водного господарства  
та природокористування



Національний університет  
водного господарства  
та природокористування

### **Контрольні запитання:**

1. Яка відмінність текстового та графічного режимів роботи монітора?
2. Що таке відеопам'ять?
3. Яке призначення відеоадаптера та відеоконтролера?
4. Назвіть основні принципи програмування графіки.
5. За допомогою якої процедури здійснюється ініціалізація графічного режиму роботи дисплея?
6. Для чого призначені модулі Graph та Graphs3d?
7. Назвіть основні процедури та функції для роботи з графікою.
8. Як здійснюється вивід тексту в графічному режимі?
9. В чому полягає “алгоритм художника”?
10. Для чого використовуються спрайти?
11. Як побудувати рухоме зображення?



Національний  
університет  
водного господарства  
та природокористування

## **Лабораторна робота №15**

### **Тема: Використання модулів**

**Мета роботи:** Ознайомлення студентів з використанням модулів, особливостями їх побудови, вивчення методів компіляції.

### **Хід роботи:**

1. Розробити алгоритм розв'язку поставленої задачі з використанням модулів.
2. Написати відповідну програму на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.

### **Завдання 15.1**

Оформити підпрограми з теми “Програмування з використанням процедур та функцій в програмах” (лабораторна робота № 9, завд.1) у вигляді окремого модуля. З використанням модуля виконати поставлене завдання.

### Вимоги до програми:

Створений модуль має містити процедури та функції вашої програми.

Основна програма повинна містити:

- 1) Введення вихідних даних;
- 2) Виклик процедур та функцій;
- 3) Обчислення заданого виразу та вивід результатів на екран.



Національний університет  
водного господарства  
та природокористування

#### Приклад 15.1

Знайти значення виразу

$$S = \frac{\sqrt{\left(\sum_{i=1}^5 t_i\right)^2 + A^2}}{\sqrt{A^2 + B^2}} + \frac{\sqrt{B^2 + \left(\sum_{i=1}^7 t_i\right)^2}}{\sqrt{\left(\sum_{i=1}^6 t_i\right)^2 + C^2}},$$

де  $\{t_i\} = (5,2; -7,1; 8,3; -3,4; 7,5; -8,4; 6,0; 5,2; 1,9; 4,3)$ ;

$A = 7,6$ ;  $b = -8,9$ ;  $C = 3,65$ .

#### Розв'язання:

```
Uses crt,p1;
```

```
Const t:array [1..10] of real=(5.2,-7.1,8.3,-3.4,7.5,-8.4,6.0,5.2,1.9,4.3);  
a=7.6;b=-8.9;c=3.65;
```

```
Var i:integer;  
s1, s2, s3, s, v1, v2, v3, v4:real;
```

```
Begin
```

```
Clrscr;
```

```
suma(t,1,5,s1);
```

```
korin(s1,a,v1);
```

```
korin(a,b,v2);
```

```
suma(t,1,7,s2);
```

```
korin(b,s2,v3);
```

```
suma(t,1,6,s3);
```

```
korin(s3,c,v4);
```

```
s:=v1/v2+v3/v4;
```

```
Writeln(' Сума s =',s:8:2);
```

```
End.
```

```
Unit p1;
```

```
Interface
```

```
Procedure suma(tt:array of real;p,k:integer;var s5:real);
```

{Підключення модуля P1}

{Опис модуля P1}

{Розділ інтерфейсу}



**Procedure** korin(k1,k2:real;var s5:real);

**Implementation**

**Procedure** suma;

**Var** i:integer;

**Begin**

s5:=0;

**For** i:=p **to** k **do**

s5:=s5+tt[i];

**End;**

**Procedure** korin;

**Begin**

s5:=sqrt(k1\*k1+k2\*k2);

**End;**

**Begin**

**End.**

**Результат:** S=2,44.

*{Розділ реалізації}  
{Опис процедури suma}*

*{Опис процедури korin}*

## Завдання 15.2

Розробити з використанням модуля користувача і стандартного модуля GRAPH програму побудови алгоритму з теми “Алгоритми та програми розгалуженої структури” (лабораторна робота №3, завд. 3.1) та “Циклічні алгоритми та програми” (лабораторна робота №4, завд. 4.1).

### **Вимоги до програми:**

- 1) модуль повинен містити процедури та функції побудови основних блоків блок-схем;
- 2) надписи в блок-схемах повинні вводитися в основній програмі.

## Приклад 15.2

Знайти значення функції, яка обчислюється залежно від значення аргументу

$$y = \begin{cases} \lg \left| \frac{\pi}{16} - x \right|, & 0 < x < \frac{1}{4}; \\ (x^2 - 2,04)^{-3,14}, & \frac{1}{4} < x < 1; \\ \arccos \frac{x}{4}, & x \geq 1. \end{cases}$$

### Розв'язання:

```
Uses crt, Lgarccos; {Підключення модуля Lgarccos}
Var x,y:real;
Begin
  clrscr;
  Read(x); {Обчислення значення виразу}
  If (0<x) and (x<1/4) then y:=Lg(abs(Pi/16-x));
  If (1/4<x) and (x<1) then y:=exp(-3.14*ln(abs(sqrt(x)-2.04)));
  If x>=1 then y:=arccos(x/4);
  Writeln(' y= ', y:6:2);
  Readkey;
End.
Unit Lgarccos; {Опис модуля Lgarccos}
Interface {Розділ інтерфейсу}
  Function Lg(x:real):real;
  Function arccos(x:real):real;
Implementation {Розділ реалізації}
  Function Lg(x:real):real; {Функція обчислення Lg}
  begin
    Lg:=ln(x)/ln(10); readln;
  end;
  Function arccos(x:real):real; {Функція обчислення arccos}
  begin
    arccos:=arctan(sqrt(1-sqr(x))/x);
  end;
END.
```

### Завдання 15.3

Створити модуль користувача та відлагодити його роботу.

### Приклад 15.3

Створити модуль для обчислення значення функції  $tg(x)$  та знаходження суми  $n$  членів арифметичної прогресії.

### Розв'язання:

```
Uses crt, trgp; {Підключення модуля trgp}
Var x,y,a1,s,d,x1:real;
```

```

n:integer;
Begin
  Clrscr;
  Textcolor(green);                                {Задання кольору тексту}
  Write(' введіть кут, тангенс якого потрібно знайти. ');
  Readln(x);
  tanp(x,y);
  Textcolor(red);
  Writeln(' тангенс x= ', y:8:4);
  Writeln(' введіть 1 член алгебраїчної прогресії, крок, кількість членів: ');
  Readln(a1,d,n);
  sumap(a0,d,n,s);
  Textcolor(red);
  Writeln(' сума рівна ', s:9:2);
  Readln;
End.
Unit trgp;                                         {Опис модуля trgp}
Interface
  Procedure tanp(x:real;var tan:real);
  Procedure sumap(a0,d:real;n:integer;var sn:real);
Implementation
  Procedure tanp;                                   {Процедура обчислення тангенса кута}
    Begin
      tan:=sin(x)/cos(x);
    End;
  Procedure sumap;                                  {Процедура обчислення суми n членів арифметичної прогресії}
    Begin
      sn:=(n+1)/2*(2*a0+n*d);
    End;
Begin
End.

```

### ***Варіанти завдань для самостійної роботи:***

1. Модуль для обчислення функцій  $\arcsin(x)$ ,  $\arccos(x)$ ,  $\text{arcctg}(x)$ ,  $\text{sh}(x)$ ,  $\text{ch}(x)$ ,  $\log_b N$ ,  $a^b$ .
2. Модуль для обчислення площ основних геометричних фігур на площині.
3. Модуль для обчислення об'ємів основних просторових тіл.
4. Модуль для роботи з комплексними числами (ввід, вивід комплексного числа, обчислення його модуля, суми та добутку двох комплексних чисел)

5. Модуль для обчислення НСД, НСК та знаходження всіх простих дільників деякого числа.
6. Модуль для обчислення подвійних, потрійних кутів та формул половинного аргументу.
7. Модуль, що реалізує перетворення суми тригонометричних функцій в добутки.
8. Модуль для розв'язання лінійних, квадратних, біквадратних та кубічних рівнянь.
9. Модуль для обчислення медіан, бісектрис та висот довільного трикутника із сторонами  $a, b, c$ .
10. Модуль для знаходження скалярного та векторного добутків двох  $n$ -вимірних векторів, їх модулів. В окремих підпрограмах організувати ввід та вивід векторів.
11. Модуль для перевірки двох  $n$ -вимірних векторів на перпендикулярність та колінеарність. В окремих підпрограмах організувати ввід та вивід векторів.
12. Модуль для вводу довільної матриці розміром  $N \times N$  та знаходження її визначника
13. Модуль для перетворення градусної міри кута в радіанну і навпаки.
14. Модуль для вводу, виводу, знаходження суми, добутку двох квадратних матриць розміром  $N \times N$ .
15. Модуль для вводу, виводу довільної матриці розміром  $N \times N$ , знаходження сліду (суми діагональних елементів), першої та другої норм матриці ( $\|A\| = \max_i \sum_{j=1}^n |a(i, j)|, \|A\| = \max_j \sum_{i=1}^n |a(i, j)|$ )
16. Модуль для вводу, виводу довільної матриці розміром  $N \times N$ , побудови транспонованої матриці до заданої, перевірки матриці на симетричність, чи є матриця діагональною, трьохдіагональною.
17. Модуль для вводу, виводу коефіцієнтів многочлена степеня  $n$   
 $f(x) = a_0 + a_1x + \dots + a_nx^n$ , знаходження коефіцієнтів многочленів  $f'(x), f''(x)$
18. Модуль для вводу, виводу коефіцієнтів многочлена степеня  $n$   
 $f(x) = a_0 + a_1x + \dots + a_nx^n$ , знаходження коефіцієнтів многочленів  $c * f(x), (x - c) * f(x)$ , де  $c = \text{const}$ .
19. Модуль для розв'язання систем лінійних алгебраїчних рівнянь.
20. Модуль, що реалізує перетворення добутку тригонометричних функцій в суми.
21. Модуль для вводу, виводу коефіцієнтів многочлена степеня  $n$   
 $f(x) = a_0 + a_1x + \dots + a_nx^n$ , знаходження коефіцієнтів многочлена первісної многочлена  $f(x)$ .

22. Модуль для вводу, виводу коефіцієнтів многочлена степеня  $n$   $f(x) = a_0 + a_1x + \dots + a_nx^n$ , знаходження коефіцієнтів многочлена - суми двох многочленів  $f(x) + g(x)$  різних степенів.
23. Модуль для обчислення інтегралу функції  $y = f(x)$  на заданому відрізку  $[a, b]$  методом лівих, правих трапецій, парабол.
24. Модуль для перетворення числа із десяткової системи числення в двійкову.
25. Модуль для вводу, виводу матриці і вектора, добутку матриць, множення матриці на вектор.

### **Контрольні запитання:**

1. Що розуміють під модулем в TP?
2. Яка відмінність між стандартним модулем та модулем користувача?
3. Як здійснюється підключення модуля?
4. В якому випадку доцільне використання модулів користувача?
5. Які процедури та функції містить модуль System?
6. Як потрібно називати файл, що містить модуль?
7. Опишіть структуру модуля.
8. Яким ключовим словом розпочинається розділ реалізації?
9. Опишіть методику конструювання модуля.
10. Які переваги та недоліки (на вашу думку) модульного підходу до програмування?

### **Лабораторна робота №16**

#### **Тема: Використання вказівників та динамічних структур даних**

**Мета роботи:** Навчитися працювати з об'єктами динамічної природи: динамічними змінними, масивами; ознайомитися з основними процедурами для роботи з динамічними змінними та пам'яттю.

#### **Хід роботи:**

1. Розробити алгоритм розв'язку поставленої задачі з використанням вказівників та динамічних структур даних.
2. Написати відповідну програму на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.

## Тема 1. Робота з динамічними масивами

### Завдання 16.1.1

Виконати завдання по обробці одновимірних масивів, організувавши їх розміщення в пам'яті ПК динамічно. Значення елементів масиву ввести з клавіатури.

Варіанти завдань взяти з лабораторної роботи №7, завд. 7.2.



Національний університет  
водного господарства  
та природокористування

### Приклад 16.1.1

Знайти і надрукувати суму додатних елементів масиву.

#### Розв'язання:

Uses crt;

Type m=array [1..2] of real;

mas=<sup>^</sup>m;

*{Динамічний масив}*

Var b:mas;

n,i:integer;

s:real;

{**b**-}

Begin

Clrscr;

New(b);

*{Виділення пам'яті під масив}*

Writeln('Кількість елементів ');

Readln(n);

Writeln('Ввід масиву ');

For i:=1 to n do

Begin

Readln(b<sup>^</sup>[i]);

If b<sup>^</sup>[i]>0 then s:=s+b<sup>^</sup>[i];

*{Знаходження суми додатних елементів}*

End;

Writeln('Сума додатніх елементів ', s:8:1);

dispose(b);

*{Звільнення пам'яті, виділеної під масив}*

{**b**+}

Readln;

End.

Результат: S=13.6.

## Завдання 16.1.2

Виконати завдання по обробці двовимірних масивів, організувавши їх розміщення в пам'яті ПК динамічно. Значення верхніх індексів та елементів масиву ввести з клавіатури.

Варіанти завдань взяти з лабораторної роботи №8, завд. 8.1.

### Приклад 16.1.2

Задана деяка матриця  $A(n, m)$ . Обчислити елементи нової матриці, які визначаються за формулою

$$b_{ij} = \frac{a_{ij}}{S},$$

де S- сума елементів, які стоять на її головній діагоналі.

#### Розв'язання:

Uses crt;

Type mm=array [1..2,1..2] of real;

mas=^mm;

*{Опис динамічного масиву}*

Var a,b:mas;

n,i,j,m:integer;

s:real;

{Sb-}

Begin

Clrscr;

New(a);

*{Виділення пам'яті під масив}*

Writeln(' Кількість елементів матриці');

Readln(n,m);

*{Визначення розмірності матриці}*

Getmem(a,n\*m\*sizeof(real));

Writeln(' Ввід масиву ');

S:=0;

For i:=1 to n do

For j:=1 to m do

Begin

Readln(a^[i,j]);

If i=j then s:=s+a^[i,j];

*{Знаходження суми елементів головної діагоналі матриці}*

End;

Clrscr;

New(b);

For i:=1 to n do

Begin

Writeln;

```

    For j:=1 to m do
      Begin
        Write(a^[i,j]:6:1);
        b^[i,j]:=a^[i,j]/s;
      End;
    End;
  Writeln;
  Writeln(' Нова матриця ');
  For i:=1 to n do begin
    Writeln;
    For j:=1 to m do
      Write(b^[i,j]:6:2);
    End;
    Freemem(b,n*m*sizeof(real));
  End;
  Dispose(a);
  {$b+}
  Readln;
  End.

```

{Формування нової матриці}

{Виведення на екран нової матриці}

{Вивільнення області пам'яті, яку займав масив B(n,m)}

### Завдання 16.1.3

#### Приклад 16.1.3

Нехай є описання: *const n=100;*

*type vkazivka=^real;*

*vector=array[1..n] of vkazivka;*

Вважати, що всі елементи вектора відмінні від *nil*. Скласти програму для знаходження першого елемента вектора, який вказує на від'ємне число.

#### Розв'язання:

Uses Crt;

Const n=100;

Label m1;

Type

vector=array[1..n]of real;

{Опис динамічного масиву}

Var A:^vector;

i,k:integer;

BEGIN

Clrscr;



```

GetMem(A,sizeof(integer));           {Виділення пам'яті під масив}
For i:=1 to n do read(A^[i]);
Clrscr;
Readln;
For i:=1 to n do
  If A^[i]<0 then                       {Визначення від'ємного елемента}
  Begin
    Writeln('Nomer pershogo videmnego elementa');
    Write(i:2);
    Goto m1;
  End;
  m1:
    Readln;
END.

```



Наці  
водно  
та пр

### **Варіанти завдань для самостійної роботи:**

1. Із клавіатури вводиться послідовність слів, розділених комами, в кінці якої ставиться крапка. З використанням динамічних змінних знайти кількість входжень заданої букви в слово найбільшої довжини.
2. Скласти програму для заповнення, виводу на екран, знаходження найменшого елемента масиву дійсних чисел довільного, як завгодно великого розміру. Використати покажчик на масив.
3. Скласти програму для заповнення, виводу на екран, знаходження суми двох матриць довільного, як завгодно великого розміру. Використати покажчик на двомірний масив.
4. Функція одержує 1-й елемент масиву із 100 покажчиків, кожен з яких посилається на рядок. Впорядкувати рядки в лексикографічному порядку, переставляючи не рядки, а покажчики на них.
5. У масив символів довжиною 1000 записано слова, відокремлені одне від одного комами. Скласти програму, яка друкує всі слова з масиву.
6. У масив символів довжиною 1000 записано слова, відокремлені одне від одного комами. Скласти програму, яка перевіряє, чи є в масиві задане слово. Якщо такого слова немає, то дописати його в кінець масиву.
7. У масив символів довжиною 1000 записано слова, відокремлені одне від одного комами. Скласти програму, яка перевіряє, чи є в масиві задане слово. Якщо таке слово є в масиві, то замінити його на обернене.
8. Нехай є описання:
 

```

const n=100;
type vkazivka=^real;
vector=array[1..n] of vkazivka;

```

Вважати, що всі елементи вектора відмінні від *nil*. Скласти програму для знаходження елемента вектора, який вказує на найбільше дійсне число.

9. Нехай є описання (див. варіант 8). Вважати, що всі елементи вектора відмінні від *nil*. Скласти програму для знаходження елементів вектора, що вказують на від'ємні числа.
10. Нехай є описання (див. варіант 8). Вважати, що всі елементи вектора відмінні від *nil*. Скласти програму, яка шукає кількість елементів, що вказують на додатні числа.
11. Нехай є описання (див. варіант 8). Вважати, що всі елементи вектора відмінні від *nil*. Скласти програму, яка в векторі всі елементи, що вказують на рівні числа, замінює на перший з цих елементів.
12. Нехай є описання (див. варіант 8). Вважати, що всі елементи вектора відмінні від *nil*. Скласти програму, яка в векторі зберігає вказівки тільки на перший з однакових елементів, а інші замінює на *nil* і розміщує в кінці масиву.
13. З використанням динамічних змінних транспонувати матрицю довільного, як завгодно великого розміру.
14. Скласти програму для обчислення добутку двох квадратних матриць розміром  $10 \times 10$ . Використати покажчики на двомірні масиви.
15. Скласти програму для обчислення скалярного добутку двох векторів довільної довжини, представлених цілими масивами. Використати покажчики на масиви.
16. Скласти програму для впорядкування за зростанням цілого масиву довільного розміру шляхом перестановки елементів. Використати покажчик на масив.

17. Нехай є описання:
 

```

const d=...; {довжина рядка}
n=...; {максимальне число рядків}
type ryadok=string[d];
vkazivka=^ryadok;
text=array[1..n] of vkazivka;

```

Порожні рядки можуть відповідати лише останнім елементам масиву, при чому такі елементи повинні бути рівні *nil*. Скласти програму для підрахунку реального числа рядків в тексті.

18. Нехай є описання (див. варіант 17). Порожні рядки можуть відповідати лише останнім елементам масиву, при чому такі елементи повинні бути рівні *nil*. Скласти програму для вставки нового рядка після *i*-го.
19. Нехай є описання (див. варіант 17). Порожні рядки можуть відповідати лише останнім елементам масиву, при чому такі елементи повинні бути рівні *nil*. Скласти програму для перестановки *i*-го та *j*-го рядків.
20. Нехай є описання (див. варіант 17). Порожні рядки можуть відповідати лише останнім елементам масиву, при чому такі елементи повинні бути рівні *nil*. Скласти програму для видалення *i*-го рядка.
21. Нехай є описання (див. варіант 17). Порожні рядки можуть відповідати лише останнім елементам масиву, при чому такі елементи повинні бути

рівні *nil*. Скласти програму для дописування після і-го рядка копії j-го рядка.

22. Нехай є описання (див. варіант 17). Порожні рядки можуть відповідати лише останнім елементам масиву, при чому такі елементи повинні бути рівні *nil*. Скласти програму для видалення всіх рядків, які співпадають з і-им, крім нього самого.

23. Нехай є описання (див. варіант 17). Порожні рядки можуть відповідати лише останнім елементам масиву, при чому такі елементи повинні бути рівні *nil*. Скласти програму для знаходження найдовшого рядка.

24. Нехай є описання (див. варіант 17). Порожні рядки можуть відповідати лише останнім елементам масиву, при чому такі елементи повинні бути рівні *nil*. Скласти програму для знаходження рядка, в якому найбільше голосних букв.

25. Нехай є описання (див. варіант 17). Порожні рядки можуть відповідати лише останнім елементам масиву, при чому такі елементи повинні бути рівні *nil*. Скласти програму для заміни довшого з двох рядків і та j на коротший з них.

## Тема 2. Робота зі списками, стеками, чергами

### Завдання 16.2.1

Скласти програму з використанням динамічної пам'яті для створення та коригування списку, елементами якого є записи. Варіанти завдань взяти з лабораторної роботи №11, завд. 11.1.

#### Приклад 16.2.1

Дано інформацію про результати екзаменаційної сесії групи студентів 1-ого курсу спеціальності "Прикладна математика" (номер, П.І.П., оцінка з математичного аналізу, оцінка з програмування, оцінка з ОС та оцінка з алгебри). Вивести на екран номери, прізвища та кількість оцінок "2", "3", "4", "5" кожного студента та підрахувати загальну кількість. Введення та виведення інформації оформити у вигляді таблиці.

#### Розв'язання:

Uses crt;

Label m2;

Type poin=<sup>^</sup>ff;

ff=record

i:string;

m,p,o,al:integer;

zv:poin;

{Опис динамічного списку}

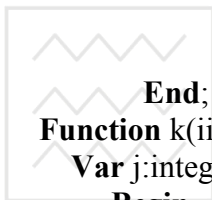
```

End;
Var spis,a,spis1,b:point;
    ch,o,h,z,p,l,v,g,q,tr,i,ii:integer;
    w:string;
    pp,ppp:pointer;
Procedure stvor; {Процедура створення списку}
Begin
Clrscr;
    Writeln(' кількість студентів ');
    Readln(l);
    New(spis); {Виділення динамічної пам'яті під список}
    a:=nil;
For ii:=1 to l do
    Begin
    Clrscr;
    New(spis);
    Writeln(ii:3,' студент ');
    Write(' прізвище');Readln(spis^.i);
    Write(' оцінка з мат. аналізу:');Readln(spis^.m);
    Write(' оцінка з програмування:');Readln(spis^.p);
    Write(' оцінка з ОС:');Readln(spis^.o);
    Write(' оцінка з алгебри:');Readln(spis^.al);
    spis^.zv:=a;
    a:=spis;
    End;
End;
Function k(ii,o:integer):integer; {Функція визначення кількості оцінок}
Var j:integer;
Begin
    j:=0;
If spis^.m=o then j:=j+1;
If spis^.p=o then j:=j+1;
If spis^.o=o then j:=j+1;
If spis^.al=o then j:=j+1;
    k:=j;
End;
Begin
Clrscr;
Mark(pp); {Визначення стану динамічної пам'яті}
    stvor;
    Writeln (' Студент математичний аналіз прогамув. ОС алгебра');
    a:=spis;
For q:=1 to l do

```



Національний  
водного господарства  
та природокористування



Національний університет  
водного господарства  
та природокористування

**Begin**

```
Writeln(spis^.i:12,spis^.m:8,spis^.p:8,spis^.o:8,spis^.a:8);  
spis:=spis^.zv;
```

**End;**

**Writeln;Writeln;**

```
Write (' Студент к-ть 2 к-ть 3 к-ть 4 к-ть 5');
```

```
spis:=a;
```

**For q:=1 to l do begin**

```
Writeln;
```

```
Write(spis^.i:12;o:=2;h:=k(q,o); {Виведення інформації за умовою}
```

```
Write (h:8);
```

```
z:=z+h;o:=3;h:=k(q,o);
```

```
Write (h:8);
```

```
tr:=tr+h;o:=4;h:=k(q,o);
```

```
Write (h:8);
```

```
ch:=ch+h;o:=5;h:=k(q,o);
```

```
Write (h:8);
```

```
p:=p+h;
```

```
spis:=spis^.zv;
```

```
m2:end;
```

**Writeln;**

```
w:='Всього';
```

```
Write(w:12,z:8,tr:8,ch:8,p:8);
```

```
Release(pp);Release(ppp); {Звільнення пам'яті з-під змінної ppp}
```

```
Readln;
```

**End.**

### Завдання 16.2.2

Скласти програму з використанням динамічних структур.

#### Приклад 16.2.2

Розробити програму обчислення значення многочлена степеня  $n$   
 $P_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$  в точці  $x$ . Коефіцієнти многочлена  $a_i$ ,  $i = \overline{1, n}$ , степінь  $n$  і число  $x$  вводяться з клавіатури користувачем. Коефіцієнти многочлена розмістити в пам'яті ПК динамічно у формі стека.

**Розв'язання:**

**Uses** crt;

```
Type point=^ads;
```

{Опис динамічного списку}

```
ads=record
```

```

koe:real;
poin:point;
End;
Var a,b:point;
i,n:integer;
x,s,aa:real;
p:pointer;
{$b-}

```

```

Procedure stvor;
Begin

```

*{Процедура створення списку}*

```

For i:=0 to n do

```

```

Begin

```

```

Write('A',i:2,' ');

```

```

Readln(aa);

```

```

New(a);

```

*{Виділення пам'яті під многочлен}*

```

a^.koe:=aa;

```

```

a^.poin:=b;

```

```

b:=a;

```

```

End;

```

```

End;

```

```

Procedure suma;

```

*{Процедура обчислення значення виразу}*

```

Begin

```

```

While a<>nil do

```

```

Begin

```

```

s:=s+a^.koe*exp(n*ln(x));

```

```

a:=a^.poin;

```

```

n:=n-1;

```

```

End;

```

```

End;

```

```

Begin

```

```

Clrscr;

```

```

Textcolor(13);

```

```

Mark(p);

```

*{Визначення поточного стану пам'яті}*

```

New(a);

```

```

Write('З яким степенем будемо оперувати ');

```

```

Readln(n);

```

```

Write('Введіть X ');

```

```

Readln(x);

```

```

Writeln('А тепер введемо коефіцієнти Ai:');

```

```

b:=nil;

```

*{Визначення кінця динамічної стрічки}*

```

stvor;

```

```

a:=b;

```

```

s:=0;

```

```

suma;
Writeln('s= ',s:8:2);
Release(p);
Readln;
End.

```

{Звільнення пам'яті з-під змінної p}

**Результати:** N=3; x=1.3; a<sub>0</sub>=1; a<sub>1</sub>=4; a<sub>2</sub>=5; a<sub>3</sub>=3;  
S=21,24.



Національний університет  
водного господарства  
та природокористування

### ***Варіанти завдань для самостійної роботи:***

1. Розробити програму формування стека, що містить цілі числа, і впорядкувати за зростанням елементів в цьому стеку. В процесі впорядкування елементи стеку переміщуватись не повинні.
2. Розробити програму формування стека, що містить цілі додатні числа, і його перетворення шляхом знищення з нього всіх парних чисел (в процесі перетворення стека його елементи в ОП переміщуватись не повинні).
3. Розробити програму формування стека, куди переміщується послідовність символів, які вводяться з клавіатури. Процес введення символів повинен завершитись, як тільки серед введених символів з'явиться крапка. Після цього програма повинна реверсувати стек. Під реверсуванням тут розуміється зміна напрямку зсилок в стеці на протилежній, тобто після реверсування вершина і дно стеку міняються місцями.
4. Розробити програму обчислення значення виразу:  $x_1x_n + x_2x_{n-1} + \dots + x_nx_1$ . При цьому значення  $x_1, x_2, \dots, x_n$  вводяться з клавіатури і динамічно розміщуються в ОП в формі стека.
5. Умова, що й в п 4. Значення  $x_1, x_2, \dots, x_n$  вводяться з клавіатури і динамічно розміщуються в ОП в формі двох стеків, один з яких реверсований по відношенню до іншого.
6. Розробити програму злиття двох стеків, що містять зростаючу послідовність цілих додатних чисел, в 3-й стек так, щоб його елементи розміщувались тільки в порядку зростання.
7. Розробити програму додавання до стека, що містить зростаючу послідовність цілих додатних чисел, нового елемента так, щоб порядок зростання в стеці не змінювався. При додаванні нового елемента в стек інші його елементи переміщуватись не повинні.
8. Розробити програму, що виявляє симетричність довільного тексту будь-якої довжини. Текст завжди повинен закінчуватись крапкою. Розв'язати дану задачу за допомогою двох стеків. В перший стек потрібно помістити весь текст, потім у другий стек перенести його половину так, щоб останній символ тексту знаходився на дні стеку.

Далі шляхом поелементного порівняння цих елементів з'ясувати симетричність тексту.

9. Розробити програму формування стека, куди розміщують цілі додатні числа, що вводяться з клавіатури. Процес введення повинен завершуватись, як тільки серед введених чисел з'являється від'ємне число. Після цього програма повинна вивести на екран вміст стека, при цьому порядок виведення чисел повинен бути зворотнім порівняно з послідовністю їх введення.
10. З клавіатури вводиться послідовність символів, що закінчується крапкою. Перевірити, чи в цій послідовності є баланс дужок. Використати динамічну структуру даних – стек. Наприклад:  $\{(a)[b]\}$  – баланс є;  $\{(a)[(b)]\}$  – балансу немає.
11. Дана непорожня послідовність слів, між сусідніми словами – кома, в кінці – крапка. Надрукувати слово найбільшої довжини.
12. Дана непорожня послідовність слів, між сусідніми словами – кома, в кінці – крапка. Надрукувати всі слова по алфавіту, вказуючи для кожного з них кількість входжень в цю послідовність.
13. Дана непорожня послідовність слів, між сусідніми словами – кома, в кінці – крапка. Надрукувати всі слова по зростанню їх довжин.
14. Дана непорожня послідовність слів, між сусідніми словами – кома, в кінці – крапка. Надрукувати всі слова, що містять цифри.
15. Дана непорожня послідовність слів, між сусідніми словами – кома, в кінці – крапка. Надрукувати всі слова в зворотному порядку.
16. Дано натуральне  $N$ . Надрукувати всі цифри десяткового запису числа  $N!$ .
17. Обчислити значення виразу в польському інверсному записі. Наприклад, вираз  $(5+7) \times 3 - 4 \times 2$  в польському інверсному записі має вигляд:  $5,7,+ ,3,\times,4,2,\times,-$ . Використати динамічну структуру даних – стек. Прим. Якщо при перегляді виразу з'являються дані, то вони заносяться в стек, а якщо з'явиться операція, то вона виконується над верхніми елементами стека із заміною їх результатом обчислення. Напр.,  $st = \langle \rangle \rightarrow \langle 5 \rangle \rightarrow \langle 5,7 \rangle \rightarrow \langle 12 \rangle \rightarrow \langle 12,3 \rangle \rightarrow \langle 36,4 \rangle \rightarrow \langle 36,4,2 \rangle \rightarrow \langle 36,8 \rangle \rightarrow \langle 28 \rangle$ .
18. Дано не порожній текстовий рядок. Утворити новий рядок, в який переписати всі символи із даного рядка, так щоб всі цифри переносились в його кінець. Використати динамічну структуру даних – черга.
19. Дано текст, який закінчується крапкою. Поява в тексті символу # означає відміну попереднього символу. Надрукувати текст, виправлений з врахуванням знака #. Напр.  $AB\#T\#\#\#HELLO\#LO \rightarrow HELLO$ .



20. Нехай задана логічна матриця, що описує лабіринт (true – стіна, false - прохід) і початкове місцезнаходження людини в лабіринті  $(x, y)$ . Запропонувати варіант обходу всіх допустимих клітин лабіринту.
21. Задано список з послідовності натуральних чисел. Додати в список після максимального елемента мінімальний.

### Завдання 16.2.3

Описати процедури створення та виводу на екран простого однонапрявленого списку, а також виконати завдання згідно варіанта.

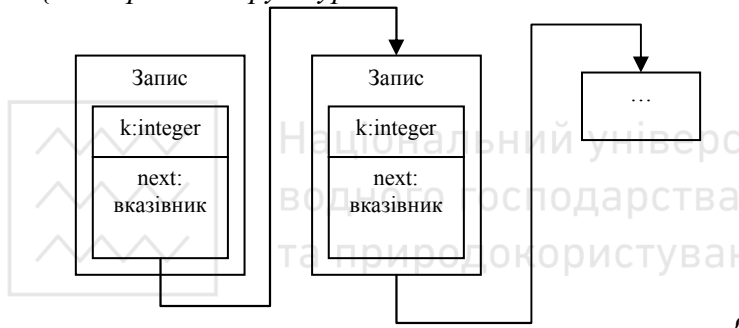
#### Приклад 16.2.3

Представити коефіцієнти многочлена  $a_n x^n + \dots + a_1 x + a_0$  у вигляді простого однонапрявленого списку і написати процедуру для обчислення значення многочлена в точці  $x$ .

**Розв'язання:**

**Uses** Crt;

*{Створення структури*



**Type** koef=<sup>^</sup>zapus;

**zapus=record**

**k:integer;**

**next:koef;**

**end;**

**Var** first,tek,tm:koef;

**x,Z:real; b,c:integer;**

**Procedure** Vvid(var n,a:integer);

*{Процедура зчитування коефіцієнтів  
многочлена}*

**Label** L,L1;

**Var** kf,i:integer;

**Begin**

**Writeln**('Введіть коефіцієнти:');

**Writeln**('Для закінчення введіть 0');

```

New(tm);                                {Створення першого елемента списку}
first:=tm;
L:
write('kf='); readln(kf);
if kf=0 then begin {Якщо кінець – перейти до вводу вільного члена}
writeln('Введіть вільний член:');
write('a='); readln(a);
writeln('Ввід завершено.');
```



Національний  
фонд  
захисту  
та  
природи  
України

```

tek^.next:=nil;                          {Завершення списку порожнім посиланням}
exit;
end;
tek:=tm;                                  {Переміщення поточного елемента в останній}
tek^.k:=kf;                               {Запис зчитаного числа у поле k}
new(tm);                                  {Створення нового елемента}
tek^.next:=tm; n:=n+1;                   {Переміщення вказівника}
goto L;
End;
{Обрахунок значення многочлена по списку коефіцієнтів}
```

**Procedure Obrobka(m:real; n,a:integer; var S:real);**

**var i,j:integer;**

**Begin**

tek:=first; i:=1; j:=n; {Поміщення поточного вказівника на початок списку}

S:=S+tek^.k\*exp(j\*ln(m)); {Додавання першого члена многочлена, помноженого на коефіцієнт, витягнутий зі списку}

while tek <> nil do begin j:=n-i; {Рух по списку до порожнього посилання}

S:=S+tek^.k\*exp(j\*ln(m)); {Додавання поточного члена многочлена, помноженого на коефіцієнт, витягнутий зі списку}

tek:=tek^.next; {Переміщення поточного вказівника за посиланням поточного елемента}

inc(i); end;

S:=S-1+a;

**End;**

**BEGIN**

clrscr;

writeln('Введіть значення x:');

readln(x);

Vvid(b,c);

Obrobka(x,b,c,Z); {Обрахунок значення многочлена в точці x}

clrscr; {Очищення екрана}

writeln('Z=',Z:5:1); {Виведення значення, отриманого в процедурі Obrobka}

**readkey;**  
**END.**

***Варіанти завдань для самостійної роботи:***

1. Описати рекурсивну процедуру (функцію), що визначає, чи є даний елемент в простому однонаправленому списку L.
2. Описати рекурсивну процедуру (функцію), що визначає, скільки разів входить даний елемент в однонаправлений список L.
3. Описати рекурсивну процедуру (функцію), яка друкує в зворотному порядку елементи однонаправленого списку L.
4. Описати рекурсивну процедуру (функцію), що знаходить максимальний елемент однонаправленого списку L.
5. Скласти процедуру, яка міняє місцями перший і останній елементи простого однонаправленого списку L.
6. Скласти процедуру, яка вставляє в простий однонаправлений список L після кожного елемента A елемент B.
7. Скласти процедуру, яка видаляє всі від'ємні елементи простого однонаправленого списку L.
8. Представити коефіцієнти многочлена  $a_n x^n + \dots + a_1 x + a_0$  у вигляді простого однонаправленого списку і написати процедуру для перевірки рівності многочленів.
9. Представити коефіцієнти многочлена  $a_n x^n + \dots + a_1 x + a_0$  у вигляді простого однонаправленого списку і написати процедуру для розкладання многочлена на нескоротні множники.
10. Представити коефіцієнти многочлена  $a_n x^n + \dots + a_1 x + a_0$  у вигляді простого однонаправленого списку і написати процедуру для обчислення значення похідної многочлена в точці x.
11. Представити коефіцієнти многочлена  $a_n x^n + \dots + a_1 x + a_0$  у вигляді простого однонаправленого списку і написати процедуру для додавання многочленів.
12. Представити коефіцієнти многочлена  $a_n x^n + \dots + a_1 x + a_0$  у вигляді простого однонаправленого списку і написати процедуру для табулювання значення многочлена на проміжку [a, b] з кроком h.
13. Описати процедуру для об'єднання двох списків в один.
14. Помістити тільки від'ємні елементи списку L<sub>1</sub> в список L<sub>2</sub>.
15. Описати функцію, яка визначає, чи входить список L<sub>1</sub> в список L<sub>2</sub>.
16. Описати процедуру, яка вставляє в список L<sub>1</sub> всі елементи списку L<sub>2</sub> після першого входження деякого елемента A, якщо такий є.
17. На основі списків L<sub>1</sub> і L<sub>2</sub> сформувати список L, включаючи в нього по одному елементу із списків L<sub>1</sub> і L<sub>2</sub>.

18. На основі списків  $L_1$  і  $L_2$  сформувати список  $L$ , включаючи в нього тільки ті елементи, які є в обох списках.
19. На основі списків  $L_1$  і  $L_2$  сформувати список  $L$ , включаючи в нього тільки ті елементи списку  $L_1$ , яких немає в списку  $L_2$ .
20. Впорядкувати по зростанню всі елементи простого однонапрявленого списку.
21. Знайти середнє арифметичне елементів непустиого списку.
22. Скласти підпрограму, яка перевіряє, чи є в списку хоча б два однакових елементів.
23. Нехай дано список  $A$ , що містить цілі числа. Помістіть в список  $B$  порядкові номери максимальних елементів списку  $A$ .
24. Скласти процедуру видалення всіх входжень елемента  $E$  в список  $L$ .
25. Описати підпрограму, яка в списку  $L$  із кожної групи розміщених поруч однакових елементів залишає тільки один.

#### Завдання 16.2.4

Скласти програму для створення та роботи з двонапрявленим (двозв'язним) списком згідно варіанту.

#### Приклад 16.2.4

Надрукувати елементи списку в зворотному порядку.

#### Розв'язання:

**Uses Crt;**

*{Оголошення списку, що містить числове поле та два поля вказівників на попередній та наступний елементи такого ж типу}*

**Type Spusok=<sup>^</sup>zapus;**

**zapus=Record**

**elem:Longint;**

**Next,Prev:Spusok;**

**End;**

**Var t,P:Spusok;**

**i,n:integer; x:longint;**

**Procedure Dod(X:Longint; Var t:Spusok);**

*{Процедура додавання елементів в список t}*

**Var P:Spusok;**

**Begin**

**New(P);**

*{Виділення пам'яті під елемент списку}*

**P^.elem:=X;**

*{Запис числа в інформаційне поле списку P}*

**If (t^.Next=Nil) then**

*{Умова досягнення кінця списку}*

**Begin**

```

    t^.Next:=P;           {Новостворений елемент записується
                          в кінець списку T}
    P^.Prev:=t;         {Створюємо зворотнє посилання з останнього
                          елемента на передостанній}
    P^.Next:=Nil;
  End;
End;
BEGIN
  clrscr;               {Очистка екрану}
  writeln('Введіть кількість елементів списку:');
  read(n);              {Зчитування числа елементів для роботи в циклі}
  writeln('Введіть елементи списку:');
  write('1- й=');
  read(x);              {Зчитування інформаційного поля для нового елемента списку}
  New(t);               {Створення списку t}
  t^.Next:=Nil;
  t^.Prev:=Nil;
  t^.elem:=x;           {Запис інформаційного поля списку}
  for i:=1 to (n-1) do begin {Цикл зчитування інформаційних полів
                          списку}

    write(i+1,'- й =');
    read(x);
    Dod(x,t);           {Запис інформаційного поля на початок списку P}
    t:=t^.next;        {Перехід до наступного елемента списку}
  end;
  clrscr;
  while t<>nil do begin
    writeln(t^.elem);  {Виведення значення інформаційного поля
                          поточного елемента списку}
    P:=t;              {Запис поточного елемента в P для подальшого
                          вивільнення пам'яті}
    t:=t^.prev;        {Перехід до попереднього елемента списку}
    Dispose(P);        {Вивільнення пам'яті, яку займала змінна P}
  End;
  readkey;
END.

```

### **Варіанти завдань для самостійної роботи:**

1. Написати програму “Телефонний довідник”, організувавши список абонентів у вигляді двонапрявленого списку. Описати процедуру, яка додає в список прізвище і номер телефону нового абонента.

2. Умова, що й у в.1. Описати процедуру, яка видаляє із списку дані про абонента із заданим прізвищем.
3. Умова, що й у в.1. Описати процедуру, яка здійснює пошук телефону по заданому прізвищу.
4. Умова, що й у в.1. Описати процедуру, яка дозволяє редагувати довільний номер телефону чи прізвище.
5. Створити кільцевий двонаправлений список по даних, взятих із однонапрявленого списку.
6. Скласти програму, яка розміщує елементи двонапрявленого списку в порядку зростання.
7. Дано кільцевий двонаправлений список. Скласти процедуру для перевірки, чи список є порожнім.
8. Дано кільцевий двонаправлений список. Скласти процедуру для перестановки у зворотному порядку всіх елементів, які розміщені між першим і останнім входженням деякого елемента А.
9. Дано кільцевий двонаправлений список. Скласти процедуру для вставки нового елемента в кінець списку.
10. Дано однов'язні списки  $L_1$  і  $L_2$ . Створити двозв'язний список  $L$ , в який будуть входити по одному разу ті елементи, які одночасно входять в обидва списки  $L_1$  і  $L_2$ .
11.  $N$  дітей розміщені по колу. Починаючи від 1-го видаляють кожного  $k$ -го елемента, замикаючи круг після видалення. Визначити порядок видалення дітей з кола, тобто вивести порядкові номери чи імена дітей в тому порядку, в якому вони видаляються.
12. Дано послідовність не менше, ніж з двох різних чисел, за якою слідує 0. Використовуючи двозв'язний список, надрукувати в зворотному порядку всі елементи, які знаходяться між найбільшим і найменшим.
13. Створити двозв'язний список, який подвоює кожне входження деякого елемента А.
14. Створити двозв'язний список слів довжиною до 12 символів. Підрахувати, який відсоток слів в ньому починається із заданої букви.
15. Дано двозв'язний список  $L$  із даними типу Студент (прізвище, група, середній бал). Впорядкувати елементи списку так, щоб прізвища йшли в алфавітному порядку.
16. Умова, що й у в.15. Вивести дані про тих студентів, які навчаються в заданій групі.
17. Умова, що й у в.15. Вивести дані про тих студентів, в яких середній бал  $> 4$ .
18. Створити двозв'язний список слів довжиною до 12 символів. Видалити із списку слова, які співпадають із заданим словом.
19. Створити двозв'язний список слів довжиною до 12 символів. Визначити слово максимальної довжини.

20. Дано впорядкований по зростанню числовий список L. Вставити в нього новий елемент, не порушуючи порядку.
21. Нехай дано текстовий файл. Створити двозв'язний список, кожен елемент якого містить кількість символів у відповідному рядку тексту.
22. З клавіатури вводиться послідовність  $x_1, x_2, \dots, x_n$ , дійсних чисел, де  $n$  – наперед невідома величина. Введення закінчується символом переведення рядка. Знайти суму добутків:  $S = x_1x_n + x_2x_{n-1} + \dots + x_nx_1$ .
23. Умова така ж, що й у в.22. Знайти добуток сум:  $S = (x_1 + x_2 + 2x_n)(x_2 + x_3 + 2x_{n-1}) \dots (x_{n-1} + x_{n-2} + 2x_2)$ .
24. Умова така ж, що й у в.22. Знайти добуток сум:  $S = (x_1^2 - x_n^2)(x_2^2 - x_{n-1}^2) \dots (x_n^2 - x_1^2)$ .
25. Нехай заданий кільцевий двозв'язний список, що містить цілі числа від 0 до 1000. Довільний елемент вибирається як робочий елемент. Для роботи зі списком визначені тільки чотири операції:
- 1) *КрокВліво* – зміна робочого елемента. Робочим стає лівий сусід даного робочого елемента;
  - 2) *КрокВправо* – робочим стає правий сусід даного робочого елемента;
  - 3) *Прочитати* – помістити в задану змінну значення робочого елемента списку;
  - 4) *Записати* – записати в робочий елемент списку значення заданої змінної (значення змінної належить проміжку 0...1000).
- Визначити кількість елементів кільцевого списку.

Національний університет

водн... ва  
та природокористування

**Контрольні запитання:**

1. Які змінні називають динамічними?
2. Які причини використання динамічних змінних?
3. Що називають вказівником?
4. Який об'єм пам'яті потрібний для зберігання вказівника?
5. Що описується в першу чергу – тип вказівника чи тип даних, на які він вказує?
6. Для чого використовується значення піл при роботі з вказівниками?
7. Що називають базовим типом?
8. Які операції здійснюються над динамічними змінними?
9. Які процедури та функції використовують для роботи із вказівниками?
10. Чи обмежена динамічна пам'ять?

## Лабораторна робота №17

### Тема: Використання об'єктно-орієнтованого програмування (ООП)

**Мета роботи:** Ознайомитися з технологією використання ООП.

#### Хід роботи:

1. Розробити алгоритм розв'язку поставленої задачі з використанням ООП.
2. Написати відповідну програму на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.

#### Завдання 17.1

Використовуючи завдання по обробці результатів екзаменаційної сесії з використанням записів і процедур (лабораторна робота №11, завдання 11.1), написати програму з використанням технології ООП.

#### Приклад 17.1

З використанням технології ООП надрукувати таблицю, що містить номери, прізвища та кількість "5", "4", "3", "2" у кожного студента групи, а також підрахувати загальну кількість "5", "4", "3", "2" в групі.

№	Прізвище	Мат. аналіз	Програмув.	ОС	Алгебра та геом.
1.	Іванчук С.О.	4	3	3	4
2.	Панченко І.А.	5	4	4	5
3.	Заєць О.М.	3	4	4	4
4.	Вельбицький П.О.	4	3	3	3
5.	Сидоренко В.Р.	2	3	3	2
6.	Кравченко З.І.	3	5	4	5
7.	Якубів Р.Н.	5	4	4	3
8.	Зоренко П.М.	4	2	3	3
9.	Берестяк Г.С.	4	5	5	5
10.	Дячик Н.С.	5	5	4	4



## Розв'язання:

Uses crt;

Type ff=record

i:string;

m,p,o,a:integer;

End;

spisok=object

*{Створення об'єкта spisok}*

Function k(i,o:integer):integer;

Procedure sum;

End;

Const t:array[1..10] of

ff=((i:'Іванчук';m:4;p:3;o:3;a:4),(i:'Памченко';m:3;p:4;o:4;a:4),

(i:'Заець';m:3;p:4;o:4;a:4),(i:'Вельбицько';m:4;p:3;o:3;a:3),(i:'Сидоренко';m:2;

p:2;o:3;a:2),

(i:'Кравченко';m:3;p:5;o:4;a:5),(i:'Якубів';m:5;p:4;o:4;a:3),(i:'Зоренко';m:4;p:2

;o:3;a:3),

(i:'Березняк';m:4;p:5;o:5;a:5),(i:'Дячик';m:5;p:5;o:4;a:4));

Var

ch,o,h,z,p,l,v,g,q,tr,kl:integer;

w:string;

spis:spisok;

Function spisok.k(i,o:integer):integer; *{Функція визначення кількості оцінок}*

Var

j:integer;

Begin

j:=0;

If t[i].m=o then j:=j+1;

If t[i].p=o then j:=j+1;

If t[i].o=o then j:=j+1;

If t[i].a=o then j:=j+1;

k:=j;

End;

Procedure spisok.sum;

*{Метод sum об'єкта spisok}*

Begin

Clrscr;

Writeln (' Студент мат.ан прог. ОС алгебра');

For q:=1 to 10 do

Writeln(t[q].i:12,t[q].m:8,t[q].p:8,t[q].o:8,t[q].a:8);

Writeln;

Write (' Студент к-ть 2 к-ть 3 к-ть 4 к-ть 5');


For q:=1 to 10 do

Begin



Національний  
університет  
водного  
господарства  
та природо  
користування

Національний університет  
водного господарства  
та природоко  
ристування



```

Write(t[q].i:12);
  For kl:=2 to 5 do
    Begin
      h:=spis.k(q,kl);
      Write (h:8);
      Case kl of
        2:z:=z+h;
        3:tr:=tr+h;
        4:ch:=ch+h;
        5:p:=p+h;
      End;
    End;
  End;
  Writeln;
  w:='Всього';
  Write(w:12,z:8,tr:8,ch:8,p:8);      {Загальна кількість "5", "4", "3", "2"}
  Readln;
End;
Begin
spis.sum;
End.

```

### Завдання 17.2

Використовуючи завдання по обробці типізованих файлів (лабораторна робота №12, завд. 12.2. та 12.4.), написати програму з використанням технології ООП.

#### Приклад 17.2

З використанням технології ООП сформувати файл "BOOK", який містить інформацію про книги вашої бібліотеки. Інформація повинна містити прізвище автора, назву книги, видавництво та рік видання. Використовуючи сформований файл, роздрукувати інформацію про книги, видані у видавництві "Просвіта".

#### Розв'язання:

```

Uses Crt;
Type rr=record
  naz,avt,vid:string;

```

```

    rik:integer;
End;
fil=object {Створення об'єкта fil}
    Procedure show;
    Procedure umov;
    Procedure stvor;
    Procedure menu;
End;

```



```

Var національний університет
    водного господарства
    та природокористування
    filik:fil; {Змінна filik типу fil}
    f2:file of rr;
    rrs:rr;
    n,p,no,k:integer;

```

```

Procedure fil.stvor; {Процедура створення анкетних даних про книгу}

```

```

Begin
    Rewrite(f2); {Відкриття і створення нового файлу з ім'ям f2}

```

```

    Repeat

```

```

        Clrscr;

```

```

        Write(' Назва книги: ');Readln(rrs.naz);

```

```

        Write(' Автор: '); Readln(rrs.avt);

```

```

        Write(' Видавництво ');Readln(rrs.vid);

```

```

        Write(' Рік видання: ');Readln(rrs.rik);

```

```

        Write(f2,rrs);

```

```

        Write(' Створювати ще каталог?(так-1) ');

```

```

        Readln(p);

```

```

    Until p>1 ;

```

```

    Close(f2); {Закриття файлу з ім'ям f2}

```

```

End;

```

```

Procedure fil.show; {Процедура перегляду даних файлу}

```

```

Begin

```

```

    Reset(f2); {Відкриття вже існуючого файлу f2}

```

```

    While not eof(f2) do

```

```

        Begin

```

```

            Read(f2,rrs); {Зчитування даних з файлу f2}

```

```

            With rrs do

```

```

                Begin

```

```

                    Write(naz:14); {Запис результатів у файл f2}

```

```

                    Write(avt:13);

```

```

                    Write(vid:13);

```

```

                    Write(rik:6);

```

```

                End;

```

```

    End;

```

```

Readln;

```

**End;**  
**Procedure** fil.umov; *{Процедура перевірки умови задачі}*

**Begin**  
Reset(f2);  
Writeln(' Книги, видані у видавництві «Просвіта» ');  
While not eof(f2) do

**Begin**  
Read(f2,rrs);  
**If** rrs.vid='Просвіта' **then with** rrs **do**

**Begin**  
Write(naz:14);Write(avt:13);  
Write(vid:13);Write(rik:6);  
**End;**

**End;**  
Readln;

**End;**  
**Procedure** fil.menu; *{Процедура створення меню програми}*

**Var**  
cc:char;  
**Begin**  
While cc<>'6' do

**Begin**  
Clrscr;  
Writeln('1----Створити файл');  
Writeln('2----Показати файл');  
Writeln('3----Вивід за умовою');  
Writeln('4----Вихід');  
cc:=readkey;

**Case** cc **of** *{Вибір пункту меню}*  
'1':filik.stvor;  
'2':filik.show;  
'3':filik.umov;  
'4':exit;

**End;**

**End;**

**End;**

**Begin**  
Clrscr;  
Assign(f2,'a:kn.txt'); *{Зв'язує файловою змінною f2 з ім'ям kn.txt зовнішнього файла на диску}*

filik.menu;  
Close(f2);

**End.**

## Завдання 17.3

### Приклад 17.3

Розробити програму **Char\_stack** і **Read\_stack** для розв'язування задачі обчислення виразу. Напишіть програму обчислення виразів з використанням об'єктів.



**Розв'язання:** итет  
водного господарства  
та природокористування

**Uses** Crt;

**Type** point=^ads;

ads=record

koe:real;

poin:pointer;

**End;**

char\_stack=object *{В склад об'єкта char\_stack входять 3 методи}*

**Procedure** vvid;

**Procedure** obch;

**Procedure** vivid;

**End;**

**Var** a,b:pointer;

i,n:integer;

x,s,aa:real;

p:pointer;

ob:char\_stack;

{Sb-}

**Procedure** char\_stack.vvid; *{Процедура введення даних}*

**Begin**

**Clrscr;**

**Mark**(p); *{Процедура, що зберігає адресу першого вільного байту області пам'яті}*

**New**(a); *{Виділення динамічної пам'яті під змінну a}*

**Write**('З яким степенем будемо оперувати ');

**Readln**(n);

**Write**('Введіть X ');

**Readln**(x);

**Writeln**('А тепер введемо коефіцієнти Ai:');

b:=nil; *{Змінна b вказує на кінець динамічної структури}*

**For** i:=0 to n do

**Begin**

**Write**('A',i:2,' ');

**Readln**(aa);

```

    New(a);
    a^.koe:=aa;
    a^.poin:=b;
    b:=a;
End;
End;
Procedure char_stack.obch;           {Процедура обчислення значення виразу}
Begin

```



```

a:=b;
s:=0;

```

```

While a<>nil do

```

```

    Begin

```

```

        s:=s+a^.koe*exp(n*ln(x));
        a:=a^.poin;
        n:=n-1;
    
```

```

    End;

```

```

End;

```

```

Procedure char_stack.vivid;         {Процедура виведення результату}

```

```

Begin

```

```

    Writeln('s= ',s:8:2);

```

```

    Release(p);

```

```

End;

```

```

Begin

```

```

    ob.vvid;
    ob.obch;
    ob.vivid;

```

```

End.

```

**Результати:** N=3; x=1.3; a<sub>0</sub>=1; a<sub>1</sub>=4; a<sub>2</sub>=5; a<sub>3</sub>=3; S=21,24.

### ***Варіанти завдань для самостійної роботи:***

- З використанням технології ООП розробити програму обчислення добутку 2-х матриць, описавши тип об'єкта **Matrix** (матриця довільного розміру) і його методи:
  - введення матриці;
  - виведення матриці;
  - знаходження добутку значень 2-х матриць.
- Розробити ООП - програму побудови квадрата, описавши при цьому тип об'єкта **Square** для його побудови та зафарбувавши його кольором, використавши методи:
  - задання сторони квадрату;
  - намалювати квадрат;

- задати нові координати лівого верхнього кута;
  - перемістити квадрат;
  - знищити квадрат з екрана монітору.
3. Розробити програму побудови круга, описавши при цьому тип об'єкта **Circle**, для його побудови та зафарбувавши його кольором, використавши методи:



Національний університет  
водного господарства  
та природокористування

- задання радіуса круга;
  - намалювати круг;
  - задати нові координати центра круга;
  - перемістити круг в інше місце;
  - знищити круг з екрана монітору.
4. З використанням технології ООП розробити програму для роботи з об'єктом **Point** – точкою використавши методи:
- задання координат точки;
  - засвічення точки;
  - погашення точки;
  - переміщення точки.

Переміщення здійснити за допомогою клавіші руху курсору.

5. Розробити програму побудови круга, описавши при цьому тип об'єкта **Circle** на основі батьківського типу **Point** (див. п.5), використавши методи:

- зсунення координат центра і радіуса кола;
- висвітлення кола на екрані;
- знищення кола.

6. Скласти програму в стилі ООП для виконання дій над векторами, описавши при цьому тип об'єкта **Vector** і використавши методи:

- задання координат вектора;
- додавання двох векторів;
- множення вектора на скаляр;
- вивід вектора на екран.

7. Модифікувати програму у в.6 для виконання дій над матрицями, описавши при цьому тип об'єкта **Matrix** на основі батьківського типу **Vector**, використавши методи:

- задання елементів матриці;
- додавання двох матриць;
- множення матриці на скаляр;
- виведення матриці на екран.

8. З використанням технології ООП розробити програму для виконання дій над комплексними числами, описавши при цьому тип об'єкта **Complex** і використавши методи:

- ініціалізація дійсної і уявної частини;
- обчислення суми двох комплексних чисел;
- обчислення добутку двох комплексних чисел;

- вивід на екран комплексного числа.

9. Скласти програму в стилі ООП для виконання дій над вектором, описавши при цьому тип об'єкта **Vector** і використавши методи:

- ініціалізація вектора;
- знаходження довжини вектора;
- множення вектора на вектор;
- виведення вектора на екран.

10. Розробити ООП-програму для роботи з об'єктом відрізок, описавши при цьому тип **Vidrizok** для його побудови і використавши методи:

- задання координат початку і кінця;
- виведення зображення відрізка на графічний екран;
- обчислення довжини відрізка;
- знищення відрізка.

11. З використанням технології ООП розробити програму для побудови ламаної, описавши при цьому тип об'єкта **Lamana** на основі батьківського типу **Vidrizok** (див. в.10), використавши при цьому методи:

- задання координат вершин ламаної;
- виведення зображення ламаної на графічний екран;
- обчислення довжини ламаної;
- знищення ламаної.

12. Розробити ООП-програму для роботи з об'єктом трикутник, описавши при цьому тип об'єкта **Trikutnik** на основі батьківського типу **Vidrizok** (див. в. 10), використовуючи методи:

- задання координат вершин трикутника;
- обчислення периметра трикутника та його площі;
- виведення зображення трикутника на графічний екран;
- знищення трикутника.

13. Розробити програму в стилі ООП для роботи з об'єктом **Trikutnik**, використавши при цьому методи:

- задання координат вершин трикутника;
- виведення зображення трикутника на екран;
- переміщення трикутника;
- знищення трикутника.

Переміщення здійснити за допомогою клавіші руху курсору.

14. З використанням технології ООП скласти програму для роботи з об'єктом **Kub**, використавши методи:

- задання довжини ребра куба;
- знаходження його об'єму та площі поверхні;
- виведення зображення куба на екран;
- знищення куба.

15. З використанням технології ООП скласти програму для побудови кулі з перерізом, використавши методи:



- задання радіуса кулі;
- виведення кулі на екран;
- обчислення об'єму та площі поверхні кулі;
- знищення кулі.

16. Розробити програму в стилі ООП для роботи з об'єктом **Piramida**, використавши при цьому методи:

- задання координат точок, що лежать в основі піраміди та координати вершини піраміди;
- виведення зображення піраміди на екран;
- знаходження площі основи піраміди та об'єму;
- знищення піраміди.



Національний університет  
водного господарства  
та природокористування

17. Розробити ООП-програму для роботи з об'єктом трапеція, описавши при цьому тип об'єкта **Trapezia**, використовуючи методи:

- задання координат вершин трапеції;
- виведення трапеції на екран;
- обчислення площі трапеції;
- знищення трапеції.

18. Скласти програму з використання технології ООП для роботи з об'єктом пряма, описавши при цьому тип об'єкта **Line** та використавши методи:

- задання координат двох точок, через які проходить пряма;
- виведення прямої на екран;
- побудова через задану точку паралельної прямої;
- побудова перпендикулярної прямої;
- знищення побудованих прямих.

19. З використанням технології ООП скласти програму для побудови конуса, описавши при цьому тип об'єкта **Konus**, використавши методи:

- задання радіуса основи та висоти конуса;
- виведення конуса на екран;
- обчислення площі бічної поверхні та об'єму конуса;
- знищення конуса.

20. Розробити ООП-програму для роботи з об'єктом циліндр, описавши при цьому тип об'єкта **Cylinder**, використавши при цьому методи:

- задання радіуса основи та висоти циліндра;
- виведення зображення циліндра на екран;
- обчислення площі повної поверхні та об'єму циліндра;
- знищення циліндра.

21. Скласти програму з використання технології ООП для роботи з об'єктом **Flower** на основі батьківського типу **Circle** (див. в. 3), використавши методи:

- виведення зображення квітки (пелюсток квітки) на екран;
- переміщення зображення;
- знищення зображення квітки.

Переміщення здійснити за допомогою клавіші руху курсору.

22. З використанням технології ООП скласти програму для побудови тетраедра, використавши методи:

- задання координат вершин тетраедра;
- виведення тетраедра на екран;
- обчислення площі повної поверхні та об'єму тетраедра;
- знищення тетраедра.

23. Розробити ООП-програму для роботи з об'єктом кута, описавши при цьому тип об'єкта **Kut** та використавши методи:

- задання значення кута в градусах;
- виведення зображення кута на екран;
- обчислення значень тригонометричних функцій даного кута;
- знищення зображення кута.

24. З використанням технології ООП написати програму для роботи із об'єктом **String**, який використовується для зберігання стрічкового типу, описавши при цьому методи:

- перевірка, чи є стрічка порожньою;
- зменшення стрічки до заданої довжини;
- розбиття стрічки на дві підстрічки.

25. Скласти програму з використанням технології ООП для роботи із об'єктом **Char**, який використовується для зберігання символного значення і містить методи визначення, чи є заданий символ буквою, цифрою чи розділовим знаком.

### **Контрольні запитання:**

1. Які основні ідеї об'єктно-орієнтованого програмування?
2. Що розуміють під об'єктом в ООП?
3. Як називаються змінні об'єктного типу?
4. Чим відрізняється об'єктний тип від типу "запис"?
5. В чому полягає інкапсуляція?
6. Дайте визначення наслідуванню.
7. Для чого використовуються в ООП віртуальні методи?
8. Яке призначення конструктора та деструктора?
9. Чи можуть змінні об'єктного типу бути динамічними?
10. Що розуміють під поліморфізмом в ООП?

## Повідомлення компілятора про помилки

Назва помилки	Пояснення
<b>1: Out of memory</b>	Дана помилка з'являється, якщо компілятор використав всю допустиму йому пам'ять. Для вирішення цієї проблеми потрібно змінити значення області <b>Destination</b> з <b>Memory</b> на <b>Disk</b> .
<b>2: Identifier expected</b>	В цьому місці повинен знаходитись ідентифікатор.
<b>3: Unknown identifier</b>	Даний ідентифікатор не був описаний.
<b>4: Duplicate identifier</b>	Повторення ідентифікатора.
<b>5: Syntax error</b>	У вихідному тексті знайдений недопустимий символ. Можливо, не поміщена в лапки стрічкова константа.
<b>6: Error in real constant</b>	Помилка в синтаксисі константи дійсного типу.
<b>7: Error in integer constant</b>	Помилка в синтаксисі константи цілого типу.
<b>8: String constant exceeds line</b>	Стрічкова константа перевищує допустимі розміри. Можливо, після стрічкової константи відсутні лапки.
<b>9: Too many nested file</b>	Компілятор дозволяє не більше 15 вкладених вихідних файлів і не більше 14 включених файлів.
<b>10: Unexpected end of file</b>	Не знайдено кінця файлу.
<b>11: Line too long</b>	Максимальна довжина стрічки не може перевищувати 127 символів.
<b>12: Type identifier expected</b>	В даній програмі не описаний даний ідентифікатор.
<b>13: Too many open files</b>	Забагато відкритих файлів.
<b>14: Invalid file name</b>	Ім'я файлу неправильне, або вказано неіснуючий шлях.
<b>15: File not found</b>	Файл не знайдений ні в одному з каталогів.
<b>16: Disk full</b>	Диск переповнений.

<b>17: Invalid compiler directive</b>	Невірна буква в директиві компілятора, один із параметрів директиви компілятора невірний або ви користуєтесь глобальною директивою коли компіляція тіла програми вже розпочалась.
<b>18: Too many files</b>	В компіляції програми або програмного модуля бере участь забагато файлів .
<b>19: Undefined type in pointer definition</b>	Невизначений тип в оголошенні вказівника.
<b>20: Variable identifier expected</b>	У вказаному курсором місці очікується ідентифікатор змінної.
<b>21: Error in type</b>	Помилка в оголошенні типу.
<b>22: Structure too large</b>	Максимальний розмір будь-якого типу – 65520 байтів.
<b>23: Set base type of range</b>	Базовий тип множини повинен являти собою інтервальний або порядковий тип даних з не більше чим 256 значеннями.
<b>24: File components may not be files or object</b>	Тип компонентів файлів не може бути об'єктним типом, файловим типом або будь-яким іншим структурованим типом, який містить компоненти файлу або об'єкта.
<b>25: Invalid string length</b>	Довжина описуючої стрічки повинна знаходитись в межах від 1 до 255.
<b>26: Type mismatch</b>	Невідповідність типу.
<b>27: Invalid subrange base type</b>	Неправильний базовий тип для інтервалу.
<b>28: Lower bound greater than upper bound</b>	При описанні типу даних Ви задали нижню границю діапазону більше верхньої.
<b>29: Ordinal type expected</b>	Очікується порядковий тип.
<b>30: Integer constant expected</b>	Очікується константа цілого типу.
<b>31: Constant expected</b>	Очікується константа.

<b>32: Integer or real constant expected</b>	Очікується константа цілого або дійсного типу.
<b>33: Type identifier expected</b>	Очікується ідентифікатор типу.
<b>34: Invalid function result type</b>	Правильними типами результату функції є всі прості, стрічкові типи.
<b>35: Label identifier expected</b>	Знайдено посилання на мітку, яка не описана в розділі <b>Label</b> .
<b>36: BEGIN expected</b>	Очікується оператор <b>Begin</b> .
<b>37: END expected</b>	Очікується оператор <b>End</b> .
<b>38: Integer expression expected</b>	Очікується вираз цілого типу.
<b>39: Ordinar expression expected</b>	Очікується вираз перераховуючого типу.
<b>40: Boolean expression expected</b>	Очікується вираз логічного типу.
<b>41: Operand types do not match operator</b>	Невідповідність типів операнду.
<b>42: Error in expression</b>	Даний ідентифікатор не може брати участі у вираженні вказаним чином.
<b>43: Illegar assignment</b>	Це повідомлення може з'явитись з наступних причин: - Файлам і нетипізованим змінним не можна присвоювати певні значення. - Ідентифікатору функції можна присвоювати значення тільки в середині розділу операторів даної функції.
<b>44: Field identifier expected</b>	Даний ідентифікатор не відповідає полю попередньої змінної типу <b>Record</b> або <b>Object</b> .
<b>45: Object file too large</b>	Turbo Pascal не може компонувати OBJ-файли розміром більше, ніж 64 Кбайт.

<b>46: Undefined external</b>	Зовнішня процедура або функція не має відповідного визначення <b>Public</b> в об'єктному файлі.
<b>47: Invalid object file record</b>	Файл .Obj містить невірний об'єктний запис.
<b>48: Code segment too large</b>	Максимально допустимий розмір коду програми або програмного модуля рівний 65520 байтів. Розбийте Вашу програму або програмний модуль на декілька інших.
<b>49: Date segment too large</b>	Максимальний розмір сегмента даних програми рівний 65520 байт, включаючи дані, описані використовуваними програмними модулями. Якщо потрібна кількість глобальних даних, опишіть великі структури з допомогою вказівників і виділіть для них пам'ять динамічно з допомогою процедури <b>New</b> .
<b>50: DO expected</b>	Очікується оператор <b>Do</b> .
<b>51: Invalid PUBLIC definition</b>	Недопустиме визначення <b>Public</b> .
<b>52: Invalid EXTRN definition</b>	Неправильне визначення <b>Extrn</b> .
<b>53: Too many EXTRN definition</b>	Turbo Pascal не може обробляти файли .Obj при більше ніж 256 оприділення <b>Extrn</b> .
<b>54: OF expected</b>	Очікується оператор <b>Of</b> .
<b>55: INTERFACE expected</b>	Очікується оператор <b>Interface</b> .
<b>56: Invalid relocatable reference</b>	Недопустима зміщена зсилка.
<b>57: THEN expected</b>	Очікується оператор <b>Then</b> .
<b>58: TO or DOWNT0 expected</b>	Очікується зарезероване слово <b>To</b> або <b>Downto</b> .
<b>59: Undefined forward</b>	Невизначене випереджаюче описання.

<b>61: Invalid typecast</b>	Недопустиме перетворення типів.
<b>62: Division by zero</b>	Операція намагається виконати ділення на нуль.
<b>63: Invalid file type</b>	Даний файловий тип не обслуговується процедурою обробки файлів. Наприклад, процедура <b>Readln</b> використовує для типізованого файлу або процедуру <b>Seek</b> —для текстового файлу.
<b>64: Cannot Read or Write variable of this type</b>	Ця помилка може виникнути із-за спроби ввести або вивести змінну невідповідного типу: - Процедури <b>Read</b> і <b>Readln</b> можуть зчитуватись змінними - Процедури <b>Write</b> і <b>Writeln</b> можуть виводити змінні символного, дійсного і стрічкового типів.
<b>65: Pointer variable expected</b>	Змінна повинна мати тип вказівника.
<b>66: String variable expected</b>	Змінна повинна мати стрічковий тип.
<b>67: String expression expected</b>	Вираз повинен мати стрічковий тип.
<b>68: Circular unit reference</b>	Циклічна залежність модулів.
<b>69: Unit name mismatch</b>	Ім'я програмного модуля, знайдене у файлі TPU, не відповідає імені, вказаному в операторі USES.
<b>70: Unit version mismatch</b>	Програмні модулі, які використовуються даною програмою, були змінені після їх компіляції.
<b>71: Duplicate unit name</b>	Ім'я даного програмного модуля вже вказано в операторі USES.
<b>72: Unit file format error</b>	TPU-файл являється недійсним.
<b>73: Implementation expected</b>	В модулі відсутній розділ реалізації.
<b>74: Constant and case types do not match</b>	Тип константи оператора <b>Case</b> не сумісний з вираженням в операторі варіанта.

<b>75: Record variable expected</b>	Очікується змінна типу <b>Record</b> .
<b>76: Constant out of range</b>	Ця помилка може з'явитись з наступних причин: - При спробі вказати масив з константами, які порушують границі. - При спробі присвоїти змінній значення константи, які виходять за діапазон змінної. - При спробі передати константу за діапазон в якості параметра процедури або функції.
<b>77: File variable expected</b>	Очікується змінна файлового типу.
<b>78: Pointer expression expected</b>	Очікується змінна типу вказівника.
<b>79: Integer or real expression expected</b>	Очікується вираз типу <b>Integer</b> або <b>Real</b> .
<b>80: Label not within current block</b>	Оператор <b>Goto</b> не може здійснити перехід на мітку, яка знаходиться за поточним блоком.
<b>81: Label already defined</b>	Дана мітка вже помічає мітку переходу.
<b>82: Undefined label in processing statement part</b>	Не визначена мітка в попередній частині оператора.
<b>83: Invalid @ argument</b>	Правильними аргументами являються імена змінних, процедур і функцій.
<b>84: Unit expected</b>	Очікується оператор <b>Unit</b> .
<b>85: “;” expected</b>	Очікується оператор “;”.
<b>86: “:” expected</b>	Очікується оператор“:”.
<b>87: “,” expected</b>	Очікується оператор“,”.
<b>88: “(” expected</b>	Очікується оператор“(”.
<b>89: “)” expected</b>	Очікується оператор“)”.





<b>90:</b> “=”expected	Очікується оператор“=”.
<b>91:</b> “:=”expected	Очікується оператор“:=”.
<b>92:</b> “[” or “(“expected	Очікується оператор“[”або“(.”.
<b>93:</b> “]” or “)”expected	Очікується оператор“]”або“.)”.
<b>94:</b> “.”expected	Очікується оператор“.”.
<b>95:</b> “..”expected	Очікується оператор“..”.
<b>96: Too many variables</b>	Забгато змінних.
<b>97: Invalid FOR control variable</b>	Параметр циклу <b>For</b> повинен бути змінною цілого типу.
<b>98: Integer variable expected</b>	Очікується змінна цілого типу.
<b>99: Files and procedure types are not allowed here</b>	Типізована константа не може мати файловий тип або тип процедури.
<b>100: String length mismatch</b>	Довжина строкової константи не відповідає кількості елементів символьного масиву.
<b>101: Invalid ordering of fields</b>	Поля в константі типу <b>Record</b> повинен записуватись в порядку їх описання.
<b>102: String constant expected</b>	Очікується константа стрічкового типу.
<b>103: Integer or real variable expected</b>	Очікується змінна цілого або дійсного типу.
<b>104: Ordinar variable expected</b>	Очікується змінна перелічуваного типу.
<b>105: INLINE error</b>	Помилка в операторі <b>Inline</b> .

<b>106: Character expression expected</b>	Очікується вираз символьного типу.
<b>107: Too many relocation items</b>	Забгато переміщуваних елементів.
<b>108: Overflow in arithmetic operation</b>	Переповнення при виконанні арифметичних операцій.
<b>109: No enclosing FOR, WHILE or REPEAT statement</b>	Відсутні оператори, що завершують оператори <b>For</b> , <b>While</b> або <b>Repeat</b> .
<b>110: Debug information table overflow</b>	Переповнення інформаційної таблиці відлагодження.
<b>111: N/A</b>	
<b>112: CASE constant out of range</b>	Значення цілочислових констант оператора <b>Case</b> повинно знаходитись в межах від $-32768$ до $32767$ .
<b>113: Error in statemen</b>	Символ, на який вказує курсор, не може бути першим символом в операторі.
<b>114: Cannot call an interrupt procedure</b>	Ви не можете безпосередньо викликати процедуру переривання.
<b>115: N/A</b>	
<b>116: Must be in 8087 mode to compile this</b>	Для компіляції необхідний режим 8087.
<b>117: Target adress not found</b>	Вказана адреса не знайдена.
<b>118: Include files are not allowed here</b>	Тут не допускаються вкладені файли.

<b>119: No inherited methods are accessible here</b>	В даному місці програми відсутні наслідкові методи.
<b>120: NIL expected</b>	Очікується оператор <b>Nil</b> .
<b>121: Invalid qualifier</b>	Невірний кваліфікатор.
<b>122: Invalid variable</b>	Недоступне посилання на змінну.
<b>123: Too many symbols</b>	Програма описує більше, ніж 64 Кбайта символів.
<b>124: Statement part too large</b>	Завеликий розділ операторів.
<b>125: N/A</b>	
<b>126: Files must be var parameters</b>	Файли повинні передаватись як параметри-змінні.
<b>127: Too many conditional symbols</b>	Забгато умовних символів.
<b>128: Misplaced conditional directive</b>	Пропущена умовна директива.
<b>129: ENDIF directive missing</b>	Пропущена директива <b>ENDIF</b> .
<b>130: Error in initial conditional defines</b>	Помилка в умовних визначеннях.
<b>131: Header does not match previous definition</b>	Заголовок не відповідає попередньому визначенню.
<b>132: Critical disk error</b>	Під час компіляції відбулась критична помилка файла.
<b>133: Cannot evaluate this expression</b>	Неможливо обрахувати даний вираз.

<b>134: Expression incorrectly germinated</b>	Некоректне завершення виразу.
<b>135: Invalid format specifier</b>	Невірний специфікатор формату.
<b>136: Invalid indirect reference</b>	Недопустима зсилка.
<b>137: Structured variable are not allowed here</b>	Тут не можна використовувати змінну структурного типу.
<b>138: Cannot evaluate without System unit</b>	Неможливо обрахувати вираз без модуля <b>System</b> .
<b>139: Cannot access this symbol</b>	Немає доступу до даного символу.
<b>140: Invalid floating-point operation</b>	Недопустима операція з плаваючою змінною.
<b>141: Cannot compile overlay memory</b>	Неможливо виконати компіляцію оверлейних модулів у пам'ять.
<b>142: Procedure or function variable expected</b>	Потрібно використовувати змінну процедурного типу.
<b>143: Invalid procedure or function reference</b>	Недопустимий вказівник на процедуру або функцію.
<b>144: Cannot overlay this unit</b>	Даний модуль не може використовуватись в якості оверлейного.
<b>145: Too many nested scopes</b>	Забагато вкладень.
<b>146: File access denied</b>	Відмовлено в доступі до файлу.

<b>147: Object type expected</b>	Очікується об'єктний тип.
<b>148: Local object types not allowed</b>	Локальні об'єктні типи не дозволені.
<b>149: VIRTUAL expected</b>	В описанні об'єкта відсутнє слово <b>VIRTUAL</b> .
<b>150: Method identifier expected</b>	Вказаний ідентифікатор не являється ідентифікатором метода.
<b>151: Virtual constructor are not allowed</b>	Конструктор не може бути віртуальним.
<b>152: Constructor identifier expected</b>	Даний ідентифікатор не являється конструктором об'єкта.
<b>153: Destructor identifier expected</b>	Даний ідентифікатор не являється ідентифікатором деструктора.
<b>154: Fail only allowed within constructor</b>	Звернення до стандартної процедури <b>Fail</b> може міститись тільки в конструкторі.
<b>155: Invalid combination of opcode and</b>	Недопустима комбінація коду команди та операндів.
<b>156: Memory reference expected</b>	Відсутня адреса.
<b>157: Cannot add subtract relocatable symbols</b>	Не можна додавати або множити переміщувані символи.
<b>158: Invalid register combination</b>	Недопустима комбінація регістрів.

<b>159: 286/287 instructions are enabled</b>	Недоступний набір команд мікропроцесорів 286/287.
<b>160: Invalid symbol reference</b>	Недопустима зсилка на ідентифікатор.
<b>161: Code generation error</b>	Помилка генерації коду.
<b>162: ASM expected</b>	Очікується ключове слово ASM.



Національний університет  
водного господарства  
та природокористування

## РОЗДІЛ III. Програмування на мовах C, C++, C#

### Лабораторна робота №1

#### Тема: Вирази. Оператори присвоєння. Стандартні функції



Національний університет  
водного господарства  
та природокористування

**Мета роботи:** Ознайомлення студентів з операторами присвоєння, стандартними функціями, записами арифметичних та логічних виразів.

#### Хід роботи:

1. Записати на мові програмування C++ арифметичний та логічний вирази.
2. Відтворити за заданими формулами запис математичного виразу.
3. Відтворити значення числа за його представленням у пам'яті ПК.
4. Виправити синтаксичні помилки у заданому записі арифметичних виразів.
5. Захистити лабораторну роботу.

#### Завдання 1.1

Записати мовою програмування математичний вираз, вибираючи ідентифікатори змінних відповідного типу за замовчуванням (без явного опису типу). Звернути увагу на тип аргументів стандартних функцій.

#### Приклад 1.1

$$l = \frac{1}{e^{-kx+0.5}} - \frac{\lg |k+x| - \sqrt{\sin^3 x}}{\left| \operatorname{arctg} \frac{x+1}{x-k} + \frac{\pi}{10} \cdot \ln \pi \right| + 1} + 2$$

#### Розв'язання:

Запишемо мовою програмування математичний вираз:  
`l:=1/exp(-k*x+0.5) - (ln10(fabs(k+x))-sqrt(power(sin(x),3)))/(fabs(atan((x+1)/  
/x-k))+M_PI*ln(M_PI)/10)+1)+2;`

**Завдання для самостійної роботи** – див.стор. 26.

### Завдання 1.2

Записати мовою програмування даний логічний вираз і визначити значення результату логічних операцій TRUE або FALSE при вказаних значеннях змінних:

$$x - y \leq z + \sqrt{x} \leq 2y \text{ при } x = 1; y = 2; z = 3.$$

**Розв'язання:**

Отримаємо  $((x-y) \leq (z + \sqrt{x})) \ \&\& \ ((z + \sqrt{x}) \leq 2 * y)$

Якщо  $x=1, y=2, z=3$ , то  $1 - 2 \leq 3 + \sqrt{1} \leq 2 * 2, -1 \leq 3 + 1 \leq 4, -1 \leq 4 \leq 4$ .

Результат логічного виразу при заданих значеннях — TRUE.

**Завдання для самостійної роботи** – див.стор. 28.

### Завдання 1.3

У поданих нижче варіантах відтворити арифметичний вираз за його записом.

#### Приклад 1.3

$$Q = \arctan((x + 1)/(y - 2)) + \ln(\text{abs}(k + x)) / \ln(10)$$

**Розв'язання:**

Запишемо початковий вираз:  $Q = \text{arctg} \frac{(x + 1)}{(y - 2)} + \lg|k + x|$ .

**Завдання для самостійної роботи** – див.стор. 29.

### Завдання 1.4

У поданих нижче варіантах відтворити число, представлене в пам'яті ПК.

#### Приклад 1.4

1. 3.45E+02
2. 0.0183907E+5



### Розв'язання:

1. 345;
2. 1839,07

Завдання для самостійної роботи – див.стор. 30.



Національний університет  
водного господарства  
та природокористування

### Завдання 1.5

Виправити синтаксичні помилки в записі арифметичних виразів.

#### Приклад 1.5.

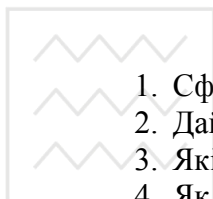
$$Q := \arctg((x + 1)/(y - 2)) + \ln(abc(k + x)) \setminus \ln 10$$

### Розв'язання:

Запишемо вираз після виправлення помилок:

$$Q = \arctan((x + 1)/(y - 2)) + \ln(abs(k + x)) / \ln(10)$$

Завдання для самостійної роботи – див.стор. 31.



### Контрольні запитання:

1. Сформууйте основні етапи розв'язання прикладної задачі на ПК.
2. Дайте означення алгоритму.
3. Які є способи опису алгоритму ?
4. Як позначають операції відношення в C++ ?
5. Який пріоритет операцій в C++ ?
6. Дайте означення типізованої константи та наведіть приклади.

## Лабораторна робота №2

Тема: Лінійні алгоритми та програми

**Мета роботи:** Розробка алгоритмів обчислення значень арифметичних виразів, написання відповідних програмна мові програмування C++, реалізація завдань на ПК.

### Хід роботи:

1. Розробити алгоритм обчислення значення арифметичного виразу.
2. Написати відповідну програму на мові програмування C++.
3. Виконати дане завдання на ПК.
4. Захистити лабораторну роботу.

### Завдання 2.1



Розробити алгоритм та записати відповідну програму обчислення значення арифметичного виразу.

#### Вимоги до програми:

- вхідні дані ввести за допомогою стандартних процедур введення;
- на друк вивести значення вхідної змінної та результати обчислень.

#### Приклад 2.1.

$$y = \frac{x^2 - z^2}{\lg|x - 7|}, x = \frac{\sin^2 a^3 - \arcsin \frac{1}{b}}{\ln|a + b| - 1}, z = \sqrt{\left| \frac{a + b}{ab} \right| + \pi}, a = 3,5, b = -2.16$$

#### Розв'язання:

Запишемо приклад розв'язання даного завдання на мові програмування C++.

```
#include <iostream.h>
#include <math.h>
int main()
{
    float x, y, z, a, b;

    cout<<"a=";
    cin>>a;

    cout<<"b=";
    cin>>b;

    x=(power(sin(power(a,3)),2)-asin(1/b)/(ln(fabs(a+b))-1);
    cout<<"x"<<x<<endl;

    y=(x*x-z*z)/(ln(fabs(x-7))/ln10);
    cout<<"y"<<y<<endl;
```

```
z=sqrt(fabs((a+b)/(a*b))+M_PI);
cout<<"z="<<z<<endl;
```

```
return 0;
}
```

**Завдання для самостійної роботи** – див.стор. 33.



Національний університет  
водного господарства  
та природокористування

## Завдання 2.2.

Для даного завдання розробити алгоритм та написати відповідну програму з виведенням результатів на екран.

### *Вимоги до програми:*

- вхідні дані ввести за допомогою стандартних процедур введення;
- на друк вивести значення вхідної змінної та результати обчислень;
- вхідні дані взяти довільними.

### **Приклад 2.2**

Скласти програму обчислення процента успішності студентів групи, середнього балу і кількості студентів-двієчників, якщо задані кількість студентів в групі, що отримали “5”, “4”, “3” на іспиті з інформатики.

### **Розв’язання:**

Запишемо приклад розв’язання даного завдання на мові програмування C++.

```
#include <iostream.h>
int main()
{
    int k, k5, k4, k3, k2;
    float v5, v4, v3, v2;
```

```
    cout<<"Кількість студентів групи: ";
    cin>>k;
```

```
    cout<<"Введіть кількість студентів, які отримали 5-ки : ";
    cin>>k5;
```

```
    cout<<"Введіть кількість студентів, які отримали 4-ки : ";
    cin>>k4;
```

```
cout<<"Введіть кількість студентів, які отримали 3-ки : ";
cin>>k3;
```

```
k2=k-(k5+k4+k3);
cout<<"В групі " << k2 << "двійшників";
```

```
//Обчислення відсотків успішності
```

```
v5=k5*100/k;
```

```
v4=k4*100/k;
```

```
v3=k3*100/k;
```

```
v2=k2*100/k;
```

```
cout<<"Відсоток студентів, які отримали 5-ки : "<<v5<<endl;
```

```
cout<<"Відсоток студентів, які отримали 4-ки : "<<v4<<endl;
```

```
cout<<"Відсоток студентів, які отримали 3-ки : "<<v3<<endl;
```

```
cout<<"Відсоток студентів, які отримали 2-ки : "<<v2<<endl;
```

```
return(0);
```

```
}
```

**Завдання для самостійної роботи – див.стор. 36.**

### **Контрольні запитання:**

1. Назвіть основні алгоритмічні структури.
2. Перелічіть основні властивості алгоритму та розкрийте їх зміст.
3. Дайте означення алфавіту мови програмування.
4. Що складає алфавіт мови C++ ?
5. Наведіть загальну структуру C++-програми.

## **Лабораторна робота №3**

**Тема: Алгоритми та програми розгалуженої структури**

**Мета роботи:** Освоєння алгоритмів та програм розгалуженої структури, складання програм з використанням операторів умовного та безумовного переходу.

### Хід роботи:

1. Розробити алгоритми для поданих нижче завдань.
2. Написати відповідні програми на мові програмування C++.
3. Визначені викладачем завдання виконати на ПК.
4. Захистити лабораторну роботу.

### Завдання 3.1

Розробити алгоритм та записати відповідну програму знаходження значення функції, яка обчислюється залежно від значення аргумента.



Національний університет  
водного господарства  
та природокористування

#### Вимоги до програми:

- вхідні дані ввести за допомогою стандартних процедур введення;
- на друк вивести значення вхідної змінної та результати обчислень;
- вхідні дані взяти довільними.

### Приклад 3.1

$$y = \begin{cases} -\cos^2(x - \pi), & x < -\frac{\pi}{4}; \\ \sqrt{|x + 1|}, & -\frac{\pi}{4} \leq x \leq 1; \\ \frac{1}{x - 1}, & x > 1. \end{cases}$$

#### Розв'язання:

Запишемо приклад розв'язання даного завдання на мові програмування

```
#include <iostream.h>
#include <math.h>

int main()
{
    float x, y;
    cout<<"Enter x: ";
    cin>>x;
    if (x>1)
        y=1/(x-1);
    else if (x>=-M_PI/4)
        y=sqrt(fabs(x+1));
    else
```

```

    y=-power(cos(x-M_PI), 2);
    cout<<"y="<<y;
    return 0;
}

```

**Завдання для самостійної роботи** – див.стор. 40.



Національний університет  
водного господарства  
та природокористування

### Завдання 3.2

Розробити алгоритм та записати відповідну програму для поданих нижче завдань.

#### *Вимоги до програм:*

- ввести і вивести задану інформацію ( три числа, в яких  $k$  - номер групи,  $l$ -номер варіанта);
- виконати завдання в кожному конкретному варіанті.

#### **Приклад 3.2**

Вибрати серед чисел додатні та надрукувати їх та їхні подвоєння.

$$a = \frac{1-3k}{5}, \quad b = \frac{2l+k}{k}, \quad d = lk + 6,5 .$$

Вибрати серед чисел додатні та надрукувати їх та їхні подвоєння.

#### **Розв'язання:**

Запишемо приклад розв'язання даного завдання на мові програмування C++.

```

#include <iostream.h>
#include <math.h>

int main()
{
    float l, k, a, b, d;
    cout<<"Enter l: ";
    cin>>l;
    cout<<"Enter k: ";
    cin>>k;
    a=(1-3*k)/5;
    b=(2*l+k)/k;
    d=l*k+6.5;
    if (a>0)
        cout<<"2*a="<<2*a;
}

```

```

if (b>0)
    cout<<"2*b="<<2*b;
if (d>0)
    cout<<"2*d="<<2*d;
return 0;
}

```

**Завдання для самостійної роботи** – див.стор. 44.



Національний університет  
водного господарства  
та природокористування

### Завдання 3.3

Розробити алгоритм і програму на одній з алгоритмічних мов, щоб виявити належність точки  $M(x,y)$  геометричній фігурі. Координати точки  $M$  та вид фігури наведені нижче.

#### *Вимоги до програми*

- вхідні дані ввести за допомогою стандартних процедур введення;
- на друк вивести значення вхідної інформації та результати роботи програми у вигляді повідомлення про належність або неналежність точки фігурі.

#### Приклад 3.3



#### Розв'язання:

Наведемо код програми для розв'язання даного завдання на мові програмування C++.

```

#include <iostream.h>
#include <math.h>

```

```

int main()
{
    float x, y, r;
    cout<<"Enter r: ";
}

```

```

cin>>r;
cout<<"Enter x: ";
cin>>x;
cout<<"Enter y: ";
cin>>y;

if ((x>=0) && (y>=0) && (power(x,2)+power(y,2)<=power(r,2)))
    cout<<"точка належить фігурі";
else
    cout<<"точка не належить фігурі";
return 0;
}

```

**Завдання для самостійної роботи** – див.стор. 47.

### Завдання 3.4

Для даного завдання розробити алгоритм та написати відповідну програму з виведенням результатів на екран.

#### *Вимоги до програми:*

- вхідні дані ввести за допомогою стандартних процедур введення ;
- на друк вивести значення вхідної інформації та результати обчислень;
- вхідні дані взяти довільними, в межах допустимих, якщо вони явно не задані в умові задачі.

#### **Приклад 3.4**

Прибуток підприємств становить відповідно a,b,c (грошових одиниць). Визначити найбільший прибуток підприємств.

#### **Розв'язання:**

Запишемо приклад розв'язання даного завдання на мові програмування C++.

```
#include<iostream.h>
```

```

int main ( )
{
    float a, b, c, max;

    cout<<"Enter a=";
    cin>>a;

```



```

cout<<"Enter b="";
cin>>b;
cout<<"Enter c="";
cin>>c;

if (a>b)
    max=a;
else
    max=b;
if (c>max)
    max=c;

cout<<"max="<<max;
return 0;
}

```



Національне агентство  
водного господарства  
та природокористування

**Завдання для самостійної роботи** – див.стор. 51.

### Завдання 3.5

Для даного завдання розробити алгоритм та написати відповідну програму з виведенням результатів на екран.

#### *Вимоги до програми:*

- вхідні дані ввести за допомогою стандартних процедур введення;
- на друк вивести значення вхідної інформації та результати обчислень;
- вхідні дані взяти довільними, в межах допустимих, якщо вони явно не задані в умові задачі.

#### **Приклад 3.5**

Скласти програму, яка визначала б, яких оцінок більше отримано під час іспиту з інформатики “4” чи “5”.

#### **Розв’язання:**

Запишемо приклад розв’язання даного завдання на мові програмування C++.

```

#include <iostream.h>

int main()
{
    float k5, k4;
    cout<<"Введіть кількість 5-ок: ";

```

```

cin>>k5;
cout<<"Введіть кількість 4-ок: ";
cin>>k4;

if (k5>k4)
    cout<<"5-ок більше";
else if (k4>k5)
    cout<<"4-ок більше";
else
    cout<<"кількість оцінок однакова";

return 0;
}

```



Національний університет  
водного господарства  
та природокористування

**Завдання для самостійної роботи – див.стор. 54.**

### ***Контрольні запитання***

1. Що таке алгоритмізація?
2. Які є способи задання даних в C++-програмі?
3. Як організувати „підказку” в C++-програмі при введенні даних?
4. Що таке багатоальтернативний оператор if...else?
5. Як реалізовується оператор вибору?



Національний університет  
водного господарства  
та природокористування

### **Лабораторна робота № 4**

**Тема: Циклічні алгоритми та програми**

**Мета роботи:** Розглянути циклічні алгоритми та програми та реалізувати запропоновані програми на ПК за допомогою циклів з передумовою та післяумовою та оператора for.

**Хід роботи:**

1. Розробити алгоритми згідно нижче наведених завдань.
2. Написати відповідні програми на мові програмування C++.
3. Визначені викладачем завдання виконати на ПК.
4. Захистити лабораторну роботу.

### Завдання 4.1.

Розробити алгоритм та записати програму обчислення значення аргумента функції на вказаному проміжку із заданим кроком, а також значень функції (табулювання функції), починаючи із заданої точки.

#### *Вимоги до програми:*

- вхідні дані (початкове, кінцеве значення та крок його зміни аргументу, кількість обчислюваних значень функції) ввести стандартною процедурою введення;
- вивести у вигляді таблиці значення аргументу та відповідного йому значення функції.

#### Приклад 4.1

$$y = \frac{\cos^2 x}{x^2 + 1},$$

- а)  $3,8 \leq x \leq 7,6$  ;  $\Delta x = 0,6$  ;
- б)  $x \geq 0,5$  ,  $\Delta x = 0,1$  ,  $n = 9$  .

#### Розв'язання:

Для випадку а) використано оператор циклу з передумовою:

```
#include <iostream.h>
#include <math.h>

int main()
{
    float x, y, xp, xk, dx;
    cout<<"Введіть початкове значення x: ";
    cin>>xp;
    cout<<"Введіть кінцеве значення x: ";
    cin>>xk;
    cout<<"Введіть крок зміни аргументу dx: ";
    cin>>dx;

    x=xp;
    while(x<=xk)
    {
        y=power(cos(x),2)/(x*x+1);
        cout<<"x="<<x<<"y="<<y<<endl;
        x=x+dx;
    }
    return 0;
}
```

```
}
```

Для випадку б) використано оператор циклу з параметром:

```
#include <iostream.h>
#include <math.h>

int main()
{
    float x, y, xp, xk, dx;
    cout<<"Введіть початкове значення x: ";
    cin>>xp;
    cout<<"Введіть крок зміни аргументу dx: ";
    cin>>dx;
    cout<<"Введіть кількість кроків: ";
    cin>>n;

    x=xp;
    for(int i=1; i<=n; i++)
    {
        y=power(cos(x),2)/(x*x+1);
        cout<<"x="<<x<<"y="<<y<<endl;
        x=x+dx;
    }
    return 0;
}
```

**Завдання для самостійної роботи** – див.стор. 58.

## Завдання 4.2

Розробити алгоритм та записати програму обчислення значення функції із заданим кроком на вказаних проміжках.

### **Вимоги до програми:**

- межі із області визначення функції і крок зміни аргументу ввести стандартною процедурою введення;
- вивести у вигляді таблиці пари чисел: (аргумент, значення функції).

### Приклад 4.2

$$y = \begin{cases} x + \sin x, & -6,5 < x < 0,5 \\ \ln(x + \sqrt[3]{x}), & 0,5 \leq x \leq 8; \end{cases}$$

$\Delta x = 0,5;$

#### Розв'язання:



Національний університет  
водного господарства  
та природокористування

```
#include <iostream.h>
#include <math.h>
```

```
int main()
{
    float x, y, x1, x2, x3, dx;
    cout<<"Введіть x1: ";
    cin>>x1;
    cout<<"Введіть x2: ";
    cin>>x2;
    cout<<"Введіть x3: ";
    cin>>x3;
    cout<<"Введіть dx: ";
    cin>>dx;

    x=x1;
    while(x<=x3)
    {
        if (x<x2)
            y=x+sin(x);
        else
            y=ln(x+power(x, 1.0/3));
        cout<<"x="<<x<<"y="<<y<<endl;
        x=x+dx;
    }
    return 0;
}
```

**Завдання для самостійної роботи** – див.стор. 62.

### Завдання 4.3

Для даного завдання розробити алгоритм та написати відповідну

програму з виведенням результатів на екран.

### **Вимоги до програми:**

- вхідні дані ввести за допомогою стандартних процедур введення;
- на друк вивести значення вхідної інформації та результати обчислень;
- вхідні дані взяти довільними, в межах допустимих, якщо вони явно не задані в умові задачі.



Національний  
водного транспорту  
та природоохоро-

### **Приклад 4.3**

Скласти таблицю для визначення суми продажу від кількості проданого товару. Ця залежність визначається формулою  $S=K*C$ , де  $S$  — сума продажу,  $K$  — кількість товару,  $C$  — ціна одиниці товару (вводиться з клавіатури). В таблиці значення  $K$  змінюється від 1кг до 25 кг з кроком 1кг.

### **Розв'язання:**

```
#include <iostream.h>
#include <math.h>

int main()
{
    float s, k, c, k1, k2, dk;
    cout<<"Введіть початкове значення K: ";
    cin>>k1;
    cout<<" Введіть кінцеве значення K: ";
    cin>>k2;
    cout<<" Введіть крок зміни K: ";
    cin>>dk;

    cout<<"Введіть ціну одиниці товару C: ";
    cin>>c;

    x=xp;
    for(k=k1; k<=k2; k+=dk)
    {
        s=k*c;
        cout<<"k="<<k<<"s="<<s<<endl;
    }
    return 0;
}
```

**Завдання для самостійної роботи** – див.стор. 65.

## Завдання 4.4

Розробити алгоритм та записати програму обчислення значення аргументів функції на вказаних проміжках із заданими кроками, а також значень функції (табулювання функції двох змінних), починаючи із заданої точки.

### Вимоги до програми:

- вхідні дані (початкові, кінцеві значення аргументів та кроки їх зміни) ввести стандартною процедурою введення;
- передбачити визначення області допустимих значень функції;
- вивести у вигляді таблиці значення аргументів та відповідне їм значення функції.

### Приклад 4.4

$$z = \sqrt{e^{xy} + 1}; -2 \leq y \leq 10, \Delta y=1, 2.1 \leq x \leq 4.6, \Delta x=0.2;$$

### Розв'язання:

```
#include <iostream.h>
#include <math.h>
```

```
int main()
{
    float x, y, z, x1, x2, y1, y2, dx, dy;

    cout<<"Введіть x1: ";
    cin>>x1;
    cout<<"Введіть x2: ";
    cin>>x2;
    cout<<"Введіть dx: ";
    cin>>dx;

    cout<<"Введіть y1: ";
    cin>>y1;
    cout<<"Введіть y2: ";
    cin>>y2;
    cout<<"Введіть dy: ";
    cin>>dy;

    for(x=x1; x<=x2; x+=dx) //використання вкладених циклів
        for(y=y1; y<=y2; y+=dy)
            {
                z=sqrt(exp(x*y)+1);
```

```

cout<<"x="<<x<<" y="<<y<<" z="<<z<<endl;
}

return 0;
}

```

**Завдання для самостійної роботи** – див.стор. 69.



Національний університет  
водного господарства  
та природокористування

### **Контрольні запитання**

1. Яка відмінність роботи алгоритму „циклу з передумовою” в порівнянні з „циклом з післяумовою” ?
2. Назвіть основні особливості оператора циклу з параметром.
3. Зобразити алгоритмічну структуру, що відповідає оператору *while*.

## **Лабораторна робота №5**

**Тема: Типові прийоми програмування**

**Мета роботи:** Ознайомитися з основними типовими прийомами програмування: обчислення значення сум та добутків, та навчитися їх застосовувати на практиці.

### **Хід роботи:**

1. Розробити алгоритми для поданих нижче завдань обчислення сум та добутків.
2. Написати відповідні програми на мові програмування C++.
3. Визначені викладачем завдання виконати на ПК.
4. Захистити лабораторну роботу.

### **Завдання 5.1**

Розробити алгоритм та записати програму обчислення значення суми та добутку.

#### **Вимоги до програми:**

- вхідні дані ( початкове і кінцеве значення індексної змінної в сумі чи добутку ) ввести стандартною процедурою введення;
- вивести обчислені значення суми та добутку .

### **Приклад 5.1**



$$y = \sum_{i=1}^{10} \frac{i^2 + 1}{i^3 + 2}, \quad f = \prod_{k=m}^n \frac{k + 3}{(k + 5)(k + 6)},$$

$m = 5, n = 13;$

### Розв'язання:



#include <iostream.h>  
 #include <math.h>

```

int main()
{
    const int n=7;
    int i, k, n, m;
    float s, p;
    cout<<"Введіть n: ";
    cin>>n;
    cout<<"Введіть m: ";
    cin>>m;

    s=0; p=1;
    for (i=1; i<=n; i++)
    {
        s=s+(i*i+1)/(i*i*i+2); //обчислення суми
    }
    for (k=n; k<=m; k++)
    {
        p=p*(k+3)/((k+5)*(k+6)); //обчислення добутку
    }

    cout<<"suma="<<s<<endl;
    cout<<"dobutok="<<p<<endl;

    return 0;
}
  
```

**Завдання для самостійної роботи** – див.стор. 71.

### Завдання 5.2

Для даного завдання розробити алгоритм та написати відповідну

програму з виведенням результатів на екран.

**Вимоги до програми:**

- вхідні дані ввести за допомогою стандартних процедур введення;
- на друк вивести значення вхідної інформації та результати обчислень;
- вхідні дані взяти довільними, в межах допустимих, якщо вони явно не задані в умові задачі.



Національний  
університет  
водного господарства  
та природокористування

**Приклад 5.2**

Знайти суму квадратів чисел натурального ряду від **a** до **b**.

**Розв'язання:**

```
#include <iostream.h>

int main( )
{
    int s, a, b;
    cout<<"Введіть a: ";
    cin>>a;
    cout<<"Введіть b: ";
    cin>>b;
    s=0;
    for(int i=a; i<=b; i++)
        s+=i*i;
    cout<<"сума квадратів = "<<s;
    return 0;
}
```

**Завдання для самостійної роботи** – див.стор. 73.

**Завдання 5.3**

Для даного завдання розробити алгоритм та написати відповідну програму для обчислення значення суми без використання масиву з виведенням результатів обчислення на екран.

**Вимоги до програми:**

- вхідні дані (початкове значення аргумента, крок зміни аргумента) ввести стандартною процедурою введення;
- вивести обчислене значення суми.

### Приклад 5.3

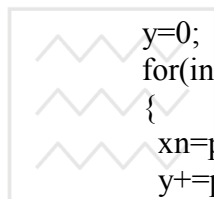
$$y = \sum_{i=2}^k (x_i - x_{i-1})^2, \text{ де } x_1 = 0,15; x_{i+1} = \sqrt[3]{x_i + \Delta x} \quad ; \quad \Delta x = 0,05.$$

#### Розв'язання:



```
#include <iostream.h>
#include <math.h>
```

```
int main()
{
    int k;
    float x1, dx, x, y, xn;
    cout<<"Введіть k: ";
    cin>>k;
    cout<<"Введіть x1: ";
    cin>>x;
    cout<<"Введіть dx: ";
    cin>>dx;
```



```
y=0;
for(int i=2; i<=k; i++)
{
    xn=power(x+dx, 1.0/3);
    y+=power(xn-x, 2);
    xn=x;
}
```

```
cout<<"y = "<<y;
return 0;
}
```

**Завдання для самостійної роботи** – див.стор. 76.

### Завдання 5.4

Скласти алгоритм і написати програму обчислення значення виразу (завдання з п. 5.3) з використанням всіх відомих операторів циклу.

## Контрольні запитання

1. Які цикли називаються вкладеними?
2. За допомогою яких операторів циклу можна організувати обчислення суми та добутку?

## Лабораторна робота №6



Національний університет  
водного господарства

### Тема: Програмування ітераційних циклічних обчислювальних процесів

**Мета роботи:** ознайомитися з програмуванням ітераційних обчислювальних процесів, знаходженням границь, нескінченних сум.

#### Завдання:

Розробити алгоритми для поданих нижче завдань обчислення значення виразів з використанням ітераційних обчислювальних процесів. Написати відповідні програми на мові програмування C++. Визначені викладачем завдання виконати на ПК. Захистити лабораторну роботу.

#### Завдання 6.1

Розробити алгоритм та записати програму для обчислення значення кореня  $n$ -го степеня  $y = \sqrt[n]{x}$  із заданою точністю  $\epsilon$ , користуючись ітераційною формулою:

$$y_{i+1} = \frac{1}{n} \left( \frac{x}{y_i^{n-1}} + (n-1)y_i \right), \quad y_0 = x, \quad i = 0, 1, \dots, n$$

#### Вимоги до програми:

- вхідні дані (початкове значення аргументу, крок зміни аргументу) ввести стандартною процедурою введення;
- вивести обчислене значення кореня;
- обчислення по даній формулі завершити, якщо  $|y_{i+1} - y_i| \leq \epsilon$ .

#### Розв'язання:

```
#include <iostream.h>
#include <math.h>
int main( )
{
```

```

float x, y, y1, eps;
int n;

cout<<"Введіть значення x";
cin>>x;
cout<<"Введіть точність eps";
cin>>eps;
cout<<"Введіть степінь n";
cin>>n;
y1=x;

do {
    y=y1;
    y1=(1.0/n)*(x/pow(y, n-1)+(n-1)*y);
} while (fabs(y1-y)>eps)
cout<<"Значення кореня дорівнює "<<y1;
return 0;
}

```



Національний університет  
водного господарства  
та природокористування

**Завдання для самостійної роботи – див.стор. 80.**

### Завдання 6.2

Скласти програму обчислення кореня нелінійного рівняння  $f(x) = 0$ , користуючись однією з ітераційних формул, якщо початкове наближення кореня  $x^{(0)}$  або проміжок ізоляції кореня  $[a, b]$  задані. Обчислення припинити, якщо різниця двох послідовних наближень до кореня не перевищує заданої точності  $\varepsilon$ .

$$|x_{i+1} - x_i| < \varepsilon.$$

#### **Вимоги до програми:**

- вхідні дані (початкове значення кореня, точність обчислень) ввести стандартною процедурою введення;
- вести обчислене значення кореня та кількість ітерацій;
- кожному конкретному варіанту для уточнення кореня скористатись заданою ітераційною формулою.

#### **Приклад 6.2**

$$X^{(i+1)} = f(x^{(i)}), \quad i=0, 1, 2, \quad X = \arctg x + 1, \quad X^{(0)} = 1.5$$

## Розв'язання:

```
#include <iostream.h>
#include <math.h>

int main( )
{
    float x, x0, x1, eps;
    int n;
    cout<<"Введіть значення x0";
    cin>>x0;
    cout<<"Введіть точність eps";
    cin>>eps;

    x1=x0;

    do {
        x=x1;
        x1=arctan(x)-x+1;
        n++;
    } while (fabs(x1-x)>eps)
    cout<<"Значення кореня дорівнює "<<x1;
    cout<<"Кількість ітерацій "<<n;
    return 0;
}
```

**Завдання для самостійної роботи** — див. стор. 81.

### Завдання 6.3

Скласти програму обчислення значення функції, яка представляється у вигляді нескінченного ряду  $f(x) = \sum_{k=0}^{\infty} a_k(x)$ , із заданою точністю  $\varepsilon = 0,01$  і числа  $k$  членів вказаної суми. Ітераційний процес завершити, якщо

$$|a_k(x)| \leq \varepsilon.$$

#### **Вимоги до програми:**

- вхідні дані (початкове значення змінної, точність обчислень) ввести стандартною процедурою введення;

- вивести обчислене значення функції та значення змінної, в якій вона обчислюється, а також кількість проведених ітерацій;
- у кожному конкретному варіанті для обчислення значення функції при проведенні проміжних обчислень члена суми по можливості скористатися рекурентною формулою.

### Приклад 6.3



Національний університет  
водного господарства  
та природокористування

$$f(x) = e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots; \quad x = 1,5$$

#### Розв'язання:

```
#include <iostream.h>
#include <math.h>
```

```
int main()
{
    float x, f, a, eps;
    int i;
```

```
    cout<<"Введіть значення x";
    cin>>x;
```

```
    cout<<"Введіть точність eps";
    cin>>eps;
```

```
    f=1;
    a=1;
    i=0;
```

```
    do {
        i++;
        a*=x/i;
        f+=a;
    } while (fabs(a)>eps)
    cout<<"f="<<f;
    return 0;
}
```

**Завдання для самостійної роботи** – див.стор. 83.

### Контрольні запитання:

1. Які цикли називають ітераційними?
2. Які оператори циклу використовуються для реалізації циклічних алгоритмічних структур?
3. Що таке тіло циклу?

### Лабораторна робота №7



#### Тема: Одновимірні масиви

**Мета роботи:** Ознайомитися з особливостями роботи одновимірних масивів, навчитися виконувати обробку одновимірних масивів, здійснювати сортування масивів.

#### Хід роботи:

1. Розробити алгоритми обробки одновимірного масиву згідно варіантів завдань для самостійної роботи.
2. Написати відповідні програми на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.

#### Завдання 7.1

Скласти алгоритм і написати програму табулювання функції з використанням масивів (для запам'ятовування в масивах аргументів та обчислених відповідних значень функції). Варіанти завдань взяти з п. 5.1.

#### Приклад 7.1

Розробити алгоритм і програму обчислення суми та добутку.

$$y = \sum_{i=1}^{10} \frac{i^2 + 1}{i^3 + 2}, \quad f = \prod_{k=m}^n \frac{k + 3}{(k + 5)(k + 6)},$$

$m = 5, n = 13.$

#### Завдання 7.2

#### Приклад 7.2

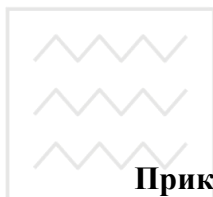
Знайти суму додатніх елементів масиву  $S(6)=(5.0,-2.3,-6.9,-1.1,2.0,6.6)$ .



### Розв'язання:

```
#include <iostream.h>
int main()
{
    int i;
    float C[6];
    float s=0;
    for(i=0; i<6; i++)
    {
        cout << "C[" << i << "]=";
        cin >> C[i];
    }
    for (i=0; i<6; i++)
        if (C[i]>0)
            s+=C[i];
    cout << "S=" << s;
}
```

**Завдання для самостійної роботи** – див.стор. 91.



Національний університет  
водного господарства  
та природокористування

### Завдання 7.3

#### Приклад 7.3

Розробити алгоритм та записати програму обчислення суми кубів додатних, добутку квадрату від'ємних та кількість нульових елементів одновимірного масиву.

### Розв'язання:

```
#include <iostream.h>
#include <math.h>

int main()
{
    int i, k=0;
    float C[6];
```

```

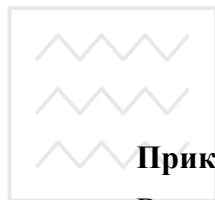
for(i=0; i<6; i++)
{
    cout << "C[" << i << "]=";
    cin >> C[i];
}
float s=0, d=1;
for (i=0; i<6; i++)
    if (C[i]>0)
        s+=power(C[i], 3);
    else (C[i]<0)
        d*=power(C[i], 2);
    else
        k++;

cout << "сума кубів додатних =" << s << endl;
cout << "добутку від'ємних =" << d << endl;
cout << "кількість нульових елементів =" << k << endl;

}

```

**Завдання для самостійної роботи** – див.стор. 93.



Національний університет  
водного господарства  
та природокористування

#### Завдання 7.4

#### Приклад 7.4

Впорядкувати елементи масиву  $C(6) = (6,3; 5,3; -2,2; -3,3; 0,0; 2,1)$ , розмістити їх по порядку зростання значень в тому ж масиві.

#### Розв'язання:

```

#include <iostream.h>
int main()
{
    int i, j;
    float C[6];
    for(i=0; i<6; i++)
    {
        cout << "C[" << i << "]=";
        cin >> C[i];
    }
}

```

```

for (i=0; i<5; i++)
  for (j=i+1; j<6; j++)
    if (C[i]<C[j])
      {
        t = C[i];
        C[i] = C[j];
        C[j] = t;
      }
}
for(i=0; i<6; i++)
{
  cout << "C[" << i << "]=" << C[i]<<endl;
}
}

```

**Завдання для самостійної роботи** – див.стор. 96.

## Завдання 7.5

### Приклад 7.5

У масиві C(n) (n<=9) визначити середнє арифметичне додатніх елементів.

#### Розв'язання:

```

#include <iostream.h>
#include <math.h>

int main()
{
  int n, i, k=0;
  float C[10];
  cout << "Введіть n (<=9) = ";
  cin >> n;
  for(i=0; i<n; i++)
  {
    cout << "C[" << i << "]=";
    cin >> C[i];
  }
  float s=0;
  for (i=0; i<n; i++)

```

```
if (C[i]>0)
{
    s+=C[i];
    k++;
}
```

```
if (k>0)
    cout << "середнє арифметичне =" << s/k;
else
    cout << "додатніх елементів не знайдено";
}
```

**Завдання для самостійної роботи** – див. стор. 98.

### **Контрольні запитання:**

1. Дайте визначення масиву.
2. Як здійснюється ввід та вивід одновимірних масивів?
3. Як здійснюється пошук мінімального та максимального елементів масиву?
4. Назвіть основні алгоритми сортування.

**Тема: Багатовимірні масиви**

**Мета роботи:** Ознайомитися з особливостями роботи двовимірних масивів, навчитися здійснювати сортуванням масиву та виконувати найпростіші дії над елементами масиву.

### **Хід роботи:**

1. Розбити алгоритми обробки двовимірного масиву згідно варіантів завдань для самостійної роботи.
2. Написати відповідні програми на одній з мов програмування.
3. Виконати дане завдання на комп'ютері .
4. Захистити лабораторну роботу.

## Завдання 8.1

Для кожного завдання, наведеного нижче, розробити алгоритм і написати відповідну програму.

### Приклад 8.1

В матриці  $C(3,4) = \begin{pmatrix} -1 & 2,5 & 1,6 & -4 \\ 0 & 0,5 & 0 & 1,3 \\ 1 & -3 & 0,4 & 0 \end{pmatrix}$  підрахувати і вивести на друк

суму додатних та від'ємних елементів, підрахувати кількість нульових елементів.

### Розв'язання:

```
#include <iostream.h>
int main()
{
    int i, j;
    float C[3][4];
    for(i=0; i<3; i++)
        for(j=0; j<4; j++)
        {
            cout << "C[" << i << "]" << " [" << j << "]=" << endl;
            cin >> C[i][j];
        }
    float sd=0, sv=0;
    int kn=0;
    for (i=0; i<3; i++)
        for (j=0; j<4; j++)
            if (C[i][j]<0)
                sv+=C[i][j];
            else if (C[i][j]>0)
                sd+=C[i][j];
            else
                kn++;
    cout << "сума додатних =" << sd << endl;
    cout << "сума від'ємних =" << sv << endl;
    cout << "кількість нульових елементів =" << kn << endl;
}
```

**Завдання для самостійної роботи** – див. стор. 102.

## Завдання 8.2

Скласти програму знаходження числових характеристик матриці згідно вказаної умови.

### Приклад 8.2

В матриці  $A(n,n)$ , ( $n \leq 9$ ) знайти суму всіх додатних елементів, крім тих, що лежать на головній діагоналі.



національний університет  
водного господарства  
та природокористування

#### Розв'язання:

```
#include <iostream.h>
int main()
{
    int i, j, n;
    float C[9][9];
    cout << "Введіть n (<=9) = ";
    cin >> n;

    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
        {
            cout << "C[" << i << "]" << j << "]=";
            cin >> C[i][j];
        }
    float sd=0;
    for (i=0; i<n; i++)
        for (j=0; j<n; j++)
            if (i!=j)
                if (C[i][j]<0)
                    sd+=C[i][j];

    cout << "сума =" << sd << endl;
}
```

**Завдання для самостійної роботи** – див. стор. 106.

## Завдання 8.3

Скласти програму обробки та перетворення двовимірного масиву. Передбачити виведення початкової та перетвореної матриці.

### Приклад 8.3

Матриця  $A(m,n)$  ( $m \leq 9$ ,  $n \leq 9$ ) містить додатні та від'ємні елементи. Сформуванати одновимірні масиви:  $B$ , який містить лише додатні елементи масиву  $A(m,n)$ , масив  $C$ , який містить лише від'ємні елементи масиву  $A(m,n)$ . Підрахувати кількість елементів в даних масивах.

#### Розв'язання:

```
#include <iostream.h>
```

```
int main()
```

```
{
```

```
int i, j, m, n;
```

```
float A[9][9];
```

```
float B[81], C[81];
```

```
cout << "Введіть m (<=9) = ";
```

```
cin >> m;
```

```
cout << "Введіть n (<=9) = ";
```

```
cin >> n;
```

```
for(i=0; i<m; i++)
```

```
for(j=0; j<n; j++)
```

```
{
```

```
cout << "A[" << i << "][" << j << "]=";
```

```
cin >> A[i][j];
```

```
}
```

```
int kd=0, kv=0;
```

```
for (i=0; i<m; i++)
```

```
for (j=0; j<n; j++)
```

```
if (A[i][j]>0)
```

```
{
```

```
B[kd] = A[i][j];
```

```
kd++;
```

```
} else if (A[i][j]<0)
```

```
{
```

```
C[kv] = A[i][j];
```

```
kv++;
```

```
}
```

```
cout << "кількість додатних =" << kd << endl;
```

```
cout << "кількість від'ємних =" << kv << endl;
```

```
}
```

Завдання для самостійної роботи – див.стор. 108.

### Завдання 8.4

Інформація про прибутки М фірм за N місяців задана у вигляді таблиці



Національний університет  
водного господарства  
та природокористування

<i>січень</i>	<i>лютий</i>	...	...	
$Z_{11}$	$Z_{12}$	...	$Z_{1n}$	- прибутки 1-ї фірми
$Z_{21}$	$Z_{22}$	...	$Z_{2n}$	- прибутки 2-ї фірми
...	...	...	...	
$Z_{m1}$	$Z_{m2}$	...	$Z_{mn}$	- прибутки m-ї фірми

Скласти алгоритм та відповідну програму згідно нижче наведеного варіанта.

Завдання для самостійної роботи – див.стор. 111.



Національний університет  
водного господарства  
та природокористування

### Завдання 8.5

#### Приклад 8.5

Дано тензор  $T(2 \times 3 \times 2) = \left\{ \begin{pmatrix} 1 & 0 & 2 \\ 3 & -1 & 1 \end{pmatrix}, \begin{pmatrix} 3 & 0 & 4 \\ 5 & 2 & 3 \end{pmatrix} \right\}$ . Знайти суму його

елементів, індекси яких парні.

**Розв'язання:**

```
#include <iostream.h>
int main()
{
    int i, j, k;
    float T[2][3][2];

    for(i=0; i<2; i++)
        for(j=0; j<3; j++)
```



```

for(k=0; k<2; k++)
{
    cout << "T[" << i << "]" << j << "]" << k << "]" =";
    cin >> T[i][j][k];
}

```

```
int s=0;
```



```

for(i=0; i<2; i++)
for(j=0; j<3; j++)
for(k=0; k<2; k++)
    if (i%2==0 && j%2==0 && k%2==0)
        s+=T[i][j][k];

cout << "сума =" << s << endl;
}

```

**Завдання для самостійної роботи** – див.стор. 114.

### **Контрольні запитання:**

1. Що таке тензор?
2. Як здійснюється ввід та вивід багатовимірних масивів?
3. Назвіть основні особливості ініціалізації значень масиву?
4. Як зберігаються багатовимірні масиви в оперативній пам'яті ПК?
5. Наведіть алгоритм сортування масиву методом бульбашки.

### **Лабораторна робота №9**

**Тема: Програми з використанням підпрограм**

**Мета роботи:** Ознайомитися з особливостями роботи підпрограм, навчитися створювати підпрограми-процедури та процедури-функції, зробити висновки щодо доцільності використання підпрограм.

#### **Хід роботи:**

1. Розробити алгоритм обчислення деяких величин з використанням підпрограм згідно варіантів завдань для самостійної роботи.

2. Написати відповідні програми на одній з мов програмування.
3. Виконати дане завдання на комп'ютері .
4. Захистити лабораторну роботу.

### Завдання 9.1

За умовою завдання виконати обчислення, які доцільно оформити у вигляді нестандартних функцій, реалізувати їх засобами алгоритмічної мови.



#### Приклади 9.1

Обчислити величину 
$$S = \frac{\sqrt{\left(\sum_{i=1}^5 t_i\right)^2 + A^2}}{\sqrt{A^2 + B^2}} + \frac{\sqrt{B^2 + \left(\sum_{i=1}^7 t_i\right)^2}}{\sqrt{\left(\sum_{i=1}^6 t_i\right)^2 + C^2}},$$
 де  $A=7,6$ ,  $B=-8,9$ ,

$C=3,65$ ;  $\{t_i\} = (5,2; -7,1; 8,3; -3,4; 7,5; -8,4; 6,0; 5,2; 1,9; 4,3)$ .

#### Розв'язання:

```
#include <iostream>
#include <math.h>
using namespace std;

float A = 7.6;
float B = -8.9;
float C = 3.65;
float T[10] = {5.2, -7.1, 8.3, -3.4, 7.5, -8.4, 6.0, 5.2, 1.9, 4.3};
```

```
float SumT(int n)
{
    float s=0;
    for(int i=0; i<n; i++)
        s+=T[i];
    return s;
}

float Len(float a, float b)
{
    return sqrt(a*a+b*b);
}
```

```

void main()
{
    float result = Len(SumT(5), A)/Len(A,B) + Len(B,
SumT(7))/Len(SumT(6), C);
    cout << result << endl;
}

```

**Завдання для самостійної роботи** – див.стор. 118.



Національний університет  
водного господарства  
та природокористування

### Завдання 9.2

За умовою завдання виконати обчислення, які доцільно оформити у вигляді підпрограми, реалізувати їх засобами алгоритмічної мови.

#### Приклад 9.2

Користуючись формулами

$$x_c = \frac{\sum_{i=1}^n m_i x_i}{\sum_{i=1}^n m_i}, y_c = \frac{\sum_{i=1}^n m_i y_i}{\sum_{i=1}^n m_i}, z_c = \frac{\sum_{i=1}^n m_i z_i}{\sum_{i=1}^n m_i}, \text{ визначити і вивести на}$$

друк координати центра мас системи матеріальних точок

$M_i(x_i, y_i, z_i), i = \overline{1,6}$ , які задані масивом мас  $m(6) = (3; 6,3; 8,5; 5,2; 2; 1,5)$  і масивами координат  $X(6) = (2; 3; 4; 1; 0; 5)$ ,  $Y(6) = (3; -4; 2; 7; -2; 0)$ ,  $Z(6) = (4; 3; 0; 2; -1; 7)$ .

**Розв'язання:**

```

#include <iostream>
#include <math.h>
using namespace std;

```

```

const int n=6;

```

```

float M[n] = {3, 6.3, 8.5, 5.2, 2, 1.5};
float X[n] = {2, 3, 4, 1, 0, 5};
float Y[n] = {3, -4, 2, 7, -2, 0};
float Z[n] = {4, 3, 0, 2, -1, 7};

```

```

float SumDob(float * F1, float * F2)
{
    float s=0;

```

```

for(int i=0; i<n; i++)
    s+=F1[i]*F2[i];
return s;
}

```

```

float Sum(float * F)
{
    float s=0;
    for(int i=0; i<n; i++)
        s+=F[i];
    return s;
}

```

```

float CenterMas(float * C)
{
    return SumDob(M, C)/Sum(M);
}

```

```

void main()
{
    cout << "cX = " << CenterMas(X) << endl;
    cout << "cY = " << CenterMas(Y) << endl;
    cout << "cZ = " << CenterMas(Z) << endl;
}

```

**Завдання для самостійної роботи** – див. стор. 123.

### **Контрольні запитання:**

1. Яке призначення підпрограм?
2. Як класифікують підпрограми в C++ ?
3. Що розуміють під поняттями формальні та фактичні параметри?
4. Що таке глобальні та локальні змінні?

## **Лабораторна робота №10**

**Тема: Робота з текстовими змінними**

**Мета роботи:** Ознайомитися з основними функціями та стандартними процедурами обробки стрічок та навчитися застосовувати їх для написання програм.

### Хід роботи:

1. Розробити алгоритми обробки текстового масиву згідно варіантів завдань для самостійної роботи.
2. Написати відповідні програми на одній з мов мови програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.



Національний університет  
водного господарства  
та природокористування

### Завдання 10.1

#### Приклади 10.1

Дано текстовий масив A(10). Знайти і надрукувати елементи найменшої довжини. Вивести на друк даний елемент, його порядковий номер і довжину.

#### Розв'язання:

```
#include <iostream>
#include <string>
using namespace std;
```

```
void main()
{
    const int n = 10;
    string A[n];
    cout << "Введіть текстовий масив" << endl;
    for (int i=0; i<n; i++)
    {
        cout << "A[" << i << "]=";
        cin >> A[i];
    }

    int minLen = A[0].length();
    for (int i=1; i<n; i++)
    {
        int len = A[i].length();
        if (len < minLen)
            minLen = len;
    }

    cout << "Найменша довжина " << minLen << endl;
```

```

for (int i=0; i<n; i++)
{
    int len = A[i].length();
    if (len == minLen)
        cout << A[i] << endl;
}

```

**Завдання для самостійної роботи** – див.стор. 129.



Національний університет  
водного господарства  
та природокористування

## Завдання 10.2

Скласти програму обробки текстових величин згідно вказаної умови. Передбачити друк початкового та результуючого текстів.

### Приклад 10.2

Дано текстові величини: “дуб”, “кок”, “тет”, “піп”, “рід”.  
Надрукувати ті слова, які є симетричними.

### Розв’язання:

```

#include <iostream>
#include <string>
using namespace std;

bool IsSymetric(string s)
{
    int i=0;
    int j=s.length()-1;
    while(i<j)
    {
        if (s[i] != s[j])
            return false;
        i++;
        j--;
    }
    return true;
}

void main()
{
    const int n = 10;
}

```

```

string A[n];
cout << "Введіть текстовий масив" << endl;
for (int i=0; i<n; i++)
{
    cout << "A[" << i << "]=";
    cin >> A[i];
}

```



Національний  
водного господарства  
та природоохорони

```

cout << "Симетричні слова" << endl;
for (int i=0; i<n; i++)
{
    if (IsSymetric(A[i]))
        cout << A[i] << endl;
}
}

```

**Завдання для самостійної роботи** – див.стор. 131.

### Завдання 10.3

Виконати завдання 7.5. з використанням текстового масиву.



Національний університет  
водного господарства  
та природоохорони

#### **Контрольні запитання:**

1. Які функції відносяться до функцій перетворення типів ?
2. Як здійснюється опис стрічкового типу в C++ ?
3. Які дії визначено над стрічками.
4. Наведіть стандартні функції для роботи з стрічками.

## Лабораторна робота №11

**Тема: Використання вказівників та динамічних структур даних**

**Мета роботи:** Навчитися працювати з об'єктами динамічної природи: динамічними змінними, масивами; ознайомитися з основними процедурами для роботи з динамічними змінними та пам'яттю.

### Хід роботи:

1. Розробити алгоритм розв'язку поставленої задачі з використанням вказівників та динамічних структур даних.
2. Написати відповідну програму на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.



Національний університет  
водного господарства  
та природокористування

### Завдання 11.1

Виконати завдання по обробці одновимірних масивів, організувавши їх розміщення в пам'яті ПК динамічно. Значення елементів масиву ввести з клавіатури.

Варіанти завдань взяти з лабораторної роботи №7, завд. 7.2.

### Приклад 11.1

Знайти і надрукувати суму додатних елементів масиву.

### Розв'язання:

```
#include <iostream>
using namespace std;

void main()
{
    int n;
    cout << "Введіть кількість елементів масиву ";
    cin >> n;
    float *A = new float[n];
    for(int i=0; i<n; i++)
    {
        cout << "A[" << i << "]=";
        cin >> A[i];
    }

    float s = 0;

    for(int i=0; i<n; i++)
        if(A[i]>0)
            s+=A[i];
}
```



```

delete[] A;

cout << "S=" << s << endl;
}

```

### Завдання 11.2

Виконати завдання по обробці двовимірних масивів, організувавши їх розміщення в пам'яті ПК динамічно. Значення верхніх індексів та елементів масиву ввести з клавіатури. Варіанти завдань взяти з лабораторної роботи №8, завд. 8.1.



Національний університет  
та природокористування

### Приклад 11.2

Задана деяка матриця  $A(n, n)$ . Обчислити нові значення елементів матриці, які визначаються за формулою  $a_{ij}^* = \frac{a_{ij}}{S}$ , де  $S$ - сума елементів, які стоять на її головній діагоналі.

### Розв'язання:

```

#include <iostream>
using namespace std;

```

```

void main()
{
    int n;
    cout << "Введіть розмірність матриці ";
    cin >> n;

```

```


//Створення матриці
float **A = new float*[n];
for(int i=0; i<n; i++)
    A[i]=new float[n];

```

```


//Введення матриці
for(int i=0; i<n; i++)
    for(int j=0; j<n; j++)
    {
        cout << "A[" << i << ", " << j << "]=";
        cin >> A[i][j];
    }

```



//Обрахунок суми елементів головної діагоналі  
 float s = 0;  
 for(int i=0; i<n; i++)  
     s+=A[i][i];  
 //Обрахунок нових значень  
 for(int i=0; i<n; i++)  
     for(int j=0; j<n; j++)  
         A[i][j]/=s;  
 //Виведення елементів матриці  
 for(int i=0; i<n; i++)  
     for(int j=0; j<n; j++)  
         cout << "A[" << i << ", " << j << "]=" << A[i][j];  
 //Вивільнення пам'яті  
 for(int i=0; i<n; i++)  
     delete[] A[i];  
 delete[] A;  
 }

**Контрольні запитання:**

- 
1. Які масиви називають динамічними?
  2. Дайте означення статичного масиву.
  3. Як виділити пам'ять під динамічний двовимірний масив?
  4. Для чого використовуються оператори *new* та *delete*?
  5. Як повернути з функції кілька значень?

## РОЗДІЛ IV. Програмування на мовах С, С++, С# з використанням ООП

### Лабораторна робота №1

#### Тема: Використання класів в С++



Національний університет  
водного господарства  
та природокористування

**Мета роботи:** Ознайомитися з особливостями ООП та його основними властивостями, навчися створювати класи об'єктів.

#### Хід роботи:

5. Розробити алгоритми використання класів згідно варіантів завдань для самостійної роботи.
6. Написати відповідні програми на одній з мов програмування.
7. Виконати дане завдання на комп'ютері.
8. Захистити лабораторну роботу.

#### Завдання 1.1

Написати програму з використанням класів в С++. Розміщення об'єктів в пам'яті комп'ютера організувати у вигляді динамічних масивів.

#### Приклад 1.1

Написати програму в якій описано клас, який містить змінну-член (значення кута в градусах) і функції-члени, в яких обчислюються значення тригонометричних функцій даного кута.

#### Розв'язання:

```
#include <iostream.h>
#include <math.h>
#include <conio.h>
//Коефіцієнт переведення з градусів в радіани
const double DEG_TO_RAD=0.0174532925;
class degree //Користувацький клас
{
double data_value; //Змінна-член класу (значення кута в радіанах)
public:
void Set_value(double angle); //Метод доступу
```

```

double Get_sin (void); //Метод, який повертає синус кута
double Get_cos(void); //Метод, який повертає косинус кута
} deg; //Об'єкт класу

```

```

void degree::Set_value(double angle) //Визначення функції-члена класу
{
data_value=angle;
}
double degree::Get_sin(void)
{
return(sin(DEG_TO_RAD*data_value));
}
double degree::Get_cos(void)
{
return(cos(DEG_TO_RAD*data_value));
}
main ()
{
//Встановлюємо значення кута рівним 25 градусів
deg.Set_value(25.0);
cout<<"Sin= "<<deg.Get_sin()<<endl;
cout<<"Cos= "<<deg.Get_cos()<<endl;
getch();
}

```

### **Варіанти завдань для самостійної роботи:**

1. Дано масив об'єктів класу „квадратна матриця 3×3”. Серед  $n$  об'єктів визначити номер матриці з найбільшою сумою діагональних елементів (найбільшим слідом).
2. Дано масив об'єктів „трикутник на площині”, який заданий координатами своїх вершин. Серед  $n$  трикутників визначити той, в якого найбільша площа.
3. Дано масив об'єктів „опуклий чотирикутник на площині” (чотирикутник заданий координатами вершин). Визначити той чотирикутник, у якого найбільший периметр.
4. Дано масив об'єктів „круг на площині”, який заданий радіусом та координатами центру. Визначити той круг, у якого найбільша площа.
5. Дано масив об'єктів „тривимірний вектор”. Визначити той вектор, який має найбільшу довжину.
6. Дано масив об'єктів „масив з 5-ти елементів”. Визначити той масив, в якого найменший максимальний елемент.

7. Дано масив об'єктів „слово”. Визначити те слово, в якому найбільша кількість цифр.
8. Дано масив об'єктів „кубічний многочлен  $a_3x^3 + a_2x^2 + a_1x + a_0$ ”. Визначити той многочлен, який приймає найменше мінімальне значення на введеному користувачем відрізьку  $[a, b]$  із заданою точністю  $\mathcal{E}$  (використати метод простого перебору).
9. Дано масив об'єктів „площина  $Ax + By + Cz + D = 0$ ”. Визначити площину, яким належить введена користувачем точка.
10. Дано масив об'єктів „тетраедр”, який заданий координатами своїх вершин в просторі. Із заданого масиву об'єктів „тетраедр” визначити тетраедр з найбільшим об'ємом.
11. Дано клас „конус”, який заданий координатами центру основи, координатами вершини та радіусом основи. Із заданого масиву об'єктів вказаного класу визначити конус з найбільшою твірною.
12. Дано клас „людина”. Із заданого масиву людей визначити тих, які народилися в щасливі дні (ділення суми цифр числа, місяця, року народження на 7 дають три однакові остачі).
13. Дано масив об'єктів класу „пряма  $ax + by + c = 0$ ”. Визначити номери тих прямих, яким належить одна із двох введених користувачем точок.
14. Дано масив об'єктів класу „точка в просторі”. Визначити кількість точок, в яких всі координати цілочислові і які попадають в перший октант.
15. Дано масив об'єктів класу „тривимірний масив”. Визначити номер об'єкта, в якого найбільший мінімальний елемент.
16. Дано масив об'єктів класу „абонент”. Визначити того абонента, в якого найбільша сума цифр телефону.
17. Дано масив об'єктів класу „куля”. Куля задається радіусом та координатами центру. Визначити номери тих куль, в які попадає введена користувачем точка.
18. Дано масив об'єктів класу „числова послідовність” (містить  $\leq 5$  членів). Визначити ту послідовність, в якій найменше середнє квадратичне відхилення між послідовними членами.
19. Дано масив об'єктів класу „перетворення площини  $\begin{cases} x = a_1\xi + b_1\eta + c_1, \\ y = a_2\xi + b_2\eta + c_2; \end{cases}$ ”. Визначити номери тих перетворень, які переводять задані користувачем точки  $(x, y)$  та  $(\xi, \eta)$  одна в одну.
20. Дано масив об'єктів класу „геометрична прогресія”. Клас характеризується першим членом  $a_0$ , знаменником  $q$  та кількістю членів  $n$ . Визначити номер тієї прогресії, в якій буде найбільший останній член.

21. Дано масив об'єктів класу „сума  $\sum_{i=1}^3 \frac{a_i}{i(i+1)}$ ”. Визначити номер об'єкта з найбільшою сумою.
22. Дано масив об'єктів класу „матеріальна точка”. Кожна точка характеризується координатами початкового положення та вектором рівномірної швидкості  $\vec{v} = (v_1, v_2, v_3)$ . Визначити номера точок, які за введений користувачем час  $t$  попадають в перший октант.
23. Дано масив об'єктів класу „квадратне рівняння  $ax^2 + bx + c = 0$ ”. Визначити номери тих рівнянь, які не мають дійсних розв'язків.
24. Дано масив об'єктів класу „арифметична прогресія”. Клас характеризується першим членом  $a_0$ , різницею  $d$  та кількістю членів  $n$ . Визначити номер тієї прогресії, в якій буде найбільша сума.
25. Дано масив об'єктів класу „функція  $\sin(ax+b)$ ”. Визначити номер функції, яка приймає найбільше значення у введений користувачем точці  $x$ .

### Контрольні запитання:

1. Чим відрізняється поняття „клас” від поняття „об'єкт”?
2. Які ключові слова використовуються для описання класів в C++?
3. Чим відрізняються класи, оголошені за допомогою ключових слів struct, union і class?
4. Які специфікатори доступу використовуються в класах?
5. За допомогою якого ключового слова оголошуються друзі класу?
6. Чи є функція-друг класу методом даного класу?
7. Чи є конструктор класу методом даного класу?
8. Які основні призначення конструкторів та деструкторів класу?
9. Чи можна оголосити дружній весь клас?

## Лабораторна робота №2

**Тема: Наслідування в C++**

**Мета роботи:** Ознайомитися з особливостями ООП та його основними властивостями, навчися створювати класи об'єктів.

**Хід роботи:**

1. Розробити алгоритми створення класів згідно варіантів завдань для самостійної роботи.
2. Написати відповідні програми на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.

### Завдання 2.1

Написати програму, описавши батьківський та похідний класи і перевантажуючи відповідні методи.

#### Приклад 2.1

Описати клас комплексних чисел і на основі його створити клас дійсних чисел, в якому перевантажити відповідні методи.

#### Розв'язання:

```

#include <iostream.h>
#include <math.h>
#include <conio.h>
//Клас комплексних чисел a+ib
class Complex
{
public:
    Complex (); //Конструктор
    ~Complex (); //Деструктор
    //Методи доступу до даних
    float GetA() const;
    void SetA(float);
    float GetB() const;
    void SetB(float);
    //Метод пошуку модуля комплексного числа
    float Modul() const;
protected:
    float a;
    float b;
};
//Похідний від класу Complex клас Real - клас дійсних чисел
class Real:: public Complex
{
public:
    Real (); //Конструктор
    ~Real (); //Деструктор

```

```
//Інші методи  
float Modul () const; //Заміщення методу базового класу  
float Squared ();  
float Cubed ();  
};
```

```
float Complex::GetA() const  
{return(a);};
```

```
float Complex::GetB() const  
{return(b);};
```

```
void Complex::SetA( float myReal)  
{  
a=myReal;  
};
```

```
void Complex::SetB( float myReal)  
{  
b=myReal;  
};
```

```
float Complex::Modul() const
```

```
{  
return(pow(a*a+b*b, 1.0/2));  
};
```

```
//Конструктор класу Real
```

```
Real::Real ( )
```

```
{  
b=0.0; //Уявна частина дійсного числа дорівнює нулю  
};
```

```
float Real::Modul() const  
{  
return(fabs(a));  
};
```

```
float Real::Squared() const  
{  
return(a*a);  
};
```

```
float Real::Cubed() const
```



```

{
return(a*a*a);
};

main()
{
Real r; //Об'єкт класу Real
Complex c; //Об'єкт класу Complex
c.SetA(5);
c.SetB(10);
r.SetA(15); //Метод r.SetB не викликаємо, оскільки b=0 в //конструкторі
класу Real
cout<<c.Modul(); //Метод базового класу Complex
cout<<r.Modul(); //Заміщений метод похідного класу Real
getch ();
}

```

### Варіанти завдань для самостійної роботи:

- Створити клас „Опуклий чотирикутник”.  
Створити відповідні методи:
  - задання координат вершин;
  - виведення координат вершин на екран;
  - обчислення площі.
 Описати похідний від нього клас „трикутник” з відповідними перевантаженими методами. Створити об’єкти „опуклий чотирикутник”, „трикутник” та обчислити їх площі.
- Створити клас „матриця” розмірності  $4 \times 4$ .  
Створити відповідні методи:
  - задання елементів матриці;
  - виведення матриці на екран;
  - знаходження максимального елемента матриці.
 Описати похідний від нього клас „одновимірний вектор розмірності 4” з відповідними перевантаженими методами. Створити об’єкти класів „матриця” та „одновимірний вектор”. Знайти максимальні елементи кожного об’єкта.
- Описати клас „дробова функція” виду  $\frac{a_2x^2 + a_1x + a_0}{b_2x^2 + b_1x + b_0}$ .  
Створити відповідні методи:
  - задання коефіцієнтів чисельника та знаменника;
  - виведення коефіцієнтів на екран;

➤ знаходження значення в заданій точці  $x_0$ .

Створити похідний від нього клас „дробово-лінійна функція  $\frac{a_1x + a_0}{b_1x + b_0}$ ” з

відповідними перевантаженими методами. Створити об'єкти класів „дробова функція” та „дробово-лінійна функція” та обчислити їх значення у відповідній точці.

4. Описати клас „гіперплощина виду  $a_4x_4 + a_3x_3 + a_2x_2 + a_1x_1 + a_0 = 0$ ” (тут  $a_0, a_1, a_2, a_3, a_4$  – задані числа,  $X = (x_1, x_2, x_3, x_4)$  – точка 4-вимірного простору).

Створити відповідні методи:

- задання відповідних коефіцієнтів;
- виведення відповідних коефіцієнтів на екран;
- визначення, чи належить задана точка гіперплощині.

Створити похідний клас „пряма  $a_1x + a_2y + a_0 = 0$ ”. Перевантажити відповідні методи. Створити об'єкти класів „гіперплощина” та „пряма”. Визначити, чи належать введені користувачем точки створеним об'єктам.

5. Створити клас „еліпсоїд  $\frac{(x_1 - b_1)^2}{a_1^2} + \frac{(x_2 - b_2)^2}{a_2^2} + \frac{(x_3 - b_3)^2}{a_3^2} = 1$ ”, де  $a_i, b_i, i = \overline{1,3}$

– задані числа.

Описати методи:

- задання відповідних коефіцієнтів;
- виведення відповідних коефіцієнтів на екран;
- визначення об'єму еліпсоїда.

Створити похідний клас „куля  $(x_1 - b_1)^2 + (x_2 - b_2)^2 + (x_3 - b_3)^2 = R^2$ ”.

Перевантажити відповідні методи. Створити об'єкти класів „еліпсоїд” та „куля”. Визначити об'єми кулі та еліпсоїда.

6. Створити клас „тривимірна матриця  $A[3][3][3]$ ”.

Створити методи:

- задання елементів матриці, використовуючи введення з клавіатури;
- задання елементів матриці використовуючи випадкові числа;
- знаходження мінімального елемента матриці.

Створити похідний клас „двовимірна матриця  $A[3][3]$ ”. Перевантажити відповідні методи. Створити об'єкти класів „тривимірна матриця” та „двовимірна матриця”. Визначити мінімальні елементи всіх матриць.

7. Створити клас „півпростір  $a_1x_1 + a_2x_2 + a_3x_3 \leq b$ ”, де  $a_1, a_2, a_3, b$  – задані числа,  $X = (x_1, x_2, x_3)$  – точка простору.

Створити відповідні методи:

- задання відповідних коефіцієнтів;
- виведення відповідних коефіцієнтів на екран;
- визначення, чи належить введена користувачем точка  $X = (x_1, x_2, x_3)$  даному півпростору.

Створити похідний клас „півплощина  $a_1x_1 + a_2x_2 \leq b$ ”. Перевантажити відповідні методи. Створити об’єкти класів „півпростір” та „півплощина”. Визначити, чи належать введені користувачем точки півпростору та півплощині.

8. Створити клас „паралелепіпед  $b_1 \leq x_1 \leq a_1, b_2 \leq x_2 \leq a_2, b_3 \leq x_3 \leq a_3$ ”, де  $a_i, b_i, i = \overline{1,3}$  – задані числа.

Описати методи:

- задання відповідних коефіцієнтів;
- виведення відповідних коефіцієнтів на екран;
- визначення чи належить введена користувачем точка заданому паралелепіпеду.

Створити похідний від нього клас „прямокутник  $b_1 \leq x_1 \leq a_1, b_2 \leq x_2 \leq a_2$ ”. Перевантажити відповідні методи. Створити об’єкти класів „паралелепіпед” та „прямокутник”. Визначити, чи належать введені користувачем точки вказаним об’єктам.

9. Створити клас „система лінійних алгебричних рівнянь

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3. \end{cases}$$

Створити відповідні методи:

- задання коефіцієнтів рівнянь та вільних членів;
- виведення СЛАР на екран;
- визначення, чи задовольняє введений користувачем вектор  $X = (x_1, x_2, x_3)$  даній системі рівнянь.

Створити похідний від нього клас „система рівнянь  $\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1, \\ a_{21}x_1 + a_{22}x_2 = b_2. \end{cases}$ ”.

Перевантажити відповідні методи. Створити об’єкти класів „СЛАР  $3 \times 3$ ”, „СЛАР  $2 \times 2$ ”. Визначити, чи задовольняють введені користувачем вектори створеним об’єктам «СЛАР  $3 \times 3$ » та «СЛАР  $2 \times 2$ ».

10. Описати клас „тривимірний підхідний дріб виду  $\frac{1}{a_1x + \frac{1}{a_2x + \frac{1}{a_3x}}}$ ”, де  $a_i$ ,

$i = \overline{1,3}$  – задані числа, причому  $a_3 \neq 0$ .

Створити відповідні методи:

- задання відповідних коефіцієнтів;
- виведення відповідних коефіцієнтів на екран;
- обчислення значення підхідного дробу у введений користувачем точці  $x$ .

Створити похідний від нього клас „дріб  $\frac{1}{ax}$ ”. Перевантажити відповідні методи. Створити об’єкти класів „тривимірний похідний дріб” та „дріб”. Обчислити значення дробів у введених користувачем точках.

11. Описати клас „тетраedr”, який визначається координатами чотирьох точок у просторі.

Створити відповідні методи:

- задання координат вершин тетраедра;
- виведення відповідних коефіцієнтів на екран;
- обчислення об’єму тетраедра.

Створити похідний від нього клас „трикутник” і перевантажити відповідні методи. Створити об’єкти класів „тетраedr” та „трикутник”. Обчислити площу трикутника та об’єм тетраедра.

12. Описати клас „многочлен  $a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$ ”, який визначається коефіцієнтами  $a_i, i = \overline{0,4}$ .

Створити відповідні методи:

- задання коефіцієнтів многочлена;
- виведення відповідних коефіцієнтів на екран;
- обчислення значення многочлена в точці  $x$ .

Створити похідний клас „лінійна функція  $a_1x + a_0$ ” з відповідними перевантаженими методами. Створити об’єкти класів „многочлен” та „лінійна функція”. Обчислити значення функцій у введений користувачем точці.

13. Дано клас „трикутник”, який визначається довжиною однієї із сторін та значеннями прилеглих двох кутів. Створити відповідні методи:

- задання значення довжини сторони та двох кутів;
- знаходження інших характеристик трикутника: величини кутів та довжин сторін;
- обчислення периметра.

Створити похідний клас „рівносторонній трикутник”. Створити об’єкти класів „трикутник” та „рівносторонній трикутник”. Знайти інші характеристики створених трикутників та їх периметри.

14. Описати клас „циліндрична система координат” та створити відповідні методи:

- перетворення координат заданої точки з декартової системи координат в циліндричну;
- перетворення координат заданої точки з циліндричної системи в декартову.

Створити похідний клас „полярна система координат” і перевантажити відповідні методи. Створити об’єкти класів „циліндрична система координат” та „полярна система координат”. Перевести введені користувачем точки з циліндричної системи в декартову та з декартової в полярну систему.

15. Описати клас „перетворення простору  $\begin{cases} x' = a_{11}x + a_{12}y + a_{13}z + a_{14}; \\ y' = a_{21}x + a_{22}y + a_{23}z + a_{24}; \\ z' = a_{31}x + a_{32}y + a_{33}z + a_{34}. \end{cases}$ ” та

створити відповідні методи:

- задання коефіцієнтів перетворення;
- виведення коефіцієнтів перетворення на екран;
- визначення образу заданої точки  $(x, y, z)$ .

Описати похідний клас „перетворення площини  $\begin{cases} x' = a_{11}x + a_{12}y + a_{13} \\ y' = a_{21}x + a_{22}y + a_{23} \end{cases}$ ” з відповідними перевантаженими методами. Створити об’єкт класу „перетворення площини” і знайти образ введеної користувачем точки.

16. Описати клас „крива другого порядку  $a_{11}x^2 + 2a_{12}xy + a_{22}y^2 + b_1x + b_2y + c = 0$ ”.

- задання коефіцієнтів;
- виведення коефіцієнтів на екран;
- визначення чи належить задана точка  $(x, y)$  даній кривій другого порядку.

Описати похідний клас „еліпс  $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ ” з відповідними перевантаженими методами. Створити об’єкт класу „еліпс” і визначити чи належить введена користувачем точка  $(x, y)$  даному еліпсу.

17. Створити клас „система 3-х векторів  $A = (a_1, a_2, a_3)$ ,  $B = (b_1, b_2, b_3)$ ,  $C = (c_1, c_2, c_3)$ ” і описати відповідні методи:

- задання координат векторів;
- виведення координат вектора на екран;

➤ визначення, чи система векторів  $A, B, C$  є лінійно незалежною.

Описати похідний клас „система двох векторів  $A = (a_1, a_2)$ ,  $B = (b_1, b_2)$ ” з відповідними перевантаженими методами. Створити об’єкти класів „система 3-х векторів” та „система 2-х векторів”. Визначити чи система даних векторів не є лінійно залежними.

18. Описати клас „людина”, який містить ім’я, прізвище, по-батькові, число, місяць та рік народження та описати відповідні методи:

➤ задання відповідних даних;

➤ визначення за поточною введеною датою віку людини;

➤ обчислення кількості зустрічань певної літери (літера вводиться користувачем) в прізвищі людини.

Описати похідний клас „студент” з відповідними перевантаженими методами, який також містить додатково рік вступу до ВУЗу. Створити об’єкти класів „людина” та „студент”. За поточною введеною датою визначити вік студента. Визначити кількість зустрічань введеної користувачем літери в прізвищі людини.

19. Дано клас „трикутник”, який визначається довжинами двох сторін та значенням кута між ними. Створити відповідні методи:

➤ задання значення довжини сторони та двох кутів;

➤ знаходження інших характеристик трикутника: величини кутів та довжин сторін;

➤ обчислення периметра.

Створити похідний клас „прямокутний трикутник”. Створити об’єкти класів „трикутник” та „прямокутний трикутник”. Знайти інші характеристики створених трикутників та їх периметри.

20. Описати клас „система трьох лінійних нерівностей

$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \leq b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \leq b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \leq b_3. \end{cases}$ ”, створити відповідні методи:

➤ задання коефіцієнтів відповідних нерівностей;

➤ виведення коефіцієнтів на екран;

➤ визначення, чи задовольняє введений користувачем вектор  $X = (x_1, x_2, x_3)$  даній системі нерівностей.

Створити похідний клас „система двох лінійних нерівностей

$\begin{cases} a_{11}x_1 + a_{12}x_2 \leq b_1, \\ a_{21}x_1 + a_{22}x_2 \leq b_2. \end{cases}$ ” і перевантажити відповідні методи.

Створити об’єкт класу „система лінійних нерівностей” і визначити чи введена користувачем точка задовольняє даній системі.

21. Описати клас „працівник фірми”, який містить прізвище та ім'я працівника; дату прийому на роботу в фірму; назву навчального закладу, який закінчив; посада. Описати відповідні методи:

- задання вказаних даних;
- визначення стажу роботи на фірмі;
- визначення чи прізвище є симетричним.

Створити похідний клас „практикант” з відповідними перевантаженими методами, який також містить назву ВУЗу, в якому вчиться практикант. Створити об'єкти класів „працівник фірми” та „практикант”. Для працівника визначити стаж його роботи на фірмі. Для практиканта визначити чи його прізвище є симетричним.

22. Описати клас „кубічне рівняння  $a_3x^3 + a_2x^2 + a_1x + a_0 = 0$ ”, який містить:

- методи задання коефіцієнтів  $a_3, a_2, a_1, a_0$  та метод виведення їх на екран;
- метод визначення, чи задовільняє введене користувачем число  $x$  даному рівнянню.

Створити похідний клас „квадратне рівняння  $b_2x^2 + b_1x + b_0 = 0$ ”.

Перевантажити відповідні методи. Поповнити даний клас методом пошуку коренів квадратного рівняння.

Створити об'єкт класів „кубічне рівняння” та „квадратне рівняння”. Знайти корені квадратного рівняння. Визначити, чи задовільняє введене користувачем число  $x$  кубічному рівнянню.

23. Створити клас „сфера  $(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = R^2$ ” і створити відповідні методи:

- задання координат  $(x_0, y_0, z_0)$  центру сфери та її радіусу  $R$ , а також виведення цих даних на екран;
- визначення площі поверхні сфери.

Створити похідний клас „коло  $(x - x_0)^2 + (y - y_0)^2 = R^2$ ”. Перевантажити відповідні методи. Створити об'єкти класів „сфера  $(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = R^2$ ” та „коло  $(x - x_0)^2 + (y - y_0)^2 = R^2$ ”. Визначити площу поверхні сфери та довжину кола.

24. Описати клас „рухома матеріальна точка  $(x, y, z)$ ”, координати якої визначаються як  $x = x_0 + a_1 \sin(t)$ ,  $y = y_0 + a_2 \cos(t)$ ,  $z = z_0 + a_3 t^2$ .

Створити відповідні методи:

- задання початкового положення точки  $(x_0, y_0, z_0)$ ;
- задання коефіцієнтів  $a_3, a_2, a_1$  та виведення їх на екран;
- визначення координат точки в заданий момент часу  $t$ .

Створити похідний клас „рухома матеріальна точка  $x$ ”, яка рухається по прямій і координата її визначається як  $x = x_0 + a_1 \sin(t), y = 0, z = 0$ . Перевантажити відповідні методи. Створити об’єкти класів „рухома матеріальна точка  $(x, y, z)$ ” та „рухома матеріальна точка  $x$ ” і визначити їх положення у введений користувачем момент часу  $t$ .

25. Задано клас „конус”, який характеризується координатами вершини, координатами центру основи та твірною. Описати метод:

- задання вказаних даних та виведення їх на екран;
- метод визначення бічної поверхні конуса;
- метод визначення радіуса основи конуса.

Описати похідний клас „коло в просторі”, який буде характеризуватись центром та радіусом. Перевантажити відповідні методи. створити об’єкти класів „конус” та ”коло в просторі”. Визначити радіус основи конуса. Визначити довжину кола.

### **Контрольні запитання:**

1. Що розуміється під „наслідуванням”?
2. Чи допускають мови програмування C++ (C#) множене наслідування?
3. Чи можливе заміщення конструктора батьківського класу в похідному класі?
4. Чи можливий виклик деструктора батьківського класу в методах похідного класу?
5. Основне призначення наслідування?
6. Чи можливе заміщення методів батьківського класу в похідному класі?
7. Як викликати метод батьківського класу в похідному класі, якщо в похідному класі також існує метод з таким-же іменем?
8. Чи можна створити об’єкт похідного класу, якщо не створено жодного об’єкта батьківського класу?
9. Побудуйте ієрархію класів для поняття „число”.
10. Побудуйте ієрархію класів для поняття „ромб”.



## Лабораторна робота №3

### Тема: Масиви в C#

**Мета роботи:** Ознайомитися з особливостями роботи одновимірних та двовимірних масивів в C#, навчитися виконувати обробку одновимірних масивів, здійснювати сортуванням масивів.

#### Хід роботи:

1. Розробити алгоритми обробки одновимірних та двовимірних масивів згідно варіантів завдань для самостійної роботи.
2. Написати відповідні програми на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.

#### Завдання 3.1

Написати програму по обробці одновимірних масивів. Кількість та значення елементів масиву ввести з клавіатури. Використати методи та властивості абстрактного класу System.Array.

#### Варіанти завдань для самостійної роботи:

1. Обчислити значення многочлена  $a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ , використавши масив коефіцієнтів  $a(n)$ .
2. Знайти максимальне із чисел, які зустрічаються в цілочисловому масиві  $a(n)$  більше одного разу.
3. Дано масив  $a(n)$ . Визначити скільки пар (додатне число, від'ємне число) знаходиться на початку масиву.
4. Знайти найменше серед тих елементів масиву  $a(n)$ , які не є елементами масиву  $b(m)$ .
5. Координати  $n$  точок на площині задані у вигляді двох масивів  $x(n)$  та  $y(n)$ . Вибрати із заданих  $n$  точок такі три, з яких можна побудувати трикутник найбільшої площі.
6. В масиві  $a(n)$  визначити кількість підпоследовностей із непарної кількості підряд ідучих нулів.
7. В масиві  $a(n)$  знайти підпоследовність максимальної довжини, в якій перше число дорівнює останньому, друге – передостанньому і т. д.

8. В масиві  $a(n)$  елементами є цифри „0” – „9”. Визначити всі різні трійки елементів масиву  $a(n)$ . Наприклад, в масиві 318731873 різні трійки – це 318, 187, 873, 731.
9. Підрахувати, скільки різних чисел міститься в заданому цілочисловому масиві  $a(n)$ .
10. Дано масив  $b(n)$ . Визначити три найбільші елементи цього масиву.
11. Координати  $n$  точок на площині задані у вигляді двох масивів  $x(n)$  та  $y(n)$ . Знайти таку точку із заданих, що круг радіуса  $R$  з центром в цій точці містить мінімальну кількість точок із даної множини точок. Радіус  $R$  вводиться користувачем.
12. Випуклий багатокутник задано своїми вершинами, координати яких задані в двох масивах  $x(n)$  та  $y(n)$ . Вершини впорядковані. Визначити, чи знаходиться точка  $A(x_A, y_A)$  всередині багатокутника.
13. Вказати індекси та надрукувати ті елементи цілочислового масиву  $x(n)$ , сума цифр яких дорівнює заданому числу  $M$ .
14. Дано цілочисловий масив  $a(n)$ . Користувачем вводиться цифра  $k$ . Вказати індекси та надрукувати ті елементи масиву  $a(n)$ , в записі яких зустрічається цифра  $k$ .
15. Серед елементів масиву  $b(n)$  знайти той, який зустрічається найчастіше і надрукувати його та кількість зустрічань.
16. Дано цілочисловий масив  $a(n)$ . Знайти в ньому два найменших елемента.
17. Дано масив  $x(n)$ . Знайти серед його елементів два, різниця яких має найбільше значення.
18. Дано цілочисловий масив  $a(n)$ . Відсортувати його таким чином, щоб всі додатні числа знаходились на початку, а від’ємні – в кінці масиву і був збережений початковий порядок слідування елементів в обох групах.
19. Дано цілочисловий масив  $a(n)$ . Визначити ту цифру, з якої найчастіше розпочинаються елементи масиву.
20. Дано цілочисловий масив  $b(n)$ . Знайти його медіану. Медіаною для  $n$  елементів називається елемент, менший (або рівний) половині з  $n$  елементів та більший (або рівний) іншій половині із  $n$  елементів.
21. Дано цілочисловий масив  $b(m)$ . Знайти мінімальне натуральне число, яке не можна подати сумою жодних двох елементів масиву  $b(m)$  (кожен елемент масиву може входити в суму лише один раз).
22. Елементи кожного з масивів  $x(n)$  та  $y(m)$  впорядковані по спаданню. Об’єднати елементи цих двох масивів в один масив  $z(n+m)$  так, щоб він знову виявився впорядкованим по спаданню.
23. Дано масив чисел  $x(n)$ . Знайти суму елементів даного масиву, які знаходяться між максимальним та мінімальним елементами, включаючи в суму обидва ці числа. Врахувати, що в масиві може бути кілька найбільших та найменших елементів.

24. Дано масив дійсних чисел  $a(n)$ . Знайти коефіцієнти многочлена  $(x - a_1)(x - a_2)(x - a_3) \dots (x - a_n)$ .
25. Дано два масиви  $a(n)$  та  $b(m)$ . Знайти найбільше серед тих чисел масиву  $a(n)$ , які не входять в другий масив  $b(m)$ .

### Завдання 3.2



Національний університет  
водного та природокористування

Написати програму по обробці двовимірних масивів. Кількість рядків та стовпців, а також значення елементів двовимірних масивів ввести з клавіатури.

1. Переформувати матрицю  $a(n, m)$  таким чином, щоб її рядки розташувались в порядку зростання їх поелементних сум.
2. Для заданої матриці  $a(n, n)$  знайти мінімум серед сум модулів елементів діагоналей, які паралельні побічній діагоналі.
3. Дано цілочислова матриця  $a(n, m)$ . Замінити нулями елементи матриці, які стоять на перетині рядків та стовпців, в яких є хоча б по одному одиничному елементу.
4. Переформувати матрицю  $a(n, m)$  таким чином, щоб її стовпці розташувались по спаданню їх поелементних сум.
5. Сітка авіаліній, яка з'єднує міста, задана матрицею зв'язності  $M(n, n)$ , де  $m_{ij} = 0$ , якщо міста з номерами  $i, j$  не пов'язані між собою напряму і  $m_{ij} = 1$ , якщо міста зв'язані. Надрукувати всі пари міст, які поєднані між собою не на пряму, але не більше ніж з однією пересадкою.
6. Переформувати матрицю  $a(n, m)$  таким чином, щоб її рядки розміщувалися по зростанню їх перших елементів.
7. Дано матриця  $a(k, k)$ , де  $a_{ij}$  – час на дорогу від населеного пункту  $i$  в населений пункт  $j$ . Знайти номер населеного пункту  $i$ , в якому найвигідніше розмістити лікарню (сумарний час поїздок із  $i$  в усі інші населені пункти має бути мінімальним).
8. Для заданої матриці  $b(n, n)$  знайти мінімум серед сум модулів елементів діагоналей, які паралельні головній діагоналі.
9. Знайти сідлові точки матриці  $a(n, m)$  (сідловою точкою називається елемент, який є мінімальним в рядку і максимальним в стовпці).
10. Переформувати матрицю  $a(n, m)$  таким чином, щоб її стовпці розміщувались по зростанню їх останніх елементів.
11. В матриці  $x(n, m)$  всі члени різні. В кожному рядку знайти мінімальний елемент і серед знайдених елементів знайти максимальний. Надрукувати індекси цього елемента.

12. Дано матриця  $a(n,m)$ . Визначити кількість особливих елементів в ній, рахуючи елемент особливим, якщо:

а) він більший суми інших елементів свого стовпця;

б) в його рядку зліва від нього знаходяться елементи менші його, а справа – більші.

13. Сусідами елемента  $a_{ij}$  в матриці  $a(n,m)$  назвемо елементи  $a_{kp}$ ,  $i-1 \leq k \leq i+1$ ,  $j-1 \leq p \leq j+1$ ,  $(k,p) \neq (i,j)$ . Елемент матриці назвемо локальним мінімумом, якщо він строго менший всіх своїх сусідів. Підрахувати кількість локальних мінімумів заданої матриці та надрукувати їх індекси.

14. Дано матриця  $b(n,m)$ . Надрукувати індекси верхнього лівого кута матриці розміром  $2 \times 2$ , яка має найбільший визначник, і значення визначника.

15. Дано матриця  $a(n,m)$ . Побудувати масив  $b(n)$ , де  $b_k = 1$  якщо  $k$ -й рядок матриці  $A$  симетричний (тобто, перший елемент дорівнює останньому, другий – передостанньому і т. д.), і  $b_k = 0$  – в протилежному випадку.

16. Дано цілочислова матриця  $a(n,m)$ . Визначити всі „досконалі” елементи матриці та їх індекси. Елемент  $a_{ij}$  „досконалий”, якщо:

а) він парний і стоїть на перетині стовпця та рядка з парними індексами;

б) він непарний і стоїть на перетині стовпця та рядка з непарними індексами;

в) він стоїть на перетині парного та непарного стовпця або рядка, і  $a_{ij} = i + j$ .

17. Є  $n$  населених пунктів і сітка залізничних колій, які поєднують ці міста. Сітка задана матрицею зв'язності  $m(n,n)$ , де  $m_{ij} = 0$ , якщо міста  $i$  та  $j$  не поєднані між собою і  $m_{ij} = 1$  – в протилежному випадку. Вияснити чи є серед  $n$  населених пунктів ізольовані, тобто такі, в які неможливо доїхати. Надрукувати номери ізольованих населених пунктів.

18. Визначити чи є задана матриця  $a(n,m)$  ортонормованою, тобто такою, в якій скалярний добуток кожної пари різних рядків дорівнює нулю, а скалярний добуток кожного стовпця на себе дорівнює одиниці.

19. Дано цілочислова матриця  $a(n,m)$ . Сформувати матрицю  $b(l,2)$ , де в першому рядку матриці  $b$  записані різні елементи матриці  $a$ , а в другому рядку – кількість зустрічань цих елементів в  $a$ . Наприклад:

$$a = \begin{pmatrix} 1 & 3 & 5 \\ 6 & 5 & 2 \\ 4 & 5 & 4 \end{pmatrix}, b = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 1 & 1 & 2 & 3 & 1 \end{pmatrix}.$$

20. Дано матриця слів  $a(n,m)$ . Сформувати матрицю  $b(n,m)$ , де  $b_{ij}$  – кількість зустрічань пар  $(c_{ij}, d_{ij})$  в усій матриці  $a(n,m)$ , де  $c_{ij}$  – перша літера, а  $d_{ij}$  – остання літера слова  $a_{ij}$ . Наприклад, для

$$a = \begin{pmatrix} com & lom & cordon \\ ar & win & lum \\ worn & world & lm \end{pmatrix} \text{ відповідає } b = \begin{pmatrix} 2 & 3 & 2 \\ 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}.$$

21. Дано дві матриці  $a(n,m)$  та  $b(k,l)$ , де  $k \leq n, l \leq m$ . Визначити чи входить матриця  $b$  в матрицю  $a$  як підматриця і якщо входить, то вивести індекси верхнього лівого кута.
22. Дано цілочислова матриця  $a(n,m)$ . Впорядкувати стовпці матриці в порядку спадання перших непарних елементів в стовпці. Якщо непарних елементів в стовпці немає, то всі його елементи обнулити і перенести на початок.
23. Дано цілочислова матриця  $a(n,m)$ . Впорядкувати рядки матриці в порядку спадання їх середніх арифметичних значень без урахування мінімального та максимального елемента рядка.
24. Визначити, чи є матриця  $a(n,n)$  оберненим магічним квадратом, тобто такою в якій не рівні жодна із сум елементів в усіх рядках та стовпцях.
25. Визначити, чи є задана квадратна матриця  $a(n,n)$  магічним квадратом, тобто такою, в якій суми елементів в усіх рядках та стовпцях однакові.

### Контрольні запитання:

1. Яка структура програм в мові програмування C#?
2. Що являє собою простір імен?
3. Що містить простір імен System?
4. Яке ключове слово використовується для оголошення простору імен?
5. Чим відрізняється тип даних в C# та відповідний тип даних в CLR (Common Language Runtime)?
6. Від якого абстрактного класу є похідним тип даних „масив” в C#?
7. Наведіть основні властивості та методи абстрактного класу System.Array.
8. Який оператор циклу в C# дозволяє перебирати всі значення в масиві?

## Лабораторна робота №4

### Тема: Перевантаження функцій

**Мета роботи:** Ознайомитися з особливостями ООП, навчитися виконувати перевантаження функцій.

#### Хід роботи:

1. Розробити алгоритми перевантаження згідно варіантів завдань для самостійної роботи.
2. Написати відповідні програми на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.

#### Завдання 4.1

Написати програму, використавши перевантажені функції, які відрізняються або кількістю параметрів або типом параметрів та типом значення, яке повертають.

#### Приклад 4.1

Нище наведено приклад, в якому використовується перевантаження функції `Double`. Функція отримує числовий аргумент одного з типів `int`, `long` або `float`, і в залежності від типу вхідних даних повертає число, яке в два, три або чотири рази більше від початкового.

#### Розв'язання:

```
#include <iostream.h>
#include <conio.h>
int Double (int); //Перевантажена функція
long Double (long);
float Double (float);

main()
{
int myInt=6500;
long myLong=65000;
float myFloat=6.5;

//Використання перевантаженої функції
cout<<"doubledInt="<<Double(myInt)<<endl;
cout<<"doubledLong="<<Double(myLong)<<endl;
```

```
cout<<"doubledFloat="<<Double(myFloat)<<endl;
getch();
};
```

```
int Double(int original)
{return(2*original);};
```

```
long Double(long original)
{return(3*original);};
```

```
float Double(float original)
{return(4*original);};
```



водного господарства  
та природокористування

### *Варіанти завдань для самостійної роботи:*

1. Використати перевантажену функцію для обчислення квадрата числа, введеного в десятковій та двійковій системі числення.
2. Обчислити суму  $n$  перших членів арифметичної прогресії та суму нескінченно спадної геометричної прогресії з  $q < 1$ .
3. Визначити суму всіх цифр, які зустрічаються у введеному користувачем слові, та суму цифр натурального числа.
4. Визначити перевантажену функцію, яка повертає площу трикутника або площу опуклого чотирикутника, залежно від кількості введених координат вершин фігури.
5. Описати перевантажену функцію в якій, залежно від кількості переданих їй аргументів, вводиться одновимірний або двовимірний масив і визначається сума елементів даного масиву.
6. Визначити перевантажену функцію, яка обчислює або площу круга, який заданий координатами центра та радіусом, або периметр трикутника, який заданий координатами трьох своїх вершин.
7. Описати перевантажену функцію, яка визначає чи належить введена точка  $(x, y, z)$  площині  $Ax + By + Cz + D = 0$  або точка  $(x, y)$  прямій  $ax + by + c = 0$ .
8. Описати перевантажену функцію, яка визначає кількість цифр „1” в дійсному або цілому числі.
9. Визначити перевантажену функцію, яка повертає цілу частину та остачу від ділення двох натуральних чисел або повертає дробові частини двох дійсних чисел (залежно від типу її аргументів).
10. Описати перевантажену функцію, яка повертає значення тригонометричних функцій  $\sin, \cos, tg$ , коли аргумент вводиться в радіанах або градусах (ціле число).

11. Описати перевантажену функцію, яка переводить числа із шістнадцяткової системи в десяткову і навпаки: якщо число введене в шістнадцятковій системі, то переводимо в десяткову, якщо в десятковій – то в шістнадцяткову.
12. Описати перевантажену функцію *Reverse*, яка реверсує натуральні числа та слова (наприклад: 1253 → 3521).
13. Описати перевантажену функцію, яка має два аргументи:
  - 1) якщо аргументи цілі числа, то повертає їх НСД;
  - 2) якщо один із аргументів дійсне число, то повертає кількість чисел після коми даного числа.
14. Описати перевантажену функцію, яка містить два параметри, кожен з яких може бути дійсним або цілим числом:
  - 1) обидва параметри цілі числа – обчислюється найбільше з чисел;
  - 2) один із параметрів дійсне число, а інший ціле – дробова частина дійсного числа записується у вигляді цілого числа і обчислюється найбільше серед даного цілого числа та цілого числа-параметра;
  - 3) обидва параметри дійсні числа – обчислюється остача від ділення дробових частин обох чисел. При цьому дробові частини записуються у вигляді натуральних чисел (наприклад, для дійсного числа  $5,76$  дробова частина у вигляді натурального числа буде  $76$ ).
15. Описати перевантажену функцію, яка містить два параметри і залежно від типів параметрів повертає наступні значення:
  - 1) обидва параметри є словами – повертає модуль різниці довжини слів;
  - 2) обидва параметри є числами – повертає найбільше число;
  - 3) один із параметрів слово, а інший число – повертає частку від ділення числа на довжину слова.
16. Описати перевантажену функцію, яка містить три параметри і залежно від типів параметрів повертає наступні значення:
  - 1) всі параметри є цілими числами – повертає середнє арифметичне;
  - 2) один з параметрів – слово, а інші цілі числа – визначає чи не входить одне із чисел, як підслово в дане слово.
17. Визначити перевантажену функцію, яка повертає НСД та НСК двох або трьох цілих чисел.
18. Визначити перевантажену функцію, яка повертає кількість спільних підслів (із трьох літер) двох або трьох слів.
19. Описати перевантажену функцію, яка визначає, чи не утворюють цифри натурального числа або цифри, що послідовно зустрічаються у слові, арифметичну прогресію.





20. Визначити перевантажену функцію, яка має один аргумент і залежно від його типу повертає:
- 1) якщо аргумент натуральне число – то розклад даного числа на прості множники;
  - 2) аргумент – слово – то різницю першої та останньої цифри в цьому слові. Якщо цифр в слові немає – то довжину слова.
21. Визначити перевантажену функцію, яка залежно від типів аргументів та їх кількості повертає:
- 1) два аргументи, які є числами – повертає найбільше з них;
  - 2) один аргумент – це слово, в якому записана послідовність цих чисел, які розділені комами і завершуються крапкою. Повертає найбільше число з послідовності.
22. Два числа можуть задаватись безпосередньо або у вигляді слів вигляду „12+135+148.”, які завершуються крапкою. Описати перевантажену функцію, яка шукає добуток двох чисел, які можуть задаватися у вищеописаному вигляді.
23. Описати перевантажену функцію, яка замінює в числі всі нулі одиничками. Число може задаватися або в звичайному вигляді, або у вигляді слова.
24. Описати перевантажену функцію, яка повертає цифру, яка найчастіше зустрічається одночасно в двох або трьох натуральних числах.
25. Записати перевантажену функцію, яка переводить комплексне число із звичайного запису в тригонометричний і навпаки.



Національний університет  
водного господарства та природоохорони

Національний університет  
водного господарства та природоохорони

### **Контрольні запитання:**

1. Для чого використовується перевантаження функцій?
2. Що розуміють під терміном „поліморфізм функцій”?
3. Як повинні відрізнятися перевантажені функції?
4. Чи допускається створення перевантажених функцій з однаковими іменами, однаковим списком параметрів, але різними типами значень, які повертаються функціями?
5. Чим відрізняється механізм перевантаження функцій в C++ та C#?

## **Лабораторна робота №5**

**Тема: Використання зсилок в C-подібних мовах програмування**

**Мета роботи:** Навчитися використовувати зсилки в C-подібних мовах.

### Хід роботи:

1. Розробити алгоритми використання зсилок в С-подібних мовах перевантаження згідно варіантів завдань для самостійної роботи.
2. Написати відповідні програми на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.



Національний університет  
водного господарства  
та природокористування

### Завдання 5.1

Написати програму згідно нищенаведених варіантів. Для повернення з функції двох і більше значень використати зсилки.

#### Приклад 5.1

Написати програму, в якій функція повертає квадрат та куб числа, введеного з клавіатури, за допомогою зсилок. Якщо введене число виходить за межі відрізка  $[0,20]$ , то на екран виводиться повідомлення про помилку.

#### Розв'язання:

```
#include<conio.h>
#include<iostream.h>
typedef unsigned short USHORT //Створення псевдоніму типу
enum ERR_CODE {SUCCESS, ERROR} //Перелічуваний тип даних
ERR_CODE factor(USHORT, USHORT &, USHORT &);
main()
{
    USHORT number, squared, cubed;
    ERR_CODE result;
    cout<<"Enter number (0-20):";
    cin>>number;
    result=factor(number, squared, cubed);
    if (result==SUCCESS)
    {
        cout<<"number="<<number<<endl;
        cout<<"square="<<square<<endl;
        cout<<"cubed="<<cubed<<endl;
    }
    else cout<<"Error \n";
    getch();
}
```

**ERR\_CODE factor(USHORT n, USHORT &rSquared, USHORT &rCubed)**

*//Функція, яка повертає квадрат та куб числа n за допомогою зсилок*

```
{  
if (n>20) return(ERROR);  
else  
    {  
        rSquared=n*n;  
        rCubed=n*n*n;  
        return(SUCCESS);  
    }  
}
```



На  
водоохоронного  
та

### **Варіанти завдань для самостійної роботи:**

1. Дано клас „тривимірний вектор”. Дано масив  $a(4)$  із об’єктів типу „тривимірний вектор”. Сформувати масив  $b(4,2)$ , де  $b(i,1)$  – максимальне значення координати  $i$ -го вектора;  $b(i,2)$  – мінімальне значення координати  $i$ -го вектора. Пошук максимального та мінімального елементів оформити у вигляді функції, яка повертає два значення.
2. Дано клас „натуральне число”. Дано два масиви  $a(5)$ ,  $b(5)$  об’єктів класу „натуральне число”. Сформувати два масиви  $nsk(5)$ ,  $ncd(5)$ , в яких записано НСК та НСД відповідних чисел масивів  $a(5)$ ,  $b(5)$ . Пошук НСК та НСД двох натуральних чисел оформити у вигляді функції, яка повертає два значення.
3. Дано клас „масив з 4-х елементів”. Дано масив  $a(3)$  об’єктів класу „масив з 4-х елементів”. Впорядкувати всі масиви в порядку зростання і сформувати двовимірний масив  $b(3,2)$ , де:  
 $b(i,1)$  – кількість перестановок при сортуванні  $i$ -го масиву;  
 $b(i,2)$  – сума елементів  $i$ -го масиву.  
Сортування, знаходження кількості перестановок та суми елементів  $i$ -го масиву оформити у вигляді функції, яка повертає два значення.
4. Дано клас „ масив з 5-х елементів ”. Дано масив  $a(4)$  із об’єктів вищевказаного класу. Сформувати двовимірний масив  $b(4,2)$ , де:  
 $b(i,1)$  – довжина найбільшої підпоследовності нулів в  $i$ -му масиві;

$b(i,2)$  – довжина найбільшої підпослідовності одиниць в  $i$ -му масиві. Пошук довжин відповідних підпослідовностей для  $i$ -го масиву оформити у вигляді функції, яка повертає два значення.

5. Дано клас „точка на площині”. Дано масив  $a(8)$  об’єктів класу „точка на площині”. Знайти найбільшу площу та найбільший периметр трикутника із всеможливих трикутників, які можна сформувати із трьох послідовних точок масиву  $a(8)$ . Пошук периметра та площі трикутника оформити у вигляді функції, яка повертає два значення.



Національний університет  
водо- та природокористування

6. Дано клас „трикутник”, який містить координати трьох його вершин на площині .

Дано масив із п’яти об’єктів типу „трикутник”. Сформувати матрицю  $a(5,3)$ , де  $a(i,1)$  – косинус 1-го кута;  $a(i,2)$  – косинус 2-го кута;  $a(i,3)$  – косинус 3-го кута  $i$ -го трикутника. Пошук косинусів кутів  $i$ -го об’єкта „трикутник” оформити у вигляді функції, яка повертає три значення.

7. Дано клас „опуклий чотирикутник”, який містить координати чотирьох його вершин на площині. Дано масив із чотирьох об’єктів типу „опуклий чотирикутник”. Сформувати матрицю  $d(4,2)$ , де:

$d(i,1)$  – довжина першої діагоналі;  $d(i,2)$  – довжина другої діагоналі  $i$ -го чотирикутника. Пошук довжин діагоналей  $i$ -го об’єкта „опуклий чотирикутник” оформити у вигляді функції, яка повертає два значення.

8. Дано клас „круг”, який містить координати центру та радіус круга. Дано масив із шести об’єктів типу „круг”. Сформувати матрицю  $a(6,2)$ , де:

$a(i,1)$  – площа круга;

$a(i,2)$  – довжина відповідного кола  $i$ -го об’єкта. Пошук довжини кола та площі круга  $i$ -го об’єкта оформити у вигляді функції, яка повертає два значення.

9. Дано клас „тривимірний вектор”. Дано два масиви  $a(4)$  та  $b(4)$  із об’єктів типу „тривимірний вектор”. Сформувати масив  $c(4,2)$ , де:

$c(i,1)$  – значення скалярного добутку векторів  $a(i)$  та  $b(i)$ ;

$c(i,2)$  – різниця між довжинами векторів  $a(i)$  та  $b(i)$ .

Пошук скалярного добутку та різниці між довжинами векторів оформити у вигляді функції, яка повертає два значення.

10. Дано клас „п’ятивимірний вектор”. Дано масив  $a(3)$  із об’єктів типу вищевказаного класу. Сформувати масив  $c(3,3)$ , де:

$c(i,1)$  – довжина  $i$ -го вектора;  $c(i,2)$  – кількість додатних координат  $i$ -го вектора;  $c(i,3)$  – кількість від’ємних координат  $i$ -го вектора.

Пошук довжини, кількості додатних координат, кількості від’ємних координат  $i$ -го вектора оформити у вигляді функції, яка повертає три значення.



Національний університет

11. Дано клас „абонент”. Дано масив  $b(3)$  об’єктів типу „абонент”. Сформувати двовимірний масив  $c(3,2)$ , де:

$c(i,1)$  – кількість простих цифр в номері телефону  $i$ -го абонента;  $c(i,2)$  – сума цифр номера телефону  $i$ -го абонента.

Пошук кількості цифр та суми оформити у вигляді функції, яка повертає два значення.

12. Дано клас „матриця розмірності  $2 \times 2$ ”. Дано масив  $a(4)$  об’єктів класу „матриця розмірності  $2 \times 2$ ”. Сформувати масив  $b(4,2)$ , де:

$b(i,1)$  – визначник  $i$ -ї матриці;  $b(i,2)$  – сума елементів  $i$ -ї матриці.

Пошук визначника та суми елементів  $i$ -ї матриці організувати у вигляді функції, яка повертає два значення.

13. Дано клас „конус”, який містить координати вершин та центра основи конуса, а також радіус основи конуса.

Дано масив  $a(5)$  об’єктів класу „конус”. Сформувати двовимірний масив  $b(5,2)$ , де:

$b(i,1)$  – довжина висоти  $i$ -го конуса;  $b(i,2)$  – довжина твірної  $i$ -го конуса.

Пошук довжин висоти та твірної  $i$ -го конуса описати у вигляді функції, яка повертає два значення.

14. Дано клас „тетраedr”, який містить координати трьох вершин основи та значення висоти тетраедра.

Дано масив  $a(3)$  об’єктів класу „тетраedr”. Сформувати двовимірний масив  $b(3,2)$ , де:

$b(i,1)$  – об’єм  $i$ -го тетраедра;  $b(i,2)$  – довжина найбільшої сторони основи  $i$ -го тетраедра.

Пошук об’єму та найбільшої сторони основи  $i$ -го тетраедра описати у вигляді функції, яка повертає два значення.

15. Дано клас „магазин”, який містить кількість типів товарів ( $\leq 6$ ) та кількість (штук) товару кожного типу.

Дано масив  $a(5)$  об'єктів класу „магазин”. Сформувати масив  $b(5,2)$ , де:  
 $b(i,1)$  – номер товару, кількість якого найбільша в  $i$ -му магазині;  $b(i,2)$  – номер товару, кількість якого найменша в  $i$ -му магазині.  
Пошук елементів  $b(i,1)$  та  $b(i,2)$  оформити у вигляді функції, повертає два значення.

16. Дано клас „будинки”, який містить кількість під'їздів ( $\leq 4$ ) та кількість квартир в кожному під'їзді.

Дано масив  $a(3)$  об'єктів класу „будинки”. Сформувати масив  $b(3,2)$ , де:  
 $b(i,1)$  – номер під'їзду (в  $i$ -му будинку) з найбільшою кількістю квартир;  
 $b(i,2)$  – кількість квартир в  $i$ -му будинку.

Пошук номеру під'їзду та кількості квартир оформити у вигляді функції, яка повертає два значення.

17. Дано клас „клієнт банку”, який містить:

- ✓ прізвище вкладника ;
- ✓ розміри вкладів даного вкладника за кожен місяць поточного року.

Дано масив  $a(5)$  об'єктів класу „клієнт банку”. Сформувати масив  $b(5,2)$ , де:

$b(i,1)$  – величина найбільшого вкладу  $i$ -го клієнта протягом року;  $b(i,2)$  – сумарну величину вкладів  $i$ -го клієнта.

Пошук відповідних величин оформити у вигляді функції, яка повертає два значення.

18. Дано клас „комп'ютер”, який містить кількість логічних дисків ( $\leq 3$ ) та об'єм кожного логічного диску.

Дано масив  $a(4)$  об'єктів класу „комп'ютер”. Сформувати масив  $b(4,2)$ , де:

$b(i,1)$  – об'єм жорсткого диску  $i$ -го комп'ютера;  $b(i,2)$  – номер логічного диску з найбільшим об'ємом в  $i$ -му комп'ютері.

Пошук відповідних величин для  $i$ -го комп'ютера оформити у вигляді функції, яка повертає два значення.

19. Дано клас „школа”, який містить кількість класів ( $\leq 5$ ) та кількість учнів в кожному класі.

Дано масив  $a(3)$  із об'єктів типу „школа”. Сформувати масив  $b(3,2)$ , де:

$b(i,1)$  кількість учнів в  $i$ -й школі;  $b(i,2)$  – кількість учнів в найбільшому класі  $i$ -ї школи.

Пошук  $b(i,1)$  та  $b(i,2)$  оформити у вигляді функції, яка повертає два значення.

20. Дано клас „куля”, який містить координати центра сфери та радіус сфери.

Дано два масиви  $a(6)$  та  $b(6)$  об’єктів класу „куля”. Створити масив  $c(6,2)$ , де:

$c(i,1)$  – об’єм більшої із відповідних куль  $a(i)$  та  $b(i)$ ;

$c(i,2) = \begin{cases} 1, & \text{якщо куля } a(i) \text{ містить кулю } b(i); \\ -1, & \text{якщо куля } a(i) \text{ міститься в кулі } b(i); \\ 0, & \text{в протилежному випадку.} \end{cases}$

Пошук елементів масиву  $c(6,2)$  оформити у вигляді функції, яка повертає два значення.

21. Дано клас „фабрика”, який містить кількість цехів ( $\leq 6$ ) та кількість робочих в кожному цесі.

Дано масив  $a(5)$  об’єктів класу „фабрика”. Сформуванати масив  $c(5,2)$ , де:

$c(i,1)$  – номер цеху з найбільшою кількістю робочих в  $i$ -й фабриці;

$c(i,2)$  – номер цеху з найменшою кількістю робочих в  $i$ -й фабриці.

Пошук елементів  $c(i,1)$  та  $c(i,2)$  оформити у вигляді функції, яка повертає два значення.

22. Дано клас „числова послідовність”, який містить кількість членів числової послідовності ( $\leq 10$ ) та елементи числової послідовності.

Дано масив  $a(6)$  об’єктів класу „числова послідовність”. Сформуванати масив  $c(3,2)$ , де:

$c(i,1)$  – найбільший елемент  $i$ -ї послідовності,

$c(i,2)$  – середнє арифметичне елементів  $i$ -ї послідовності.

Пошук елементів  $c(i,1)$  та  $c(i,2)$  оформити у вигляді функції, яка повертає два значення.

23. Дано клас „поїзд”, який містить кількість вагонів ( $\leq 5$ ) та кількість місць в кожному вагоні.

Дано масив  $a(3)$  об’єктів класу „поїзд”. Сформуванати масив  $c(3,2)$ , де:

$c(i,1)$  – середня кількість місць на один вагон в  $i$ -му поїзді;

$c(i,2)$  – вагон з найбільшою кількістю місць в  $i$ -му поїзді.

Пошук елементів  $c(i,1)$  та  $c(i,2)$  оформити у вигляді функції, яка повертає два значення.

24. Дано клас „сузір’я” (двовимірний випадок), який містить кількість зірок-точок у сузір’ї ( $\leq 5$ ) та координати зірок-точок.

Дано масив  $a(4)$  об'єктів класу „сузір'я”. Сформувати масив  $b(4,2)$ , де:  
 $b(i,1)$  – найбільша відстань між двома зірками в  $i$ -му сузір'ї;  
 $b(i,2)$  – радіус кола з центром в початку координат, в яке попадають всі зірки  $i$ -го сузір'я.  
Пошук елементів  $b(i,1)$  та  $b(i,2)$  оформити у вигляді функції, яка повертає два значення.

25. Дано клас „комплексне число”.

Дано масив  $a(6)$  об'єктів класу „комплексне число”. Сформувати двовимірний масив  $b(6,2)$ , де:

$b(i,1)$  – модуль  $i$ -го комплексного числа;

$b(i,2) = \begin{cases} 1, & \text{якщо число є дійсним;} \\ 0, & \text{в протилежному випадку.} \end{cases}$

Пошук модуля  $i$ -го комплексного числа та визначення, чи є воно дійсним, описати у вигляді функції, яка повертає два значення.

### **Контрольні запитання:**

1. Які обмеження (недоліки) мають функції в C-подібних мовах?
2. Якими засобами можна подолати обмеження питання першого?
3. Дайте означення зсилки.
4. Яким чином здійснюється передача аргументів у функції? Чи зміниться змінна-аргумент функції, якщо її обнулити в тілі функції?
5. Який ключовий символ використовується в C++ для вказання того, що аргумент у функцію передається за зсипкою?
6. Які ключові слова використовуються в C# для використання зсилок у функціях?

### **Лабораторна робота №6**

**Тема: Створення найпростіших Windows- додатків в середовищі. Visual Studio .NET**

**Мета роботи:** навчитися створювати найпростіші Windows-додатків в середовищі Visual Studio .NET

**Хід роботи:**



1. Розробити алгоритми створення найпростіших Windows-додатків в середовищі Visual Studio .NET
2. Написати відповідні програми на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.

### Завдання 6.1



В середовищі програмування Visual Studio .NET розробити проект для виконання нижче наведених варіантів завдань. Розробити зручний інтерфейс, забезпечити правильність введення даних. При подвійному натисканні миші реалізувати очищення полів для введення даних.

#### *Варіанти завдань для самостійної роботи:*

1. Написати програму знаходження площі рівностороннього трикутника, якщо відомий його периметр.
2. Написати програму, для обчислення об'єму кулі, якщо дано її радіус.
3. Написати програму для підрахунку швидкості, з якою спортсмен пробігає дистанцію, якщо задано дистанція та час витрачений на біг.
4. Написати програму для визначення оптимальної ваги людини, якщо задано її зріст.
5. Дано три числа  $a$ ,  $b$ ,  $c$ , які задають сторони рівнобедреного трикутника. Визначити площу даного трикутника.
6. Дано три числа  $a$ ,  $b$ ,  $c$ , які задають сторони рівнобедреного трикутника ( $c$  – основа). Визначити висоту трикутника, опущену на основу  $c$ .
7. Написати програму для обчислення площі паралелограма, якщо задано його сторони  $a$ ,  $b$  та гострий кут  $\alpha$ .
8. Написати програму для обчислення площі прямокутника, якщо задано його сторони  $a$  та  $b$ .
9. Відома довжина кола  $L$ . Знайти площу круга, обмеженого цим колом  $S = \pi R^2$ .
10. Написати програму для обчислення вартості поїздки, якщо задано відстань, вартість палива, споживання палива (літрів/100 км).
11. Написати програму, яка б розраховувала вартість купівлі із врахуванням знижки. Вхідними даними мають бути: кількість продукції, вартість за одиницю продукції, знижка.
12. Знаючи площу прямокутного трикутника і один катет  $a$ , знайти периметр даного трикутника.
13. Дано три числа  $a$ ,  $b$ ,  $c$ , які задають сторони трикутника. Визначити периметр трикутника.
14. Написати програму для знаходження площі сфери за заданим радіусом  $R$  ( $S = 4\pi R^2$ ).

15. Написати програму знаходження об'єму піраміди, якщо задані площа основи  $S_{осн}$  та висота піраміди  $h$  ( $V = \frac{1}{3} S_{осн} h$ ).
16. Скласти програму знаходження площі прямокутного трикутника, якщо відомі два його катети  $a$  та  $b$  ( $S = \frac{ab}{2}$ ).
17. Написати програму для знаходження площі рівнобедреного прямокутного трикутника, якщо відома його гіпотенуза  $c$  ( $S = \frac{c^2}{4}$ ).
18. Написати програму для знаходження площі ромба за заданими діагоналями  $l$  та  $f$  ( $S = \frac{1}{2} lf$ ).
19. Написати програму для знаходження площі трапеції за заданими основами  $a$ ,  $b$  та висотою  $h$   $S = \frac{a+b}{2} h$ .
20. Написати програму для знаходження площі круга діаметром  $d$  ( $S = \pi r^2$ , де  $r$  – радіус круга).
21. Капітал в сумі 1000 грн. внесений на депозит до банку під 4,5 процентів річних. Написати програму, яка б розраховувала накопичення капіталу за 7 років. ( $K_n = K_0 \left(1 + \frac{p}{100}\right)^n$ ).
22. Написати програму для знаходження катету  $b$  прямокутного трикутника, якщо відомий інший катет  $a$  та гіпотенуза  $c$ . Використати теорему Піфагора  $c^2 = a^2 + b^2$ .
23. Дано три числа  $a$ ,  $b$ ,  $c$ , що задають сторони трикутника. Написати програму для визначення площі трикутника використовуючи формулу Герона  $S = \sqrt{p(p-a)(p-b)(p-c)}$ , де  $p$  – півпериметр трикутника.
24. Написати програму, яка б дозволяла проводити операції над двома числами (міні калькулятор): додавання, віднімання, множення, ділення із використанням допоміжних змінних.

### Контрольні запитання:

1. Які в .Net є простори імен для створення додатків з графічним інтерфейсом?
2. Які основні типи даних, що відповідають елементам керування, містять простір імен System. Windows. Forms?
3. Які ви знаєте інструменти середовища Visual Studio.Net для проектування windows-додатків?
4. Які класи є батьківськими для класу Form?
5. Назвіть основні властивості класу Form (які визначені безпосередньо в даному класі).

6. Назвіть основні методи класу `Form` (які визначені безпосередньо в даному класі).
7. Який клас є батьківським для всіх класів, що представляють графічні елементи керування?
8. Назвіть основні події класу `Control`.
9. Назвіть основні методи та властивості класу `Control`.



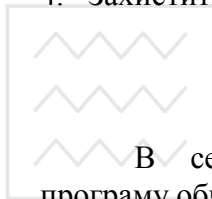
## Лабораторна робота №7

### Тема: Робота з компонентами `Button`, `EditBox`, `Label`, `CheckBox`, `RadioButton`, `GroupBox`

**Мета роботи:** навчитися працювати з компонентами `Button`, `EditBox`, `Label`, `CheckBox`, `RadioButton`, `GroupBox`.

#### Хід роботи:

1. Ознайомитися з роботою компонент `Button`, `EditBox`, `Label`, `CheckBox`, `RadioButton`, `GroupBox`.
2. Написати відповідні програми на одній з мов програмування, використовуючи компоненти.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.



#### Завдання 7.1

В середовищі програмування *Visual Studio.NET* розробити програму обчислення значення функції при заданому значенні аргументу.

##### **Вимоги до проекту:**

- значення змінної вводиться на формі (використати при введенні компоненту *EditBox*);
- значення функції виводиться на формі (використати при виведенні компоненту *EditBox*);
- для текстових підписів використати компоненту *Label*;
- обчислення функції має відбуватись при натисканні кнопки миші на командній кнопці (використати компоненту *Button*);
- на формі має бути наявний незалежний перемикач (*CheckBox*), який при включеному стані, залежно від результату, має забезпечити колір форми за наступним правилом:

$$\text{Колір форми} = \begin{cases} \text{зелений,} & \text{якщо } y > 0; \\ \text{синій,} & \text{якщо } y < 0; \\ \text{червоний,} & \text{якщо } y = 0; \end{cases}$$

- забезпечити правильність введення даних;
- при подвійному натисканні мишею на формі забезпечити очищення полів для введення та виведення даних.



Національний університет  
водного господарства  
та природокористування

### **Варіанти завдань для самостійної роботи:**

$$1. y = \frac{2x}{1-x^2} + \sqrt{x(x+3)};$$

$$2. y = \sqrt[4]{|x-3|^{x+1}} - \log_{1+x} \cos x;$$

$$3. y = 2^{\cos x} + \frac{x^4 + 1}{x^2 \sin^2 x} + x^{1+\lg x};$$

$$4. y = \log_{\cos x} \sin x + \log_{\sqrt{2} \sin x} (1 + \cos x);$$

$$5. y = \operatorname{tg}^2 x + \sqrt{(x+2)^2 - 8x};$$

$$6. y = \frac{\sin x}{1 - \lg(x^2 - 1)} + \log_3 \sqrt{x^2 - 8};$$

$$7. y = \sqrt{x+2} \sqrt{\cos x - \frac{1}{2}} x^6 + 5^{3x};$$

$$8. y = \frac{(\sin x + \cos x)^2 - 2 \sin^2 x}{1 + \operatorname{ctg}^2 x} - \sin^{-1} 4x;$$

$$9. y = 3(\log_2 \sin x)^2 + \log_2 (1 - \cos 2x);$$

$$10. y = \sqrt{\frac{\operatorname{tg} x}{3 + \operatorname{tg} x}} - \log_{\cos x} 2;$$

$$11. y = 2^{\cos x} + \sin x \sqrt{1 + 3 \operatorname{ctg} x};$$

$$12. y = \sin 3x + \sin^3 x - \operatorname{tg}^2 \frac{x}{2};$$

$$13. y = \sqrt{\ln^2 (\sin x + 2) + x^2};$$

$$14. y = \sqrt[3]{x^5 + 5x^4 - 4x^3};$$

$$15. y = \sqrt[5]{x^5 + 3x^3} + 2 \sin x;$$

$$16. y = \sin^{\frac{2}{5}}(3x) + \cos x;$$

$$17. y = 2^{2x} - 4 \cos x;$$

$$18. y = (x^2 - 2x)^3 - 4 \log_3(x^4 + 1);$$

$$19. y = \sqrt[3]{x^4 + 3x^2} - 2 \sin x;$$

$$20. y = \sqrt{x^3 + 3x} + 2 \sin x;$$

$$21. y = \sqrt{x^2 + 3} - 2 \log_5(x^2 + 1);$$

$$22. y = \sin^3(3x) - \cos x;$$

$$23. y = \cos^2(2x) - \sin x;$$

$$24. y = \cos^3(2x) - \log_2(x^2 + 2);$$

$$25. y = \operatorname{tg}(2x) - \lg(x^3 + 2).$$

## Завдання 7.2

В середовищі програмування *Visual Studio.NET* розробити віконні додатки. Розробити зручний інтерфейс та забезпечити правильність введення даних.

### *Варіанти завдань для самостійної роботи:*

1. На формі розмістити групу залежних перемикачів, що відповідають наступним кольорам: синій, голубий, зелений, салатний. Також на формі зобразити круг, який змінює свій колір залежно від того, який перемикач ввімкнений.
2. На форму вводиться п'ять чисел. Визначити чи дана послідовність є строго зростаючою. Якщо так, то на формі з'являється круг, зафарбований в зелений колір, якщо ні – то квадрат червоного кольору.
3. На форму вводиться п'ять чисел. Якщо дана послідовність є строго спадною, то змінити колір фону форми на салатний, в протилежному випадку ніяких дій не відбувається.
4. На формі розміщено три круги : червоний, жовтий і зелений ( вертикально). При натисканні мишкою на круги має з'являтися відповідний надпис: “стій”, “зачекай”, “іди”.
5. На форму вводиться ціле число. Перевірити чи є воно симетричним (яке читається зліва направо і справа наліво однаково). Якщо так – то вивести зелений круг, в протилежному випадку – синій. Також підрахувати кількість цифр у ньому.
6. На форму вводиться ціле число. Підрахувати кількість цифр в ньому і вивести її на форму. Якщо дана кількість парна - то змінити колір фону форми на жовтий, в протилежному випадку – на зелений.
7. На формі розміщений круг. Він змінює своє забарвлення наступним чином: введене число є простим – зелений колір, інший варіант – червоний. Якщо ввімкнений незалежний перемикач, то відбувається

перевірка на парність та непарність введеного числа. Про це повідомляє відповідний напис на формі.

8. На формі розміщені 3 фігури: круг, квадрат і трикутник. Написати програму, яка б змінювала колір будь-яких двох при натисканні на будь-яку третю фігуру.
9. На формі вводяться п'ять чисел. Якщо більшість з них непарні, то вивести на форму зелений круг, в протилежному випадку – червоний круг.
10. На формі вводяться п'ять чисел. Якщо більшість з них парні, то вивести на форму зелений квадрат, в протилежному випадку – червоний.
11. На формі вводиться число. Написати програму для перевірки чи є число парним, чи непарним. Якщо число є парним, то намалювати червоний квадрат, якщо непарне, то намалювати жовте коло.
12. На формі вводяться три числа (компонента - EditBox). Змінити колір компоненти, в якій знаходиться найбільше число, на зелений, середнє число – на жовтий, найменше число – на червоний колір. У випадку рівності будь-яких чисел – змінити на синій колір. Перевірка здійснюється після натискання командної кнопки (компонента Button).
13. На формі вводяться три числа (компонента EditBox). Якщо всі три числа рівні, то змінити колір компоненти EditBox на зелений, в протилежному випадку – на червоний. Перевірку на рівність здійснювати після натискання командної кнопки (компонент Button).
14. На формі вводиться число. Якщо введене число більше нуля, то відбувається закриття додатку, якщо нуль – виводиться повідомлення про повторне введення числа. У випадку, коли число більше одиниці, форма зафарбовується у блакитний колір.
15. Вводиться послідовність чисел через кому. Визначити довжину найдовшої підпослідовності однакових чисел. Якщо довжина менша або рівна 2 на формі з'являється червоний круг.
16. На формі вводиться п'ять чисел. Визначити, чи утворюють дані числа послідовність Фібоначчі. За позитивність чи негативність відповіді відповідає квадрат на формі (компонента **Picture**). Якщо так – квадрат набуває зеленого кольору, ні – червоного.
17. Вводиться послідовність чисел через кому, яка закінчується крапкою. На формі встановлено незалежний перемикач. Якщо він ввімкнений, то шукаються кількість тільки парних чисел, якщо ні – непарних.
18. На формі вводиться три числа. Визначити, чи утворюють дані числа геометричну прогресію. За позитивність чи негативність відповіді відповідає круг розміщений на формі (компонент TShape). Якщо так – круг набуває жовтого кольору, ні – синього.
19. Організувати малювання на формі фігур (коло, квадрат і ін.), зафарбованих певним кольором за вибором користувача. Для вибору

фігур та кольору використати компонента групи залежних перемикачів, для відображення фігур – компонента *Picture*.

20. На формі розміщені дві групи залежних перемикачів, по три перемикачі в кожній. Кожен перемикач відповідає за певний колір.

Наприклад:

перша група перемикачів:

– зелений,

– синій,

– голубий,

друга група перемикачів:

– жовтий,

– червоний,

– оранжевий,

розміщений на формі круг зафарбовується у колір, який утворюється від змішування кольорів від першої та другої групи.

21. Розробити програму, яка дозволяє змінювати колір форми та колір шрифтів. Для цього розмістити на формі дві групи незалежних перемикачів, які відповідатимуть за зміну кольору та зміну шрифту.

22. Написати програму для визначення, чи існує такий трикутник  $\Delta ABC$  із заданими сторонами, наприклад, а)  $AB=5\text{см}$ ,  $AC=18$ ,  $BC=12$ ; б)  $AB=7\text{см}$ ,  $AC=8$ ,  $BC=12$ . якщо трикутник існує, то повинно з'явитися діалогове вікно із відповідним текстом, в протилежному випадку відбувається закриття форми.

23. На формі розміщені три фігури: круг, квадрат та трикутник. Якщо натискаємо мишкою на круг, то трикутник змінює свій колір, а квадрат – ні. Якщо на квадрат – то всі фігури змінюють свій колір. При натисканні на трикутник – круг і квадрат змінюють свій колір.

24. Вводиться три числа. Визначити, чи утворюють дані числа арифметичну прогресію. За позитивність чи негативність відповіді відповідає круг розміщений на формі. Якщо так – круг набуває зеленого кольору, ні – червоного.

### **Контрольні запитання:**

1. Опишіть ієрархію класів елементів керування.
2. Безпосередньо нащадком якого класу є клас `TextBox`?
3. Назвіть основні властивості класу `Button`.
4. Які основні властивості класів `CheckBox` та `RadioButton`?
5. Чим відрізняються класи `CheckBox` та `RadioButton`?
6. Чи є можливість зробити вигляд кнопки „об’ємним”?
7. Яка властивість дозволяє задати зображення, що буде виводитись на кнопки?

8. Чи має клас `RadioButton` властивість `ThreeState`? За що відповідає дана властивість?
9. Які елементи керування забезпечують роботу зі списками?



## Лабораторна робота №8

### Тема: Циклічні алгоритми та програми

**Мета роботи:** повторити основні алгоритмічні структури, навчитися працювати з циклами.

#### Хід роботи:

1. Розробити циклічні алгоритми.
2. Написати відповідні програми на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.

#### Завдання 8.1

В середовищі програмування *Visual Studio .NET* розробити програму обчислення значення функції починаючи із заданої точки  $x_0$ , коли заданий крок  $\Delta x$  та кількість кроків  $n$ , для відповідного аргументу функції (провести табулювання функції). Серед значень функції знайти мінімальне, максимальне та середнє арифметичне значення.

##### **Вимоги до проекту:**

- значення  $x_0$ ,  $\Delta x$  вводяться на формі,
- значення  $n$  (кількість кроків) вводяться на формі за допомогою компоненти *TSpinEdit*;
- значення точності (значення  $t$  – величина може набувати цілі значення від 0 до 6, задає кількість цифр після коми для значення функції) вводиться за допомогою поєднання компонент *TEdit* та *TUpDown*;
- значення функції виводяться на форму з використанням компоненти *TStringGrid*;
- реалізувати контроль наявності факту введення даних;
- забезпечити контролювання правильності введення числових даних для меж табулювання та кроку
- реалізувати контроль за межами табуляції ( $a \leq b$ );



- мінімальне, максимальне та середнє арифметичне значення виводяться на іншу форму.

**Варіанти завдань для самостійної роботи:**

1.  $y = \frac{2t+8}{|\cos 3t|+1}, x \geq 0,1, \Delta t = 0,3, n = 7;$

2.  $y = \frac{\ln|x+1|+5}{2x+3}, x \geq 5, \Delta x = 0,4, n = 6;$

3.  $y = \frac{x^2+2x}{\cos 5x+2}, x \geq 0,6, \Delta x = 0,1, n = 5;$

4.  $y = \frac{t - \ln 2t}{3t+1}, x \geq 0,6, \Delta x = 2,5, n = 5;$

5.  $y = \frac{(x+2)^2}{\sqrt{x^2+1}}, x \geq 6,5, \Delta x = 0,3, n = 4;$

6.  $y = \frac{\operatorname{tg} 2t - 3t}{t+3}, t \geq -0,5, \Delta t = 0,2, n = 5;$

7.  $y = \frac{(3x+2)^2}{\sin x+3}, x \geq 0,2, \Delta x = 0,7, n = 6;$

8.  $z = \frac{2 \sin^2(x+2)}{x^2+1}, x \geq 0, \Delta x = 0,1, n = 5;$

9.  $y = \frac{5 \lg x}{x^2-1}, x \geq 1,2, \Delta x = 1,5, n = 6;$

10.  $y = \frac{5 \operatorname{tg}(x+7)}{(x+3)^2}, x \geq 0,2, \Delta x = 0,1, n = 5;$

11.  $y = \frac{\arccos x}{2x+1}, x \geq 0, \Delta x = 0,2, n = 4;$

12.  $y = \frac{x^3-2}{3 \ln x}, x \geq 2, \Delta x = 1,5, n = 5;$

13.  $z = \frac{t + \sin 2t}{t^2-3}, t \geq 3,1, \Delta t = 0,8, n = 6;$

14.  $y = \frac{x^3+2x}{3 \cos \sqrt{x}+1}, x \geq 0,3, \Delta x = 0,8, n = 7;$

15.  $y = \frac{\cos^3 t^2}{1,5t+2}, t \geq 0, \Delta t = 0,3, n = 5;$

16.  $y = \frac{x + \cos 2x}{3x}, x \geq 0,8, \Delta x = 0,2, n = 6;$

$$17. y = \frac{e^{2x} - 8}{x + 3}, x \geq 1,5, \Delta x = 0,3, n = 6;$$

$$18. y = \frac{\cos^2 x}{x^2 + 1}, x \geq 0,5, \Delta x = 0,1, n = 9;$$

$$19. y = \frac{3x + 1}{\operatorname{arctg} x}, x \geq 0,4, \Delta x = 0,1, n = 5;$$

$$20. y = \frac{1,5x - \ln 2x}{3x + 1}, x \geq 0,8, \Delta x = 1,2, n = 6;$$

$$21. y = \frac{2,5x^3}{e^{2x} + 2}, x \geq -0,8, \Delta x = 0,25, n = 6;$$

$$22. y = \frac{6x + 4}{\sin 3x - x}, x \geq 2,8, \Delta x = 0,3, n = 6;$$

$$23. y = \frac{\operatorname{tg} 0,5x}{x^3 + 7,5}, x \geq 0,5, \Delta x = 0,05, n = 8;$$

$$24. y = \frac{x + \cos 2x}{x + 2}, x \geq 0,6, \Delta x = 1,5, n = 6;$$

$$25. y = \frac{\sin^3 x + \cos x}{x + 1} - x, x \geq 0,5, \Delta x = 1,5, n = 8.$$

### **Контрольні запитання:**

1. Безпосередньо який клас є батьківським для DomainUpDown та NumericUpDown?
2. Скільки операторів циклу є в мові програмування C#?
3. Які відмінності між елементами керування DomainUpDown та NumericUpDown?
4. За допомогою яких елементів керування зручно виводити значення протабульованої функції?
5. Чим відрізняється синтаксис циклічних операторів в мовах програмування C++ та C#?
6. Який простір імен в .Net містить інструменти для забезпечення операцій введення та виведення даних?

## Лабораторна робота №9

### Тема: Робота з одновимірними масивами. Використання меню при створенні проектів

**Мета роботи:** Ознайомитися з особливостями роботи одновимірних масивів, навчитися виконувати обробку одновимірних масивів, використовувати меню при створенні проектів.



Національний університет  
водного господарства  
та природокористування

#### Хід роботи:

1. Розробити алгоритми обробки одновимірного масиву згідно варіантів завдань для самостійної роботи.
2. Написати відповідні програми на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.

#### Завдання 9.1

В середовищі програмування *Visual Studio.Net* розробити програму, яка б для заданого масиву  $D(n)$  із  $n$  елементів виконувала наведені нижче варіанти завдань:

#### Вимоги до проекту:

- розробити головне меню (використати компоненту *MainMenu*), яке б мало, наприклад, наступні розділи:

*Дані*

*Розмірність масиву*

*Генеруються випадковим чином*

*Задаються користувачем*

*Редагування*

*Очистити дані*

*Допомога*

*Інформація про користування програмою*

*Інформація про проект*

*Вихід*

- розробити контекстні мені для кожної компоненти, що знаходиться на формі, та для форми (використати компоненту *PopUpMenu*); в контекстному меню мають бути команди, які стосуються окремої компоненти, або додатку в цілому (якщо меню для форми).
- забезпечити правильність введення даних.

### **Варіанти завдань для самостійної роботи:**

1. Забезпечити заміну елементів, значення яких менші за 1 на середнє арифметичне. Підрахувати кількість проведених замінів.
2. Знайти середнє арифметичне елементів масиву і на місці від'ємних елементів поставити це значення.
3. Визначити середнє арифметичне додатних елементів масиву.
4. Знайти кількість додатних елементів масиву.
5. Визначити номери додатних елементів масиву.
6. Підрахувати кількість елементів масиву, значення яких більше 2,3.
7. Визначити порядкові номери від'ємних елементів масиву.
8. Визначити кількість від'ємних елементів масиву.
9. Обчислити добуток додатних елементів масиву.
10. Знайти мінімальний елемент масиву.
11. Знайти і надрукувати суму додатних елементів масиву.
12. Визначити середнє арифметичне додатних елементів та суму від'ємних.
13. У масиві  $D(n)$  із  $n$  елементів є хоча б один нуль. Визначити кількість елементів після першого нуля.
14. Сформувати новий масив, замінивши елементи, які стоять на парних місцях на елементи, які стоять на непарних місцях та навпаки.
15. Знайти різницю між максимальним серед від'ємних елементів та вказати його номер.
16. Знайти максимальний серед від'ємних елементів та вказати його номер.
17. Знайти мінімальний серед від'ємних елементів та вказати його номер.
18. Знайти мінімальний серед додатних елементів масиву.
19. У масиві  $D(n)$  із  $n$  елементів є хоча б один нуль. Знайти мінімальний по модулю елемент у частині масиву після останнього нуля.
20. Визначити порядкові номери невід'ємних елементів масиву.
21. Знайти максимальний по модулю елемент масиву.
22. Обчислити добуток елементів масиву.
23. Знайти середнє арифметичне невід'ємних елементів масиву, які стоять на непарних місцях.
24. Знайти суму від'ємних елементів, які стоять на парних місцях в масиві.
25. Знайти суму додатних і від'ємних елементів масиву, після першого нуля.

### **Контрольні запитання:**

1. Який циклічний оператор в мові C# дозволяє перебирати всі елементи масиву?
2. Який клас є базовим для всіх меню у Visual Studio.Net?
3. Який клас відповідає для головного меню?
4. Який клас відповідає для контекстного меню?

5. Який клас у Visual Studio.Net забезпечує роботу з масивами?
6. Які основні властивості класу MenuItem?
7. Чи можна створити об'єкт класу Menu?



## Лабораторна робота №10

### Тема: Робота з багатовимірними масивами. Створення багатовіконних додатків

**Мета роботи:** Ознайомитися з особливостями роботи багатовимірних масивів, навчитися виконувати обробку багатовимірних, створювати багатовіконні додатки.

#### Хід роботи:

1. Розробити алгоритми обробки багатовимірних масивів згідно варіанту.
2. Написати відповідні програми на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу.

#### Завдання 10.1

В середовищі програмування *Visual Studio.Net* розробити програми, що реалізують нижченаведені варіанти завдань. Забезпечити виконання наступних вимог:

- створити головне меню (використати головне меню *MainMenu*);
- розмірність матриць вводити на окремій формі;
- для введення матриць використати компоненту *Memo*;
- виведення результатів має здійснюватися на окремому формі.

#### Варіанти завдань для самостійної роботи:

1. Дана квадратна матриця цілих чисел  $A$  розмірності  $n \times n$ . Визначити:
  - 1) кількість рядків, в яких середнє арифметичне елементів цього рядка дорівнює номеру рядка;
  - 2) кількість стовпчиків, в яких середнє арифметичне елементів цього стовпчика дорівнює номеру стовпчика;
  - 3) кількість елементів  $a_{ij}$  таких, що середнє арифметичне елементів матриці дорівнює  $i + j$ .
2. Дана квадратна матриця цілих чисел  $A$  розмірності  $n \times n$ . Визначити:

- 1) кількість елементів  $a_{ij}$  таких, що  $a_{ij} = i + j$ ;
- 2) кількість елементів  $a_{ij}$  таких, що  $a_{ij} > i + j$ ;  
кількість елементів  $a_{ij}$  таких, що  $a_{ij} < i + j$ .
3. Дана квадратна матриця цілих чисел  $A$  розмірності  $n \times n$ . Визначити:
  - 1) номери елементів, для яких  $a_{ij} = i + j$ ;
  - 2) номери елементів, для яких  $a_{ij} = i - j$ ;

номери елементів, для яких  $a_{ij} = i = j$ .

4. Дана матриця  $A (m \times n)$ . Знайти мінімальний та максимальний елемент головної діагоналі, кількість елементів матриці (окрім елементів головної діагоналі), які дорівнюють максимальному її елементу та мініимальному.

5. Дана квадратна матриця  $A$  розмірності  $n \times n$ . Сформувати два масиви  $B = \{b_i\}$  та  $C = \{c_j\}$  за правилом

$$b_i = \begin{cases} 1, & \text{якщо відповідна паралельна діагональ до головної діагоналі матриці } A \\ & \text{є зростаючою послідовністю;} \\ -1, & \text{якщо спадною;} \\ 0, & \text{в протилежному випадку;} \end{cases}$$

$$c_j = \begin{cases} 1, & \text{якщо відповідна паралельна діагональ до побічної діагоналі матриці } A \\ & \text{є спадною послідовністю;} \\ -1, & \text{якщо зростаючою;} \\ 0, & \text{в протилежному випадку;} \end{cases}$$

6. Дана квадратна матриця  $A$  розмірності  $n \times n$ . Сформувати матрицю  $C = \{c_{ij}\}$ , де  $c_{ij}$  – кількість цифр в числі, записаному в  $a_{ij}$ . Якщо кількість цифр перевищує 4, то  $c_{ij} = 0$ .

7. Дана квадратна матриця  $A$  розмірності  $n \times n$ . Сформувати два масиви  $B = \{b_i\}$  та  $C = \{c_j\}$  за правилом

$$b_i = \begin{cases} 1, & \text{якщо } i\text{-й рядок матриці } A \text{ утворює арифметичну прогресію;} \\ 0, & \text{в протилежному випадку;} \end{cases}$$

$$c_j = \begin{cases} 1, & \text{якщо } j\text{-й стовпець матриці } A \text{ утворює арифметичну прогресію;} \\ 0, & \text{в протилежному випадку;} \end{cases}$$

8. Дана матриця  $A (m \times n)$ . Сформувати матрицю  $B = \{b_{ij}\}$ , де  $b_{ij}$  дорівнює остачі від ділення числа  $a_{ij}$  на число  $i$ , та матрицю  $C = \{c_{ij}\}$ , де  $c_{ij}$  дорівнює сумі парних елементів матриці  $B$  в  $j$ -му стовпці.

9. Дано дві матриці  $A$  і  $B$  ( $m \times n$ ). Сформувати матрицю  $C = \{c_{ij}\}$  за

правилом 
$$C_{ij} = \begin{cases} 0, & a_{ij} = b_{ij}, a_{ij} - \text{непарне} \\ 1, & a_{ij} > b_{ij}, b_{ij} - \text{парне} \\ -1, & a_{ij} < b_{ij}, b_{ij} - \text{непарне} \\ 10, & \text{в протилежних випадках} \end{cases}$$

10. Дана квадратна матриця  $A$  розмірності  $n \times n$  ( $n$  – парне число), яка складається із цілих чисел. Якщо в непарних рядочках кількість парних чисел непарна, то дані рядки поміняти з непарними стовпчиками, інакше – з парними.

11. Дана квадратна матриця  $A$  розмірності  $n \times n$ . Передбачити введення 2-х чисел  $k$  та  $j$ , де  $k, j \leq n$ . Якщо добуток елементів  $k$ -го стовпчика менше рівний добутку елементів  $j$ -го рядка, то поміняти їх місцями, інакше – провести заміну елементів  $k$ -го стовпчика на добуток елементів  $j$ -го рядка.

12. Дана матриця  $A$  ( $m \times n$ ). Здійснити перетворення наступним чином: якщо сума індексів елементу матриці більша за сам елемент – то замінити значення даного елементу на одиницю, рівна значенню елементу – то замінити значення даного елемнту на два, менша – на три. Підрахувати кількість 1, 2, 3 в матриці. На формі зобразити круг і зафарбувати його наступним чином, якщо найбільшу кількість в матриці складають  $\begin{cases} 1, & \text{червоним;} \\ 2, & \text{жовтим;} \\ 3, & \text{зеленим.} \end{cases}$  Якщо максимум складають декілька номерів, то круг замалювати синім кольором.

13. Дано дві матриці  $A$  і  $B$  ( $m \times n$ ). Сформувати матрицю  $C = \{c_{ij}\}$ , де  $c_{ij}$  – сума кількостей парних елементів  $i$  – го рядка матриці  $A$  та  $j$  – го стовпця матриці  $B$ .

14. Дана квадратна матриця  $A$  розмірності  $n \times n$  ( $n$  – парне число), яка складається із цілих чисел. Якщо в парних стовпчиках кількість непарних чисел парна, то дані в стовпчиках замінити на дані непарних рядків, а якщо ні – то парних рядків.

15. Дана матриця  $A$  ( $m \times n$ ). Замінити всі елементи даної матриці, сума індексів яких парна, добутками відповідних індексів.

16. Дана матриця  $A$  ( $m \times n$ ). Отримати нову матрицю шляхом ділення всіх її елементів на найбільший по модулю елемент.

17. Дана матриця  $A$  ( $m \times n$ ). Визначити середні арифметичні значення елементів стовпчиків.

18. Дана матриця  $A$  ( $m \times n$ ). Визначити різниці максимального та мінімального елементів кожного стовпчика.

19. Дана матриця  $A$  ( $m \times n$ ). Визначити суми та добутки елементів рядків.

20. Дана матриця  $A (m \times n)$ . Знайти кількість від'ємних елементів, що лежать нижче головної діагоналі.
21. Дана матриця  $A (m \times n)$ . Знайти мінімальний елемент для даної матриці та вивести його. Замінити елементи матриці, що лежать нижче головної діагоналі, на мінімальний елемент.
22. Дана матриця  $A = \{a_{ij}\} (n \times n)$ . Знайти  $c = \frac{a_{\min} + a_{\max}}{2}$ , де  $a_{\min} = \min \{a_{ij}\}$ ,  
 $a_{\max} = \max \{a_{ij}\}$ .
23. Дано матрицю  $B (n \times n)$ . Перетворити матрицю, замінивши додатні елементи – номером рядка, в якому вони знаходяться, від'ємні елементи – номером стовпчика, в якому вони знаходяться, нульові елементи – сумою відповідного рядка та стовпчика.
24. Провести перетворення матриці  $A$  таким чином: додатні елементи замінити на число 2, від'ємні - на 1, нульові - на 3.
25. Дана матриця  $A (m \times n)$ . Знайти максимальний елемент матриці, вивести його. Замінити елементи головної діагоналі на максимальний елемент матриці.

#### **Контрольні запитання:**

1. Чи передбачений окремий клас в .Net для створення діалогових вікон?
2. Який метод класу форми використовується для відкриття модального діалогового вікна?
3. Для чого використовується перелічуваний тип даних DialogResult?
4. Який клас в .Net зручно використовувати для роботи з багатовіконними додатками?
5. Який клас в .Net використовується для створення головного меню?
6. Які простори імен в .Net використовується для роботи з графічними зображеннями?



## РОЗДІЛ V. Програмування на мові Delphi

### Загальні вимоги до виконання лабораторних робіт

Вся тематика лабораторних робіт спрямована на розробку в середовищі програмування *Delphi* проектів для розв'язання задач різного призначення. Відповідно до вимог програми з даної дисципліни розроблено 8 лабораторних робіт, виконання яких кожним студентом є обов'язковим для допуску до здачі відповідного іспиту (заліку). Практикум містить необхідний мінімум теоретичних відомостей, приклади програм з детальними коментарями до них, завдання для самостійної роботи.

### Порядок виконання лабораторних робіт

1. Ознайомитись з темою, метою роботи та засвоїти матеріал теоретичної частини.
2. Користуючись прикладами, які наведені в кожній лабораторній роботі, вивчити методи роботи з компонентами.
3. Вибрати відповідний варіант (за вказівкою викладача) та ознайомитись із завданням.
4. Розробити проект, що означає виконати наступні етапи:
  - проектування інтерфейсу проекту.
  - програмування: здійснити програмування процедур опрацювання подій.
5. Зберегти результати виконаної роботи в індивідуальній робочій папці.
6. Продемонструвати результати виконаної роботи викладачу.
7. Оформити звіт про виконання лабораторної роботи, який включає: назву теми, завдання, хід роботи, тексти програм, коментарі до них.
8. Захистити звіт про виконання лабораторної роботи. При захисті вимагається пояснити всі етапи виконання лабораторної роботи та відповісти на контрольні питання, які зазначені в кінці лабораторної роботи.

## Лабораторна робота №1

**Тема: Ознайомлення з середовищем візуального об'єктно-орієнтованого програмування Delphi. Створення найпростіших програм.**

### **Мета роботи:**

- Ознайомитись з середовищем візуального об'єктно-орієнтованого програмування **Delphi**.
- Ознайомитись з такими поняттями: **проект, компонента, палітра компонент, об'єкт, інспектор об'єктів, форма, редактор коду**.
- Ознайомитись з складовими проекту.
- Ознайомитись та навчитись використовувати основні компоненти сторінки **Standard** палітри компонент.

### **Теоретичні відомості:**

В процесі виконання завдань даної лабораторної роботи будуть використані наступні компоненти: **TEdit, TLabel, TButton**. Їх призначення та основні властивості наведено нижче.

#### Компонента **TEdit**

##### Призначення:

Вікно редагування, яке призначене для введення/відображення одного текстового рядка. Має свої власні внутрішні команди редагування. Показує однострічкове поле редагування.

##### Основні властивості:

Для компоненти **TEdit** текст, що вводиться та виводиться, є значенням властивості **Text** типу **String**. Дану властивість можна задавати під час проектування або програмно. Дії із вирівнюванням тексту та перенесенням стрічки неможливі.

Властивість **AutoSize** забезпечує автоматичне визначення висоти вікна редактору залежно від розміру тесту. Властивість **AutoSelect** визначає, чи буде автоматично виділятися весь текст при передачі фокусу у вікно редагування (слід задавати значення **true** у випадку, коли вікно призначене здебільшого для заміни поточного тексту, ніж для редагування). У випадку, коли компоненту **TEdit** використовують для відображення тексту, то для властивості **ReadOnly** слід задавати значення **False** (також доцільно встановити для властивості **AutoSelect** значення **False**).

При використанні вікон редагування для введення, виведення та редагування чисел потрібно використовувати функції перетворення типів:

– при виведенні використовуються функції **FloatToStr**, **FloatToStrF**, **IntToStr**;

– при введенні використовують функції *StrToFloat*, *StrToInt*.

У випадку, коли текст, що вводиться не є числом, тобто містить недопустимі символи для числа, то функції перетворення генерують виключення *EConvertError* (в програмах завжди потрібно передбачати обробку даного виключення).

Як приклад, можна навести код програми, що забезпечує повідомлення користувачу, про помилку введення та попереджує помилкове введення (даний приклад не є найкращим, оскільки користувач дізнається про помилку після того, як програма намагається використати введені дані).



```
var i:integer;
.....
try
.....
i:=StrToInt(Edit1.Text);
.....
except
  on EConvertError do
    ShowMessage('Помилка при введенні числа: повторіть
                спробу');
end;
```

Наступний приклад процедури не дозволяє користувачу в компоненті *TEdit* вводити неправильні символи (наприклад, користувачу можна вводити тільки цифри).

```
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
  if not ( ( ( Key >= '0' ) and ( Key <= '9' ) ) or ( Key = #8 ) ) then
    Key:=Chr(0);
end;
```

Властивість *MaxLength* визначає максимальну довжину тексту, що вводиться (якщо *MaxLength* = 0, то довжина тексту необмежена). Властивість *Modified* доступна тільки під час виконання програми та показує, чи проводилось редагування тексту у вікні редактора.

Серед найбільш використовуваних методів можна назвати метод *SetFocus*, за допомогою якого здійснюється передача фокуса елемента керування.

### Компонента TLabel

Призначення:

Текстовий підпис. Використовують для розміщення у вікні не дуже довгих записів, які постійно наявні і допомагають лише орієнтуватися в інтерфейсі, чи роботі програми.

*Основні властивості:*

Текст, що відображається у компоненті *TLabel*, визначається значенням її властивості *Caption*. Дану властивість можна задавати в процесі прехтування або задавати чи змінювати програмно в процесі виконання програми.

*Наприклад: `Label1.Caption := 'Текст задається програмно';`*

Якщо існує необхідність відобразити числову інформацію, то використовуються функції *FloatToStr* та *IntToStr*, що перетворюють відповідно дійсні числа та цілі в стрічку. Для формування тексту, що складається із декількох фрагментів, використовують операцію „+”, що означає конкатенацію.

*Наприклад*, в програмі є змінна *X*, що відображає кількість балів, які отримав студент на іспиті. Для відображення даної інформації в мітці (рис.5.1) *Label1* можна здійснити оператором:

*`Label1.Caption := 'Кількість балів за іспит: ' + IntToStr(X);`*

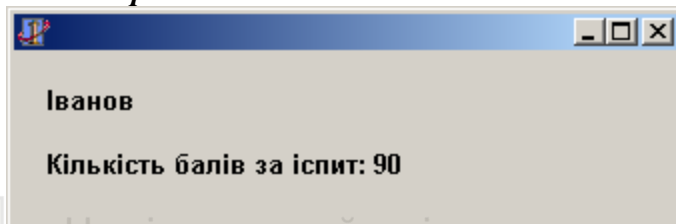


Рис.5.1. Приклад відображення інформації на мітці *Label1*

Колір фону компоненти *TLabel* визначається властивістю *Color*, а колір тексту – підвластивістю *Color* властивості *Font*. Дані властивості – це єдині можливі елементи оформлення надпису.

Альтернативою компоненти *TLabel* є компонента *TStaticText*, яка крім вище зазначених властивостей має властивість *BorderStyle*, що визначає рамку тексту. При стилі *sbsNone* мітка *StaticText* за зовнішнім виглядом ні чим не відрізняється від мітки *TLabel*.

Розміщення компоненти на формі визначається, зокрема, властивостями *Top*, *Left*, *Height*, *Width*, *Aline*, *Anchors*, *Constraints*. Ці властивості визначають координати компоненти, її розміри та їх зміни при зміні користувачем розмірів батьківського вікна.

Розмір компоненти *TLabel* та *TStaticText* визначається властивістю *AutoSize*. Якщо властивість має значення *true*, то розмір по вертикалі та горизонталі визначаються розміром надпису. Якщо властивість *AutoSize* встановлюється як *false*, то вирівнювання тексту в середині компоненти визначається властивістю *Alignment*. Компонента *TLabel* має також

властивість **Wordwrap** – можливість перенесення слів довгого надпису, що перевищує довжину компоненти, на нову стрічку. Для реалізації такої можливості потрібно, щоб властивість **WordWrap** приймала значення **true**, а властивість **AutoSize** мала значення **false** (щоб розмір компоненти не визначався розміром надпису) та зробити висоту компоненти такою, щоб в ній можна було розмістити декілька рядків. У випадку, коли **Wordwrap** не встановлено в **true** при **AutoSize** із значенням **false**, то довгий текст, що не вміщається в мітці, просто обрізається. Хоча в мітці **TStaticText** перенесення довгого тексту здійснюється автоматично, якщо значення **AutoSize** приймає значення **false** та розмір компоненти достатній для розміщення декількох рядків.



### Компонента **TButton**

*Призначення:*

Створює командну кнопку. Використовується для створення кнопок, якими користувач виконує команди в додатку.

*Основні властивості:*

Командна кнопка **TButton** розташовується на сторінці **Standard**. Основна властивість із точки зору зовнішнього вигляду – властивість **Caption** (надпис, або називають заголовок). В надписах кнопок можна передбачати використання клавіш швидкого доступу. Для цього виділяється один із символів надпису підкреслюванням. Перед символом, який повинен відповідати клавіші швидкого доступу, ставиться символ **&**, в свою чергу цей символ в надписі не з'являється, а наступний за ним символ буде підкресленим. Тоді користувач замість натискання миші може використовувати комбінацію клавіш **Alt** та клавішу виділеного символу. Наприклад, для кнопки (Рис.5.2) „Close” можна забезпечити виконання події, яка визначена для неї, комбінацією клавіш **Alt** та ‘c’.

Основна подія будь-якої кнопки – **OnClick**, що виникає при одноразовому натисканні мишею. Для обробки даної події автоматично створюється процедура, в якій записуються оператори, що повинні виконуватися при одноразовому натисканні мишею.

Якщо властивість компоненти **Cancel** приймає значення **true**, то кнопка матиме еквівалентні властивості клавіші **Esc**. Дана властивість використовується для створення кнопок «Відмінити» в різноманітних діалогових вікнах (у такому випадку слід задавати значення **true**).

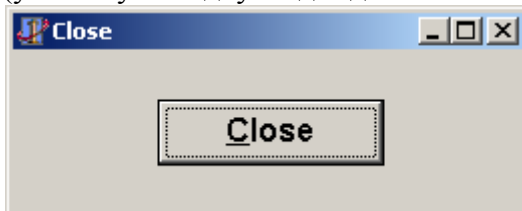


Рис.5.2. Приклад створення кнопок швидкого доступу

Властивість **Default** при значенні **true** забезпечує створення кнопок, які еквівалентні за властивостями клавіші **Enter**. Виконання події даної кнопки можна продублювати натисканням на клавіатурі клавіші **Enter**, якщо вона навіть не буде знаходитись на даний момент у фокусі. Хоча у випадку, коли в момент натискання **Enter** у фокусі знаходиться інша кнопка, то все-таки вступить в дію обробка події для тієї кнопки, що знаходиться у фокусі.



Серед методів, що притаманні кнопкам слід визначити один – **Click**. Виконання даного методу еквівалентне одноразовому натисканні миші, тобто викликається подія кнопки **OnClick**. (цим можна скористатися, щоб продублювати якимись іншими діями користувача одноразову фіксацію миші). В свою чергу невідомо, яка компонента буде знаходитись у фокусі у момент цієї події, тому потрібно перехопити її на рівні форми. Таке перехоплення можна зробити, якщо задати властивість форми **KeyPreview** в **true**.

Наприклад, нам потрібно, щоб при одноразовому натисканні мишею, при натисканні на клавішу із символом 'C' або 'c' будь-який момент виконувались операції, передбачені при обробці події **OnClick** кнопки **Button1**. Тоді процедура форми, що здійснює обробку події **OnKeyPress**, матиме наступний вигляд:

```
procedure TForm1.FormKeyPress(Sender: TObject; var Key: Char);
```

```
begin
```

```
if ((Key='C') or (Key='c')) then Button1.Click;
```

```
end;
```

### Приклади розробки Delphi-додатків:

#### Приклад 5.1

В середовищі програмування **Delphi** розробити програму для підрахунку суми двох чисел. При розробці програми використати компоненти класу **TEdit**, **TLabel** палітри **Standard**. Реалізувати програму без використання допоміжних змінних.

**Етап проектування** інтерфейсу проекту:

При формуванні інтерфейсу проекту (рис.5.3) використаємо компоненти класу **TEdit**, **TLabel**:

```
Form1.Caption:='Сума чисел';
```

```
Edit1.Text:='';
```

```
Edit2.Text:='';
```

```
Edit3.Text:='';
```

***Label1.Caption:= '+';***  
***Label2.Caption:= '=';***

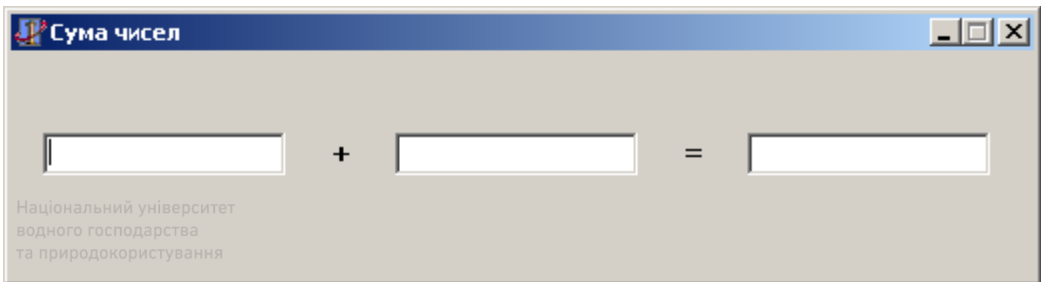


Рис.5.3. Вікно форми проекту прикладу 5.1

***Етап програмування*** проекту:

При натисканні мишкою на знак ***маємо*** отримати результат сумування двох чисел. Обробку події натискання на текстовий надпис “=” здійснює наступна процедура:

```
procedure TForm1.Label2Click(Sender: TObject);  
begin  
  
Edit3.Text:=FloatToStr(StrToFloat(Edit1.Text)+  
StrToFloat(Edit2.Text));  
end;
```

***Результат виконання*** проекту:

Результат виконання програми подано на рис.5.4.



Рис.5.4. Вікно форми під час виконання проекту прикладу 5.1

### Приклад 5.2

В середовищі програмування **Delphi** розробити програму, яка б дозволяла знаходити розв'язки квадратного рівняння. Розробити зручний інтерфейс, забезпечити правильність введення даних.

**Етап проектування інтерфейсу проекту:**

При формуванні інтерфейсу проекту (рис.5.5) використаємо компоненти класу **TEdit**, **TLabel**, **TImage**:

**Label1.Caption:= 'Дана програма дозволяє розв'язувати квадратні рівняння';**

**Label2.Caption:= 'a';**

**Label2.Caption:= 'b';**

**Label3.Caption:= 'c';**

**Label5.Caption:= ' ';** {на рис.5.5 компоненти **Label5**, **Label6**, **Label7** не

**Label6.Caption:= ' ';** відображаються за рахунок відсутнього

**Label7.Caption:= ' ';** надпису}

**Button1.Caption:= 'Розв'язок';**

**Button2.Caption:= 'Нові дані';**

**Image.Picture:= '1.bmp'** {на вставленому малюнку зобразити рівняння}

**Edit1.Text:= '';**

**Edit2.Text:= '';**

**Edit3.Text:= ''.**

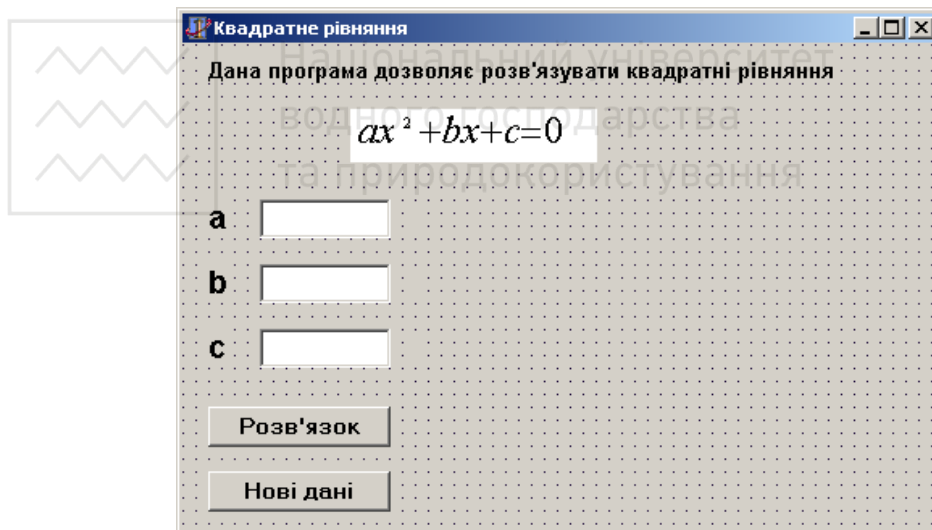


Рис.5.5. Вікно форми проекту прикладу 5.2



**Етап програмування** проекту:

Процедура обробки події натискання мишею кнопки “Розв’язок” матиме наступний вигляд:

```
procedure TForm1.Button1Click(Sender: TObject);  
var D,a,b,c,x,x1,x2: real;  
begin  
  // для початку здійснюється перевірка, чи введені всі дані  
  if (Edit1.Text='') or (Edit2.Text='') or (Edit3.Text='') then  
    begin  
      ShowMessage('Введіть всі коефіцієнти');  
      exit;  
    end;  
  a:=StrToFloat(Edit1.Text);  
  b:=StrToFloat(Edit2.Text);  
  c:=StrToFloat(Edit3.Text);  
  if a=0 then  
    begin  
      ShowMessage('Коефіцієнт a=0, рівняння не є квадратним');  
      exit;  
    end  
    else D:=b*b-4*a*c;  
    if D<0 then Label5.Caption:='Дане рівняння дійсних коренів  
                                     немає '  
  else  
    if D=0 then  
      begin  
        Label5.Caption:='Дане рівняння має один дійсний  
                                     корінь';  
        x:=-b/(2*a);  
        Label6.Caption:='x='+FloatToStr(x);  
      end  
    else  
      begin  
        Label5.Caption:='Дане рівняння має два дійсні  
                                     корені';  
        x1:=(-b+sqrt(D))/(2*a);  
        x2:=(-b-sqrt(D))/(2*a);  
        Label6.Caption:='x1='+FloatToStrF(x1,ffFixed,5,2);  
        Label7.Caption:='x2='+FloatToStrF(x2,ffFixed,5,2);  
      end;  
  end;
```



Національний університет  
водного господарства  
та природокористування



Національний університет  
водного господарства  
та природокористування

В програмі також варто забезпечити аналіз символів, що будуть вводиться користувачем. Це означає, що потрібно блокувати символи, які несумісні із представленням чисел. Процедура *TForm1.Edit1KeyPress* є універсальною, оскільки використовується для всіх полів введення:

```

procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
var str:string[30];
begin
  if Sender=Edit1 then str:=Edit1.Text
  else if Sender=Edit2 then str:=Edit2.Text
  else str:=Edit3.Text;
  case Key of
    '0'..'9',Chr(8); // визначаються допустимі символи при введенні
    ',': if pos(',',str)<>0 then Key:=Chr(0); //забезпечується введення
    // тільки одного символу ','
    '-': if Length(str)<>0 then Key:=Chr(0); //визначається, що символ '-'
    // має стояти тільки першим
    else Key:=Chr(0) // недопустимі символи блокуються
  end
end;

```

Для введення нових даних реалізована процедура *TForm1.Button2Click*, що дозволяє очистити форму від попередніх результатів.

```

procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Text:=' ';
  Edit2.Text:=' ';
  Edit3.Text:=' ';
  label5.Caption:=' ';
  label6.Caption:=' ';
  label7.Caption:=' ';
  Edit1.SetFocus;
end;

```

**Результат виконання проекту:**

Наприклад, для рівняння  $x^2 - 7x + 12 = 0$  маємо наступні результати, які подано на рис.6.

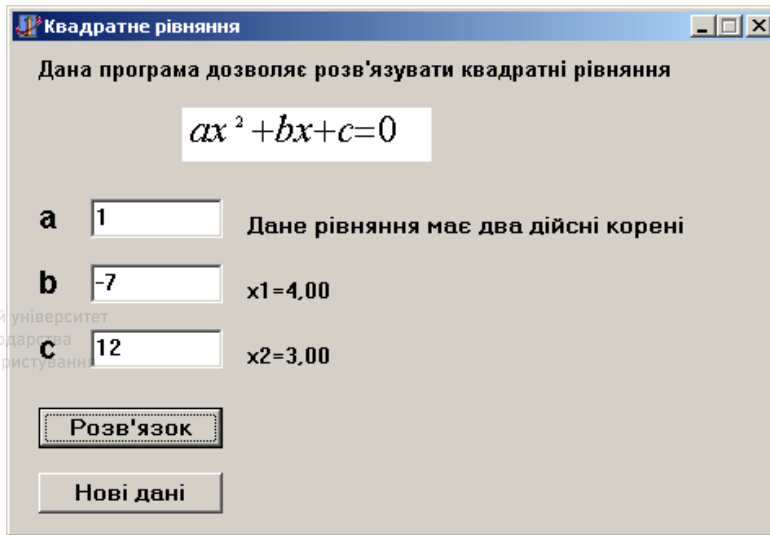


Рис.5.6. Вікно форми виконання проекту прикладу 5.2

### Завдання для виконання лабораторної роботи:

В середовищі програмування **Delphi** розробити проект для виконання нижче наведених варіантів завдань. Розробити зручний інтерфейс, забезпечити правильність введення даних. При подвійному натисканні миші реалізувати очищення полів для введення даних.

#### Варіанти завдань:

1. Написати програму, яка б дозволяла проводити операції над двома числами (міні калькулятор): додавання, віднімання, множення, ділення із використанням допоміжних змінних.
2. Дано три числа  $a$ ,  $b$ ,  $c$ , що задають сторони трикутника. Написати програму для визначення площі трикутника, використовуючи формулу Герона  $S = \sqrt{p(p-a)(p-b)(p-c)}$ , де  $p$  – півпериметр трикутника.
3. Написати програму для знаходження катету  $b$  прямокутного трикутника, якщо відомий інший катет  $a$  та гіпотенуза  $c$ .
4. Капітал в сумі 1000 грн. внесений на депозит до банку під 4,5 процентів річних. Написати програму, яка б розраховувала накопичення капіталу за 7 років.
5. Написати програму для знаходження площі круга діаметром  $d$ .
6. Написати програму для знаходження площі трапеції за заданими основами  $a$ ,  $b$  та висотою  $h$ .

7. Написати програму для знаходження площі ромба за заданими діагоналями  $l$  та  $f$ .
8. Написати програму для знаходження площі рівнобедреного прямокутного трикутника, якщо відома його гіпотенуза  $c$ .
9. Скласти програму знаходження площі прямокутного трикутника, якщо відомі два його катети  $a$  та  $b$ .
10. Написати програму знаходження об'єму піраміди, якщо задані площа основи  $S_{осн}$  та висота піраміди  $h$ .
11. Написати програму для знаходження площі сфери за заданим радіусом  $R$ .
12. Дано три числа  $a$ ,  $b$ ,  $c$ , які задають сторони трикутника. Визначити периметр трикутника.
13. Знаючи площу прямокутного трикутника і один катет  $a$ , знайти периметр даного трикутника.
14. Написати програму, яка б розраховувала вартість купівлі із врахуванням знижки. Вхідними даними мають бути: кількість продукції, вартість за одиницю продукції, знижка.
15. Написати програму для обчислення вартості поїздки, якщо задано відстань, вартість палива, споживання палива (літрів/100 км).
16. Відома довжина кола  $L$ . Знайти площу круга, обмеженого цим колом.
17. Написати програму для обчислення площі прямокутника, якщо задано його сторони  $a$  та  $b$ .
18. Написати програму для обчислення площі паралелограма, якщо задано його сторони  $a$ ,  $b$  та гострий кут  $\alpha$ .
19. Дано три числа  $a$ ,  $b$ ,  $c$ , які задають сторони рівнобедреного трикутника ( $c$  – основа). Визначити висоту трикутника, опущену на основу  $c$ .
20. Дано три числа  $a$ ,  $b$ ,  $c$ , які задають сторони рівнобедреного трикутника. Визначити площу даного трикутника.
21. Написати програму для визначення оптимальної ваги людини, якщо задано її зріст.
22. Написати програму для підрахунку швидкості, з якою спортсмен пробігає дистанцію, якщо задано дистанція та час витрачений на біг.
23. Написати програму, для обчислення об'єму кулі, якщо дано її радіус.
24. Написати програму знаходження площі рівностороннього трикутника, якщо відомий його периметр.
25. Огорожа має форму кола та обмежує ділянку площею  $S$ . Якою буде сторона квадрата, якщо цією огорожею обмежити квадратну ділянку?

### **Контрольні запитання:**

1. Що таке проект?
2. Перерахуйте склад вікна *Delphi*.
3. Назвіть основні елементи вікна форми.

4. Назвіть основні елементи Інспектора Об'єктів.
5. Яким чином здійснюється розміщення компонент на форму?
6. Дайте означення події в *Delphi*.
7. Які файли включає *Delphi*-додаток?
8. Як здійснюється компіляція програми?



## Лабораторна робота №2

**Тема:** Розробка лінійних програм.

### **Мета роботи:**

- Навчитись працювати з основними візуальними компонентами палітри **Standard: TEdit, TButton, TCheckBox, TLabel, TRadioGroup, TRadioButton** та палітри **Additional: TShape, TStaticText, TBitBtn, TLabeledEdit**.
- Розглянути та вивчити властивості компонент панелі **Standard, Additional** та методи роботи з ними.

### **Теоретичні відомості:**

В процесі виконання завдань даної лабораторної роботи будуть використані наступні компоненти: **TEdit, TLabel, TButton, TBitBtn, TRadioButton, TRadioGroup, TGroupBox, TCheckBox, TShape, TLabeledEdit**. Призначення та основні властивості нових компонент, які в попередній лабораторній роботі не використовувалися, наведено нижче.

### **Компонента TBitBtn**

#### **Призначення:**

Використовується для створення кнопок, на яких розташовується бітова графіка.

#### **Основні властивості:**

Властивості аналогічні до компоненти **TButton**. Визначимо особливі властивості кнопки з піктограмою. Зображення на цій кнопці задається властивістю **Glyph**. При натисканні кнопки з крапками в стрічці властивості **Glyph** в Інспекторі об'єктів викликається вікно, яке показано на рис.5.7.

Натискаючи на кнопку **Load**, відбувається перехід у звичайне вікно відкриття файлу малюнка, де можна вибрати файл бітової матриці \*.bmp бажаного зображення. Після вибору, натиснувши на кнопку **OK**, вибране зображення з'явиться на кнопці зліва від надпису (коли властивість **Layout** приймає значення **biGlyphLeft**). Файл зображення для кнопки може

містити до чотирьох зображень піктограм розмірності 16×16. Кількість піктограм на кнопці можна дізнатися із властивості *NumGlyphs*, значення якої з'являється після завантаження. Більшість зображень для кнопок використовує дві піктограми. Перша зліва піктограма відповідає малюнку на кнопці, яка знаходиться у звичайному стані, друга зліва відповідає недоступній кнопці, коли її властивість *Enabled* визначається як *False*.

Розташування зображення та надпису на кнопці визначається властивостями *Margin*, *Layout* та *Spacing*. Коли властивість *Margin* приймає значення **-1**, то зображення та надпис розміщуються в центрі кнопки, якщо *Margin* приймає значення **>0**, то залежно від значення властивості *Layout* зображення та надпис зміщуються до тієї, чи іншої границі кнопки. Властивість *Layout* визначає розташування зображення відносно надпису. Може приймати наступні значення: *blGlyphLeft* – зліва (приймається по-замовчуванню); *blGlyphRight* – справа; *blGlyphTop* – зверху; *blGlyphBottom* – внизу.

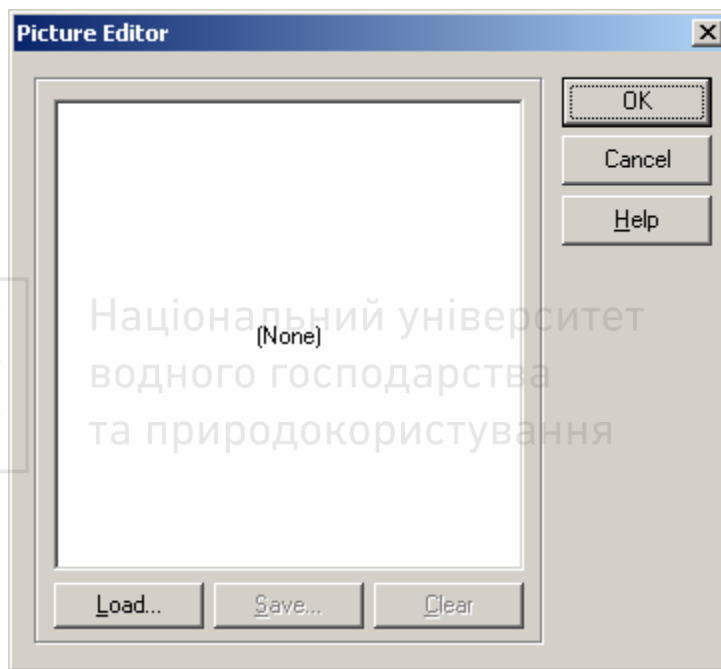


Рис.5.7. Вікно *Picture Editor*

Властивість *Spacing* задає число пікселів, що розділяють зображення та надпис на поверхні кнопки (*Spacing* =4 (по замовчуванню), *Spacing* =0, тоді зображення та надпис будуть розміщені один біля одного, *Spacing* =-1, тоді текст з'явиться по середині між зображенням та краєм кнопки).

Властивість кнопки *Kind* визначає тип кнопки (*bkCustom*, *bkOK*, *bkCancel*, *bkHelp*, *bkNo*, *bkHelp*, *bkAbort*, *bkRetry*, *bkIgnore*, *bkAll*). В

даних типах вже зроблені відповідні надписи, введені піктограми, задані ще деякі властивості.

### Компонента **TRadioButton**

#### *Призначення:*

Залежний перемикач (радіокнопка). Використовується у сукупності з іншими такими ж перемикачами для вибору однієї альтернативи з декількох взаємовиключаючих альтернатив.



#### *Основні властивості:*

Радіокнопки утворюють групи взаємопов'язаних індикаторів, з яких можна вибрати лише один. Вони використовуються для вибору користувачем однієї з декількох взаємовиключаючих альтернатив. При використанні радіокнопок для відображення аналогічних даних, управління кнопками здійснюється програмно.

Властивість **Caption** містить надпис, що з'являється біля кнопки. Значення властивості **Alignment** визначає, з якої сторони від кнопки з'явиться надпис: **taLeftJustify** – зліва, **taRightJustify** – справа (по замовчуванню).

Властивість **Checked** визначає, чи вибрана дана кнопка користувачем, чи ні. Під час проектування можна встановити в **true** значення **Checked** тільки у однієї кнопки з групи.

Радіокнопки **RadioButton** можуть розміщуватися не тільки на панелі **RadioGroup**, **GroupBox**, але і в будь-якій панелі іншого типу, а також безпосередньо на формі. Група взаємопов'язаних кнопок в цих випадках визначається тією віконною компонентою, що містить кнопки. Зокрема, для радіокнопок, що розміщені на формі, контейнером є сама форма.

### Компонента **TRadioGroup**

#### *Призначення:*

Група регулярно розміщених радіокнопок, з яких у будь-який момент часу може бути увімкнена тільки одна. **TRadioGroup** є комбінацією групового вікна **TGroupBox** з набором радіокнопок **TRadioButton**.

#### *Основні властивості:*

Радіокнопки утворюють групи взаємопов'язаних індикаторів, з яких можна вибрати лише один. Вони використовуються для вибору користувачем однієї з декількох взаємовиключаючих альтернатив. При використанні радіокнопок для відображення аналогічних даних, управління кнопками здійснюється програмно.

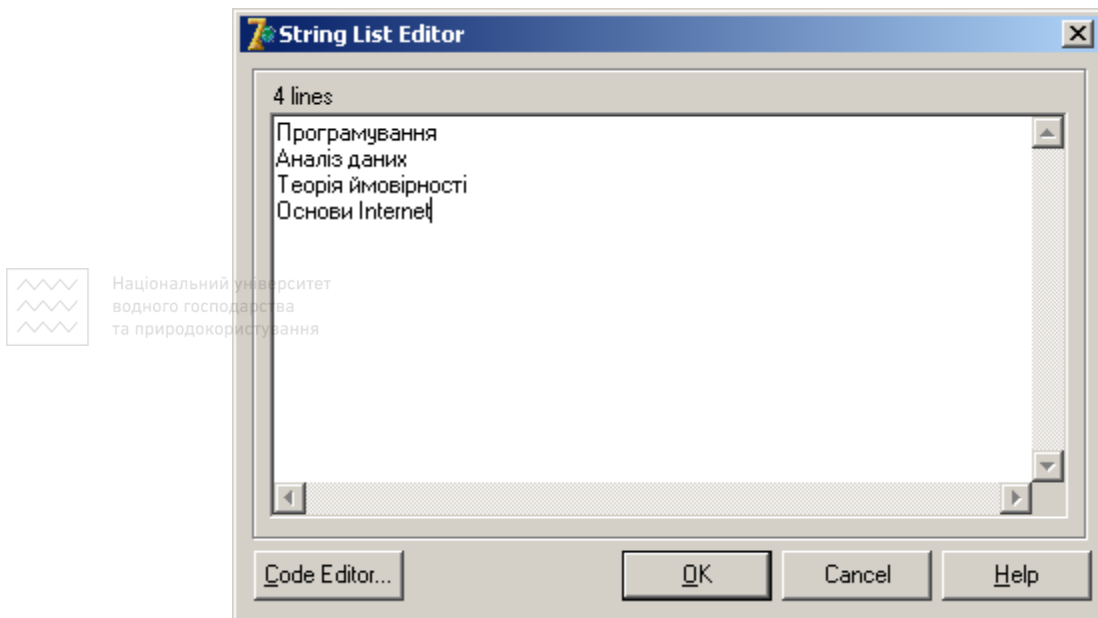


Рис.5.8. Вікно редактора стрічок

Панель групи радіокнопок **TRadioGroup** може містити регулярно розташовані стовпчиками та рядками радіокнопки. Надпис у лівому верхньому куті панелі визначається властивістю **Caption**. А надписи кнопок та їх кількість визначаються властивістю **Items**, що має тип **TStrings**. Натиснувши на кнопці з багатокрапкою біля цієї властивості у вікні Інспектора Об'єктів, потрапимо у редактор списку стрічок (Рис. 8.). В ньому можна занести надписи, які потрібно відображати біля кнопок, по одній в стрічці. Кількість стрічок в редакторі відповідає кількості кнопок на панелі (Рис.5.9.).

Властивість **Columns** задає кількість стовпчиків, в яких будуть розміщені кнопки (по замовчуванню **Columns = 1**, тобто кнопки розміщуються одна під одною).

Властивість **ItemIndex** показує індекс вибраної кнопки, таким чином дозволяє визначити, яку з кнопок вибрав користувач (індекси починаються з 0, по замовчуванню **ItemIndex = -1**, що говорить про відсутність вибраної кнопки). Значення **ItemIndex** можна задавати програмним чином.



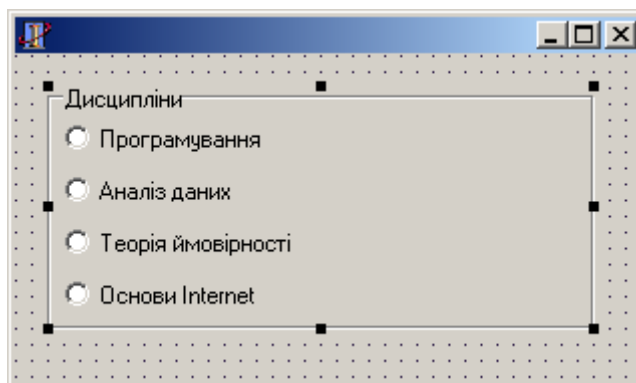


Рис.5.9. Кількість кнопок на панелі відповідає кількості стрічок в редакторі

**RadioGroup** використовується, якщо надписи кнопок мають приблизно однакову довжину (оскільки при розміщенні кнопок іде орієнтація на надпис максимально довжини), або наприклад, якщо кількість кнопок в кожному із стовпчиків однакова.

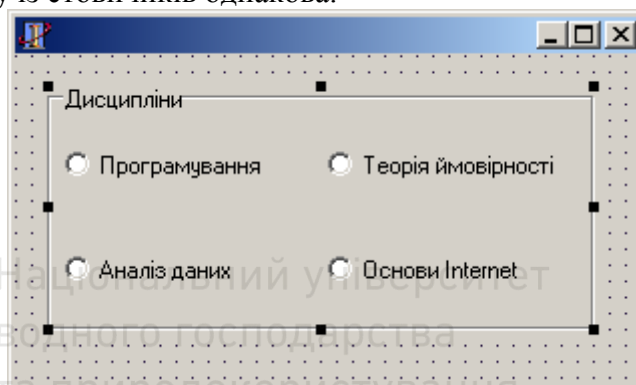


Рис.5.10. Розміщення радіо кнопок, якщо властивість *Columns=2*

### Компонента **TGroupBox**

#### Призначення:

Контейнер (панель), що об'єднує групу пов'язаних органів управління, таких як радіокнопки, незалежні індикатори.

#### Основні властивості:

Аналогічно до компоненти **TRadioButton** компонента **TGroupBox** має властивість **Caption**, що визначає надпис у лівому верхньому куті панелі. На панелі **TGroupBox** можна розмістити радіокнопки в будь-якому порядку (Рис.5.11).

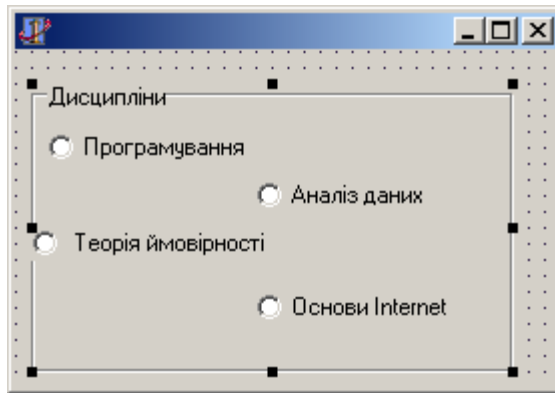


Рис.5.11. Панель *TGroupBox*

### Компонента *TCheckBox*

#### Призначення:

Індикатор або незалежний перемикач. Використовується для включення та виключення певної опції (властивості).

#### Основні властивості:

Аналогічно до радіокнопки, в індикаторі *CheckBox* надпис задається властивістю *Caption*, а його розташування по відношенню до індикатора – властивістю *Alignment*.

Серед основних властивостей компоненти можна виділити властивість *State*, яка може приймати наступні значення (Рис.12):

- 1) *cbChecked* – поява чорного прапорця;
- 2) *cbGrayed* – поява сірого прапорця, або сіре вікно індикатора (даний стан допускається, якщо властивість *AllowGrayed* приймає значення *true*);
- 3) *cbUnchecked* – пусте вікно індикатора.

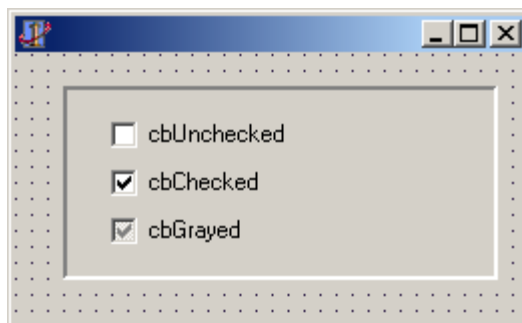


Рис.5.12. Незалежні перемикачі різних стилів

Властивості *State* та *AllowGrayed* можна встановлювати під час проектування або програмно під час виконання.

Перевірку стану індикатора можна також за допомогою властивості *Checked*. Властивість *Checked* приймає значення *true*, то індикатор

вибраний, тобто *State = cbChecked*, іншому випадку, коли *Checked* дорівнює *false*, то *State* приймає значення *cbUnchecked* або *cbGrayed*.

### Компонента TShape

*Призначення:* Використовується для малювання фігур.

*Основні властивості:*

Компоненту *TShape* тільки умовно можна віднести до засобів відображення графічної інформації, оскільки дана компонента – це просто різні геометричні фігури, які відповідним чином заштриховані.



Властивість *Shape* (форма), яка може приймати наступні значення (Рис.5.13):

- 1) *stCircle* – круг;
- 2) *stEllipse* – еліпс;
- 3) *stRectangle* – прямокутник;
- 4) *stSquare* – квадрат;
- 5) *stRoundRect* – прямокутник із заокругленими кутами;
- 6) *stRoundSquare* – квадрат із заокругленими кутами.

Властивість *Brush* (кисть). Дана властивість є об'єктом типу *TBrush*, що має ряд підвластивостей, зокрема: колір (*Brush.Color*) та стиль (*Brush.Style*) заливки фігури. Властивість *Pen* (перо) визначає стиль ліній.

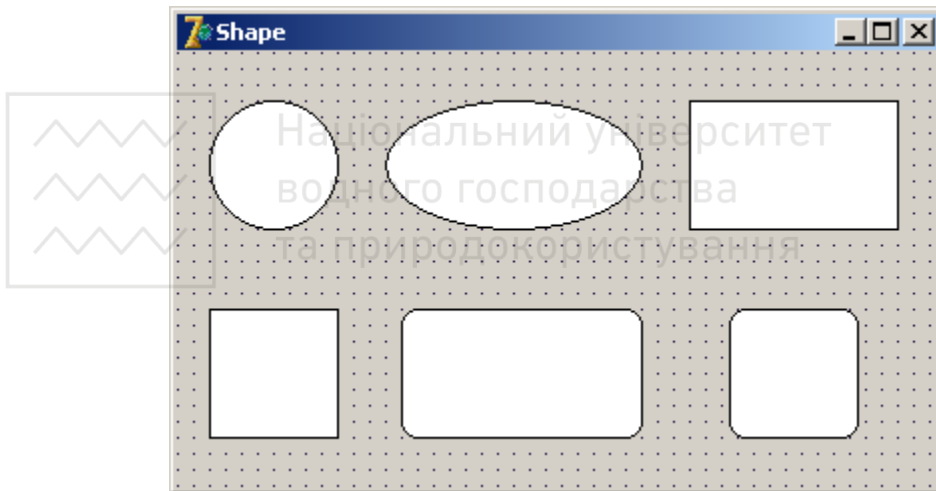


Рис.5.13. Форми компоненти *Shape*

### Компонента TLabelEdit

*Призначення:*

Вікно редагування із приєднаною до нього міткою. Тобто, дана компонента є комбінацією *TEdit* та *TLabel*.

*Основні властивості:*

Властивість *EditLabel* є об'єктом класу *TBoundLabel*, що має властивості, які притаманні мітці *TLabel*: *Caption*, *Color*, *Font*, *Layout*,

*ShowAccelChar*, *WordWrap*, *Canvas*. Властивість *LabelPosition* компоненти *TLabelEdit* вказує з якої сторони розміщена (*lpAbove* – зверху, вирівнювання по лівому краю, *lpBelow* – знизу, вирівнювання по лівому краю, *lpLeft* – зліва, *lpRight* – справа).

## Приклади розробки Delphi-додатків:

**Приклад 5.3** В середовищі програмування *Delphi* розробити програму обчислення значення функції:  $y = \cos^2(2x) - \sin x$  при заданому значенні аргументу  $x$ . Залежно від результату забезпечити зміну кольору вікна для виведення результату за наступним правилом:

$$\text{Колір} = \begin{cases} \text{жовтий,} & \text{якщо } y > 0; \\ \text{блакитний,} & \text{якщо } y < 0; \\ \text{червоний,} & \text{якщо } y = 0. \end{cases}$$

**Етап проектування** інтерфейсу проекту:

Під час проектування (рис.5.14) використаємо компоненти *TLabelEdit*, *TLabel*, *TImage* та *TButton*:

*Label1.Caption:= 'Функція';*

*LabeledEdit1.LabeledEdit.Caption:= 'Введіть x';*

*LabeledEdit2.LabeledEdit.Caption:= 'Значення функції y';*

*Button1.Caption:= 'Обчислення';*

*Button2.Caption:= 'Вихід';*

*Image.Picture:= '2.bmp'*

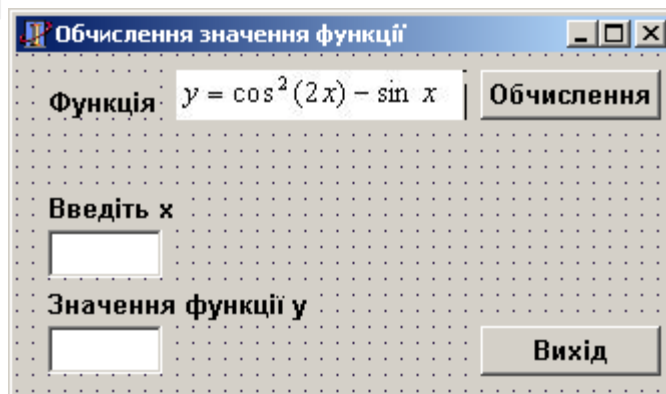


Рис.5.14. Вікно форми проекту прикладу 1.3

**Етап програмування** пректу:

Для введеного користувачем значенням  $x$  при натисканні на кнопку „Обчислення” підраховується значення функції  $y$ . Процедура обробки події, яка виникає при натисканні на кнопку „Обчислення” матиме вигляд:

```
procedure TForm1.Button1Click(Sender: TObject);
  var x,y:real;
begin
  x:=StrToFloat(LabeledEdit1.Text);
  y:=sqr(cos(2*x))-sin(x);
  if y>0 then
    LabeledEdit2.Color:=clyellow
  else if y<0 LabeledEdit2.Color:=clblue;
  else LabeledEdit2.Color:=clred;
  LabeledEdit2.Text:=FloatToStrF(y,ffFixed,5,3);
end;
```

**Результат виконання** проекту:

При виконанні програми водиться значення виразу та визначається колір компоненти *Edit1* відповідно до результату (рис.5.15).

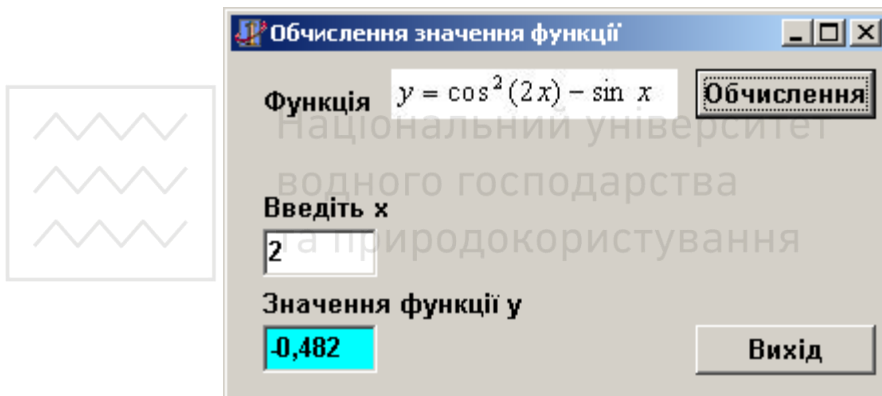


Рис.5.15. Вікно форми виконання проекту прикладу 5.3

#### **Приклад 5.4**

В середовищі програмування *Delphi* розробити програму для визначення, чи є число парним, чи непарним. Для введення числа використати компоненту *TEdit*. Якщо число є парним, то намалювати червоний квадрат, якщо непарне, то намалювати жовте коло.

**Етап проектування** інтерфейсу проекту:

Під час проектування (рис.5.16) використаємо компоненти класів *TEdit*, *TLabel*, *TShape* та *TButton*:

```

Label1.Caption:= 'Введіть ціле число';
Label2.Caption:= ' ';
Button1.Caption:='Перевірка на парність';
Shape1.Shape:='stSquare';

```

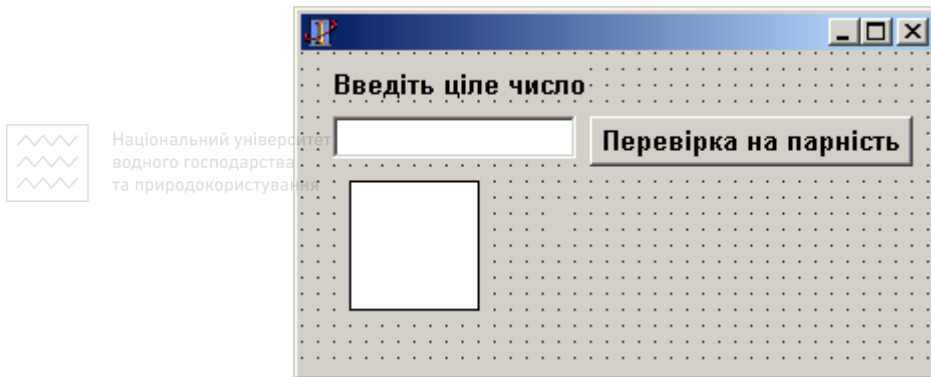


Рис.5.16. Вікно форми проекту прикладу 5.4

**Етап програмування** проекту:

Процедура обробки події, яка виникає при натисканні на кнопку „Перевірка на парність” матиме вигляд:

```

procedure TForm1.Button1Click(Sender: TObject);
var x:integer;
begin
if Edit1.Text="" then Showmessage('Введіть число') else
begin
Shape1.Visible:=true;
x:=StrToInt(Edit1.Text);

if x mod 2 = 0 then
begin
Shape1.Shape:=stsquare;
Shape1.Brush.Color:=clred;
Label2.Caption:='Число парне'
end
else

begin
Shape1.Shape:=stcircle;
Shape1.Brush.Color:=clyellow;
Label2.Caption:='Число непарне';
end;

```

```
end;  
end;
```

При введенні нового значення наступна процедура обробки події для компоненти *Edit1* забезпечує знищення попередніх результатів:

```
procedure TForm1.Edit1Change(Sender: TObject);  
begin
```



Національний університет  
водного господарства  
та природокористування

```
Shape1.Visible:=false;  
Label2.Caption:=' ';
```

```
end;
```

*Результат виконання проекту:*

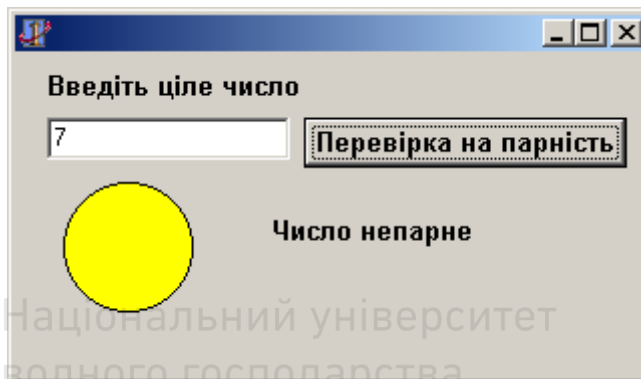


Рис.5.17. Вікно форми виконання проекту, якщо введене значення непарне

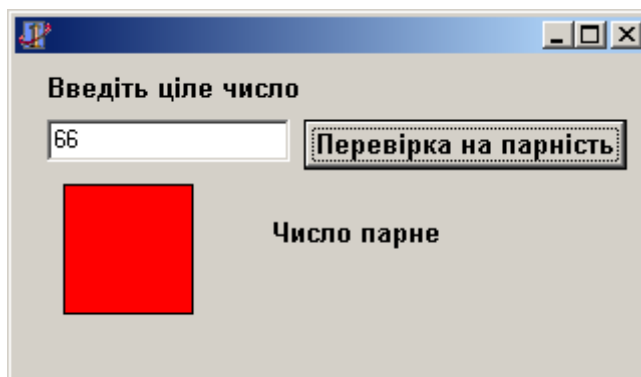


Рис.5.18. Вікно форми виконання проекту, якщо введене значення парне

## Завдання для виконання лабораторної роботи:

### Завдання 1.

В середовищі програмування **Delphi** розробити програму обчислення значення функції при заданому значенні аргументу.

#### Вимоги до проекту:

- значення змінної вводиться на формі (використати при введенні компоненти **TEdit** або **TLabelledEdit**);
- значення функції виводиться на формі (використати при виведенні компоненти **TEdit** або **TLabelledEdit**);
- для текстових підписів використати компоненту **TLabel**;
- обчислення функції має відбуватись при натисканні кнопки миші на командній кнопці (використати компоненту **TButton** або **TBitBtn**);
- на формі має бути наявний незалежний перемикач (**TCheckBox**), який при включеному стані, залежно від результату, має забезпечити колір форми за наступним правилом:

$$\text{Колір форми} = \begin{cases} \text{зелений,} & \text{якщо } y > 0; \\ \text{синій,} & \text{якщо } y < 0; \\ \text{червоний,} & \text{якщо } y = 0; \end{cases}$$

- забезпечити правильність введення даних;
- при подвійному натисканні мишею на формі забезпечити очищення полів для введення та виведення даних.

#### Варіанти завдань:

- $y = \cos^3(2x) - \log_2(x^2 + 2)$ ;
- $y = \cos^2(2x) - \sin x$ ;
- $y = \sin^3(3x) - \cos x$ ;
- $y = \sqrt{x^2 + 3} - 2\log_5(x^2 + 1)$ ;
- $y = \sqrt{x^3 + 3x} + 2\sin x$ ;
- $y = \sqrt[3]{x^4 + 3x^2} - 2\sin x$ ;
- $y = (x^2 - 2x)^3 - 4\log_3(x^4 + 1)$ ;
- $y = 2^{2x} - 4\cos x$ ;
- $y = \sin^{\frac{2}{5}}(3x) + \cos x$ ;
- $y = \sqrt[5]{x^5 + 3x^3} + 2\sin x$ ;
- $y = \sqrt[3]{x^5 + 5x^4} - 4x^{\frac{2}{3}}$ ;



$$12. y = \sqrt{\ln^2(\sin x + 2) + x^2};$$

$$13. y = \sin 3x + \sin^3 x - \operatorname{tg}^2 \frac{x}{2};$$

$$14. y = 2^{\cos x} + \sin x \sqrt{1 + 3 \operatorname{ctg} x};$$

$$15. y = \sqrt{\frac{\operatorname{tg} x}{3 + \operatorname{tg} x}} - \log_{\cos x} 2;$$

$$16. y = 3(\log_2 \sin x)^2 + \log_2(1 - \cos 2x);$$

$$17. y = \frac{(\sin x + \cos x)^2 - 2 \sin^2 x}{1 + \operatorname{ctg}^2 x} - \sin^{-1} 4x;$$

$$18. y = \sqrt{x + 2\sqrt{\cos x - \frac{1}{2}x^6}} + 5^{3x};$$

$$19. y = \frac{\sin x}{1 - \lg(x^2 - 1)} + \log_3 \sqrt{x^2 - 8};$$

$$20. y = \operatorname{tg}^2 x + \sqrt{(x + 2)^2 - 8x};$$

$$21. y = \log_{\cos x} \sin x + \log_{\sqrt{2} \sin x}(1 + \cos x);$$

$$22. y = 2^{\cos x} + \frac{x^4 + 1}{x^2 \sin^2 x} + x^{1 + \lg x};$$

$$23. y = \sqrt[4]{|x - 3|^{x+1}} - \log_{1+x} \cos x;$$

$$24. y = \frac{2x}{1 - x^2} + \sqrt{x(x + 3)};$$

$$25. y = (\cos^2(2x) - \sin x) \sqrt{x^2 - 2}.$$

### Завдання 2.

В середовищі програмування **Delphi** розробити віконний додаток для обчислення значення арифметичного виразу.

### Вимоги до проекту:

- значення змінних вводиться на формі (використати при введенні компоненту **TEdit** або **TLabelledEdit**);
- значення функції виводиться на формі (використати при виведенні компоненти **TEdit** або **TLabelledEdit**);
- для текстових підписів використати компоненту **TLabel**;
- обчислення функції має відбуватись при одноразовій фіксації мишкою на командній кнопці (використати компоненту **TButton** або **TBitBtn**);
- забезпечити правильність введення даних;

- на формі має бути група залежних перемикачів (використати компоненту **TRadioGroup**), яка б забезпечувала зміну кольору компоненти **TEdit**, в якій виводиться значення функції;
- на формі має бути незалежний перемикач (компонента **TCheckBox**), якщо він ввімкнений, то всі аргументи функції беруться по модулю;
- передбачити вивід відповідного повідомлення, якщо введені значення аргументів виходять за межі області визначення функції, або якщо зроблена спроба обчислити значення функції при неповних даних (**messagebox**, **showmessage**).



### Варіанти завдань:

$$1. z = \frac{|x-1| + e^{-y}}{12.34 - \lg \sqrt{|x|}}, \text{ якщо } y = 2a\sqrt[3]{a+b}, x = \operatorname{arccctg} \frac{e^a + e^b}{\sqrt{a+e}};$$

$$a = 1,75, b = 0,4;$$

$$2. m = \lg^2 |y - 5.5| + \sin^2 \frac{y}{4}, \text{ якщо } y = \ln|\pi - x| + \lg \left| \frac{\pi}{x} \right|,$$

$$x = \sqrt{|\sin e^2 + 3.41|};$$

$$3. z = \operatorname{arctg}(\sin^2 x + \operatorname{tg}^3 y), \text{ якщо } y = \sin^2(\alpha - \beta)^3,$$

$$x = \ln|\alpha + 2,3| - \lg|\beta - 3,2|, \alpha = 15,3, \beta = -0,012;$$

$$4. z = \sqrt{|\sin^3(x-1) + \cos y|}, \text{ якщо } y = 23,41, x = \log_y \left| \frac{\pi}{e} + 1 \right| + \operatorname{tg} y;$$

$$5. z = \sqrt[3]{|\ln|x|^{-1} - \sqrt{|x+1|}|}, \text{ якщо, } x = \frac{\arcsin \omega^{-1} + \ln|\omega|}{(-2)e^{-\omega}}, \omega = 3,47;$$

$$6. z = \sqrt[3]{|\pi - y|} + \sin^2 \pi x + 1,67, \text{ якщо } y = \operatorname{tg}^4(\beta - 1)^2 - 0,035,$$

$$x = \operatorname{ctg} \frac{\alpha - 1}{e} + 2^{\frac{\alpha+1}{2}}, \alpha = 4,4, \beta = 1,87;$$

$$7. z = \ln \left| \frac{x\sqrt{x} + \cos^3 y^2}{1,604 - \operatorname{arccctg} y} \right|^{2,1}, \text{ якщо } x = \log_{|a|} b + \frac{(a-b)^2}{b} \cdot e^{-a},$$

$$y = \sqrt[3]{|\cos a^2 + ab + 0,06|}, a = -0,2, b = 7;$$

$$8. z = c \cdot e^{-2,5x+y^2} - \sqrt[3]{cx}, \text{ якщо } x = \frac{\lg|c + \alpha|}{\operatorname{arccctg} \frac{\pi}{\alpha}} + 0,17,$$

$$y = \frac{\sin^2 \frac{\alpha^3}{2} - \operatorname{ctg} \frac{c}{4}}{\ln|\alpha| + \ln c^2};$$

$$c = 4,5, \alpha = 2,01$$

9.  $p = \frac{e^{-xy} + 17,4}{\sqrt[3]{\sin^2 xy}}$ , якщо  $x = (a^2 + b^2)^{-4,1}$ ,  $y = \operatorname{arctg}^3 \frac{1}{b}$ ,

$$a = -2,0004, b = 0,87;$$

10.  $\varphi = \arccos\left(\frac{x^2}{0,13}\right)^{-1} + \ln|y^{-1}|$ , якщо  $x = \sqrt{(k+6,1)^3}$ ,

$$y = \ln k^4 + \lg m^{-6}, \quad k = 14, m = 0,42;$$

11.  $t = \ln|m-y| + \cos^3 my$ , якщо  $m = \sqrt{|x+a|} + 17,14 \cdot \lg \frac{\pi}{3}$ ,

$$y = a \cdot \sqrt[3]{\sin^4 x^3} + 12,47, \quad x = 3,4, a = -1,17.$$

12.  $y = \frac{x^2 - z^2}{\lg|x-7|}$ , якщо  $x = \frac{\sin^2 a^3 - \arcsin \frac{1}{b}}{\ln|a+b|-1}$ ,  $z = \sqrt{\left|\frac{a+b}{ab}\right|} + \pi$ ,

$$a = 3,5, b = -2,16;$$

13.  $y = -\sqrt{|\lg x - \ln z|} + 1,31$ , якщо  $x = \frac{e^{-2,5a} + \sin^2 a^3}{2 \lg|ba|}$ ,

$$z = \frac{\operatorname{arctg}^3(b-a) + \sqrt[3]{ab}}{1 + \log_b a}, \quad a = 0,6, b = 3,12;$$

14.  $r = \operatorname{ctg} \frac{x+y}{(x-y)^2} + 1,3$ , якщо  $x = \sin^4 e^{-b} + |ab|$ ,  $y = \ln|a-b| + \lg \frac{\pi}{a}$ ,

$$a = 1,77, b = -0,62;$$

15.  $\alpha = \frac{e^{-3,5|x|+\sqrt{\pi}}}{\operatorname{arctg}^3(y-1)}$ , якщо  $x = a + \cos \frac{\pi}{b}$ ,  $y = \ln\left|\frac{\pi}{16} - b\right|$ ,

$$a = 0,5, b = 1,4 \cdot 10^3;$$

16.  $\gamma = \operatorname{arctg} \frac{x+1}{y-2} + \lg|k+x|$ , якщо  $x = \sqrt{|m+n|^3} + 17,14mn$ ,

$$y = \sqrt[3]{|km-3|} + \frac{\pi}{6}, \quad m = 3, n = 2,2, k = 0,801;$$

17.  $t = \frac{x^2 - y^3}{e^{-(x+y)}}$ , якщо  $x = \sqrt{8,67 + e^y + |y|}$ ,  $y = \sqrt[3]{\lg e} + \sqrt[5]{|\cos e - 2|}$ ;

$$18. x = \arccos \frac{\pi - z}{3} + e, \text{ якщо } z = \sqrt{|y + \sin^2 y|}, y = 0,3 \log_5 e^{-2,3};$$

$$19. a = \gamma \cdot \sqrt[3]{y + 0,01} + \sin^2 \pi x, \text{ якщо } y = \operatorname{tg}^4(x - 1), \\ y = \lg|\gamma + 6,6| + 0,77, \quad \gamma = -3,41;$$

$$20. z = \lg|x + 1| - \ln^3|2^x - 1|, \text{ якщо } x = e^{ky-5,1} + \cos^2 ky, \quad y = \sqrt{|k - e|}, \\ k = 2,26;$$

$$21. t = \frac{\ln|m - y| + \cos^3 my}{\sqrt{|m + y|^3 + 17,14}}, \text{ якщо } y = (2m)^{-e} + \operatorname{arctg}\sqrt{e}, m = 2,7 \cdot 10^{-3};$$

$$22. f = \frac{x^e - e^{-x} + 0,12}{\sqrt{|\sin(y - 1)|}}, \text{ якщо } x = e^{-\pi} + \pi^{-e}, \quad y = \lg a^3 - \operatorname{arctg} a, \\ a = 6,45;$$

$$23. j = \log_{\pi} |x|^{-m} + \left| \frac{\pi}{5} - y \right|, \text{ якщо } x = \operatorname{arctg} \frac{5,4}{m} + mn, \\ y = \sqrt{|m - 3|} + \ln n^2, m = -2, n = 3,87;$$

$$24. \varepsilon = e^2 \cdot \log_2 x^4 - \sqrt{|y + 1|}, \text{ якщо } x = 21,4(a - 0,5)^2 + \cos \frac{\pi}{b},$$

$$y = \ln \left| \frac{\pi}{a} - b \right| + \operatorname{tg}^2 b^3, a = 0,7, b = -4;$$

$$25. b = (\beta + |z|)^{-e} + \sqrt[3]{|z| + 0,1}, \text{ якщо } \beta = e^{k-5,1} + \lg|k + x|, \\ z = \ln^3|2^x - 1| - 12,47, x = 0,03, k = 4.$$

### Завдання 3.

В середовищі програмування **Delphi** розробити віконні додатки із використанням основних компонент палітр **Standard** та **Additional**. Розробити зручний інтерфейс та забезпечити правильність введення даних.

### Варіанти завдань:

1. Вводиться три числа. Визначити, чи утворюють дані числа арифметичну прогресію. За позитивність чи негативність відповіді відповідає круг, розміщений на формі (компонента TShape). Якщо так – круг набуває зеленого кольору, ні – червоного.
2. На формі розміщені три фігури: круг, квадрат та трикутник. Якщо натискаємо мишкою на круг, то трикутник змінює свій колір, а квадрат – ні. Якщо на квадрат – то всі фігури змінюють свій колір. При натисканні на трикутник – круг і квадрат змінюють свій колір.

3. Написати програму для визначення, чи існує такий трикутник  $\Delta ABC$  із заданими сторонами. Значення сторін  $AB$ ,  $AC$ ,  $BC$  задаються із використанням компонент *TEdit* або *TLabelEdit*. Якщо трикутник існує, то повинно з'явитися діалогове вікно із відповідним текстом, в протилежному випадку відбувається закриття форми.
4. На формі розміщено три круги: червоний, жовтий і зелений (вертикально). При натисканні мишкою на круги має з'являтися відповідний надпис: “стій”, “зачекай”, “іди”.
5. Розробити програму, яка дозволяє змінювати колір форми та колір шрифтів. Для цього розмістити на формі дві групи незалежних перемикачів, які відповідатимуть за зміну кольору та зміну шрифту.
6. Організувати малювання на формі фігур (коло, квадрат і ін.), зафарбованих певним кольором за вибором користувача. Для вибору фігур та кольору використати компоненту *TRadioGroup*, для відображення фігур – компоненту *TShape*.
7. На формі вводиться три числа. Визначити, чи утворюють дані числа геометричну прогресію. За позитивність чи негативність відповіді відповідає круг, розміщений на формі (компонент *TShape*). Якщо так – круг набуває жовтого кольору, ні – синього.
8. Вводиться послідовність чисел через кому, яка закінчується крапкою. На формі встановлено незалежний перемикач. Якщо він ввімкнений, то шукаються кількість тільки парних чисел, якщо ні – непарних.
9. На формі вводиться п'ять чисел. Визначити, чи утворюють дані числа послідовність Фібоначчі. За позитивність чи негативність відповіді відповідає квадрат на формі (компонента *TShape*). Якщо так – квадрат набуває зеленого кольору, ні – червоного.
10. Вводиться послідовність чисел через кому. Визначити довжину найдовшої підпослідовності однакових чисел. Якщо довжина менша або рівна 2, на формі з'являється червоний круг.
11. На формі вводиться число. Якщо введене число більше нуля, то відбувається закриття додатку, якщо нуль – виводиться повідомлення про повторне введення числа. У випадку, коли число більше одиниці, форма зафарбовується у блакитний колір.
12. На формі вводяться три числа (компонента *TEdit*). Якщо всі три числа рівні, то змінити колір компоненти *TEdit* на зелений, в протилежному випадку – на червоний. Перевірку на рівність здійснювати після натискання командної кнопки (компонент *TButton*).
13. На формі вводяться три числа (компонента - *TEdit*). Змінити колір компоненти, в якій знаходиться найбільше число, на зелений, середнє число – на жовтий, найменше число – на червоний колір. У випадку рівності будь-яких чисел – змінити на синій колір. Перевірка здійснюється після натискання командної кнопки (компонента *TButton*).

14. На формі вводяться п'ять чисел. Якщо більшість з них парні, то вивести на форму зелений квадрат, в протилежному випадку – червоний.
15. На формі вводяться п'ять чисел. Якщо більшість з них непарні, то вивести на форму зелений круг, в протилежному випадку – червоний круг.
16. На формі розміщені 3 фігури: круг, квадрат і трикутник. Написати програму, яка б змінювала колір будь-яких двох при натисканні на будь-яку третю фігуру.
17. На формі розміщений круг. Він змінює своє забарвлення наступним чином: введене число є простим – зелений колір, інший варіант – червоний. Якщо ввімкнений незалежний перемикач, то відбувається перевірка на парність та непарність введеного числа. Про це повідомляє відповідний напис на формі.
18. На форму вводиться ціле число. Підрахувати кількість цифр в ньому і вивести її на форму. Якщо дана кількість парна - то змінити колір фону форми на жовтий, в протилежному випадку – на зелений.
19. На форму вводиться ціле число. Перевірити, чи є воно симетричним (яке читається зліва направо і справа наліво однаково). Якщо так – то вивести зелений круг, в протилежному випадку – синій. Також підрахувати кількість цифр у ньому.
20. На форму вводиться п'ять чисел. Якщо дана послідовність є строго спадною, то змінити колір фону форми на салатовий, в протилежному випадку ніяких дій не відбувається.
21. На форму вводиться п'ять чисел. Визначити, чи дана послідовність є строго зростаючою. Якщо так, то на формі з'являється круг, зафарбований в зелений колір, якщо ні – то квадрат червоного кольору.
22. На формі вводяться два числа. Якщо остача від ділення першого на друге дорівнює 1, то на формі відобразити червоне коло, якщо дорівнює 2 – синій квадрат. У всіх інших випадках ніяких дій не відбувається.
23. На формі задається натуральне число. Знайти середнє арифметичне цифр у записі цього числа. Якщо отриманий результат більше 5, то зобразити круг червоного кольору, в протилежному випадку – зеленого кольору.
24. На формі вводяться три числа  $a, b, c$ . Якщо число  $c$  попадає в проміжок  $[a; b]$ , то зробити колір форми зеленим та зобразити на ній червоний квадрат. Якщо число  $c$  не попадає у відрізок  $[a; b]$ , то відобразити на формі нове вікно TEdit, куди вивести різницю  $c-a$ .
25. На формі вводяться чотири додатних цілих числа. Користувач випадково клацає мишкою на будь-якому з них. Якщо це число є середнім арифметичним інших трьох, то з'являється зелений трикутник, якщо більше за середнє арифметичне – то червоне коло,



якщо менше – то квадрат, сторона якого дорівнює сумі всіх чотирьох чисел.

### Контрольні запитання:

1. Перерахуйте склад вікна *Delphi*.
2. Що являє собою вікно редактора коду?
3. Перерахувати способи створення нового вікна форми.
4. Перерахувати способи переходу з вікна форми у вікно Редактора коду.
5. Які можливості підтримує вікно Редактора коду?
6. Назвіть основні елементи Інспектора об'єктів.
7. Які групи палітри компонент Ви знаєте?



Національний  
водного господарства  
та природоохорони

### Лабораторна робота №3

**Тема: Розробка циклічних програм.**

#### **Мета роботи:**

- Навчитись використовувати візуальні компоненти сторінки палітри *Samples: TSpinEdit*, сторінки *Standard: TEdit, TButton, TLabel, Tmemo*, сторінки *Additional: TBitBtn, TLabeledEdit, TStringGrid* та сторінки *Win 32: TRichEdit, TUpDown*.
- Вивчити основні властивості та методи роботи з компонентами панелі *Additional, Standard* та *Samples*.

#### **Теоретичні відомості:**

При розробці програм для даної лабораторної роботи пропонується використати наступні компоненти: *TEdit, TLabel, TButton, TBitBtn, TRadioButton, TRadioGroup, TGroupBox, TCheckBox, TShape, TLabeledEdit, TSpinEdit, TUpDown, TStringGrid, Tmemo, TRichEdit*. Нижче наведено призначення та основні властивості компонент, які не розглядалися у передніх лабораторних роботах.

#### **Компоненти TSpinEdit та TUpDown.**

**Призначення:**

**TSpinEdit** – кнопка-лічильник з вікном редагування. Дана компонента забезпечує ведення цілих чисел. Майже те саме, що і комбінація *TEdit* та *TUpDown*.

**Основні властивості:**

Компонента *TSpinEdit* являє собою поєднання *TEdit* та *TUpDown*, оформлене як окремий тип компонента.

Основна властивість компоненти *TUpDown Associate* пов'язує кнопки зі стрілочками із певною віконною компонентою, зазвичай з *TEdit*, при цьому у властивості компоненти *TUpDown Associate* вибирається вікно редагування *Edit*.

Властивість *AlignButton* компоненти *TUpDown*, яке може приймати значення *udLeft* або *udRight*, визначає, де будуть розміщуватися кнопки відносно вікна редагування, зліва чи справа.

Властивість *Orientation*, яка може приймати значення *udHorizontal* або *udVertical*, визначає горизонтальне або вертикальне розташування кнопок.

Властивість *ArrowKeys* при значенні *true* дозволяє керувати кнопками *TUpDown* за допомогою клавіш клавіатури зі стрілочками.

Властивість *Thousands* визначає наявність чи відсутність розділяючого пробілу між кожними трьома цифрами розрядів числа, що вводиться.

Властивості *Min* та *Max* компоненти *TUpDown* задають відповідно мінімальне та максимальне значення чисел.

Властивість *Increment* задає приріст числа при кожному натисканні на кнопку *TUpDown*.

Властивість *Position* є основною властивістю компоненти *TUpDown* та визначає поточне значення числа.

Властивість *Wrap* визначає, як поводить себе компонента при досягненні максимального чи мінімального значення (якщо *Wrap = false*, то при збільшенні або зменшенні числа до максимального або мінімального значення це число фіксується на певному значенні і натискання кнопки, що реалізує намагання збільшити максимальне число або зменшити мінімальне, ні до чого не призводять. Якщо *Wrap = true*, то спроба збільшити максимальне число призводить до скидання на мінімальне значення. Аналогічно, спроба зменшити мінімальне число призводить до його скидання на максимальне значення.

Якщо в компоненті *TEdit*, що пов'язується із *TUpDown*, задавати властивість *ReadOnly* рівною *false*, то користувач зможе задавати значення не користуючись кнопками зі стрілочками, що є зручним, коли необхідне значення далеке від вказаного по замовчуванню, а крок приросту *Increment* в *TUpDown* малий.

Головним недоліком компоненти *TUpDown* та відповідно пов'язаною з нею компонентою *TEdit* є можливість помилкового введення не цифр, а інших символів. Таким чином потрібно забезпечити правильність введення даних (див. Лабораторна робота №1), або використовувати компоненту *TSpinEdit*, в якій допускаються при введенні тільки числа.



водного тиску  
та природи



Властивості компоненти *TSpinEdit* схожі на розглянуті вище, мають тільки інші імена: властивість *Min*, *Max*, *Position* називаються відповідно *MinValue*, *MaxValue*, *Value*. В цілому компонента *TSpinEdit* у багатьох відношення краща, ніж просте поєднання *TUpDown* та *TEdit*, за виключенням, коли вимагаються якісь із описаних вище властивостей *TUpDown*.

### Компонента *TStringGrid*

**Призначення:**

Таблиця стрічок. Відображення тестової інформації у вигляді таблиці з можливістю переміщуватися по стрічках та стовпчиках та можливістю здійснювати вибір.

*Основні властивості:*

Компонента *TStringGrid* – це таблиця, що містить стрічки. Дані в таблиці можуть бути тільки для читання або для редагування. Таблиця може мати смуги прокрутки, можна задати заголовки стовпчиків та стрічок, що постійно наявні у вікні компоненти. Кожній лунці може бути поставлений у відповідність деякий об'єкт. Компонента *TStringGrid* призначена першу чергу для відображення таблиць текстової інформації, але в свою чергу може містити і графічну інформацію.

Основні властивості компоненти, що визначають текст, який відображається:

- ***Cells[ACol, ARow: Integer]: string*** – стрічка, що міститься в лунці з індексами стовпчика *ACol* та рядка *ARow*;
- ***Cols[Index: Integer]: TString*** – список стрічок, що містяться у стовпчику із індексом *Index*;
- ***Rows[Index: Integer]: TString*** – список стрічок, що містяться у стрічці із індексом *Index*;
- ***Objects [ACol, ARow: Integer]: TObject*** – об'єкт, що пов'язаний зі стрічкою, який міститься у лунці з індексами стовпчика *ACol* та стрічки *ARow*.

Всі ці властивості доступні під час виконання. Задавати тексти можна програмно або по окремих лунках, або відразу по стовпчиках та стрічкам за допомогою методів класу *TStrings*.

Властивість ***ColCount*** і ***RowCount*** визначають відповідно кількість стовпчиків та стрічок.

Властивості ***FixedCols*** та ***FixedRows*** – кількість фіксованих стовпчиків та стрічок, що не прокручуються. Колір фону фіксованих лунок визначається властивістю ***FixedColor***.

Властивість ***ScrollBars*** визначає наявність в таблиці смуг прокручування.

Властивість ***Options*** є множиною, що визначає багато властивостей таблиці: наявність розділяючих вертикальних та горизонтальних ліній у

фіксованих (*goFixedVertLine* та *goFixedHorzLine*) в нефіксованих (*goVertLine* та *goHorzLine*) лунках, можливість для користувача змінювати за допомогою миші розміри стовпчиків та стрічок (*goColSizing* та *goRowSizing*), переміщувати стовпчики та стрічки (*goColMoving* та *goRowMoving*), можливість редагувати вміст таблиці (*goEditing*). В основному компонента *TStringGrid* використовується для вибору користувачем певних значень, що відображені в лунках. Властивості *Col* та *Row* показують індекси стовпчика та стрічки виділеної лунки. Надається також можливість виділення користувачем множини лунок, стрічок та стовпчиків. Серед множини подій компоненти *TStringGrid* слід відмітити подію *OnSelectCell*, що виникає в момент вибору лунки користувачем.



### Компоненти *TMemo*, *TRichEdit*:

*Призначення:*

Вікно редагування багатострічкового тексту (*TMemo*), вікно редагування багатострічкового тексту у форматі *RTF* (*TRichEdit*). Використовується для введення та відображення багатострічкового тексту.

*Основні властивості:*

В компоненті *TMemo* формат (шрифт, атрибути вирівнювання) однакові для всього тексту та визначаються властивістю *Font*. Компонента *TRichEdit* працює із текстом у форматі *RTF*. При бажанні змінити атрибути тексту можна задати властивість *SelAttributes*. Дана властивість типу *TTextAttributes*, що в свою чергу має властивості: *Color* (колір), *Name* (імя шрифту), *Size* (розмір), *Style* (стиль) та ряд інших.

Наприклад, напишемо програму, яка дозволяла користувачу міняти атрибути тексту. Введемо на формі компоненти *TRichEdit*, діалог вибору шрифту *TFontDialog* та кнопку *TButton*. В процедуру обробки події *OnClick* введемо наступний текст:

```
if FontDialog1.Execute then  
    RichEdit1.SelAttributes.Assign(FontDialog1.Font);  
    RichEdit1.SetFocus;
```

Запустивши програму, можна побачити, що, виділивши потрібний фрагмент та натиснувши на кнопку, відкриється діалогове вікно „Шрифт” (Рис. 19.), де можна вибрати атрибути тексту (змінювати шрифт, розмір, колір, стиль).

Компонента *TRichEdit* також має властивість *DefAttributes* (доступна лише під час виконання), що містить атрибути по замовчуванню, які діють до того моменту, коли змінюються атрибути у властивості *SelAttributes*. Але значення атрибутів в *DefAttributes* зберігаються і в будь-який момент ці значення можуть бути методом

**Assign** присвоєні атрибутам властивості **SelAttributes** для того, щоб повернутися до попереднього стилю.

За вирівнювання, відступи та інше у межах поточного абзацу відповідає властивість **Paragraph** (доступна лише в процесі виконання додатку) типу **TParaAttributes**. Даний тип має в свою чергу ряд наступних властивостей: **Alignment** – визначає вирівнювання тексту, **FirstIndent** – визначає число пікселів відступу червоної стрічки, **Numbering** – управляє вставкою маркерів, **LeftIndent** – визначає відступ від лівого поля, **RightIndent** – визначає відступ від правого поля, **TabCount** – визначає кількість позицій табуляції (має зміст при значенні властивості компоненти **WantTabs = true**), **Tab** – значення позиції табуляції в пік селях (має зміст при значенні властивості компоненти **WantTabs = true**).

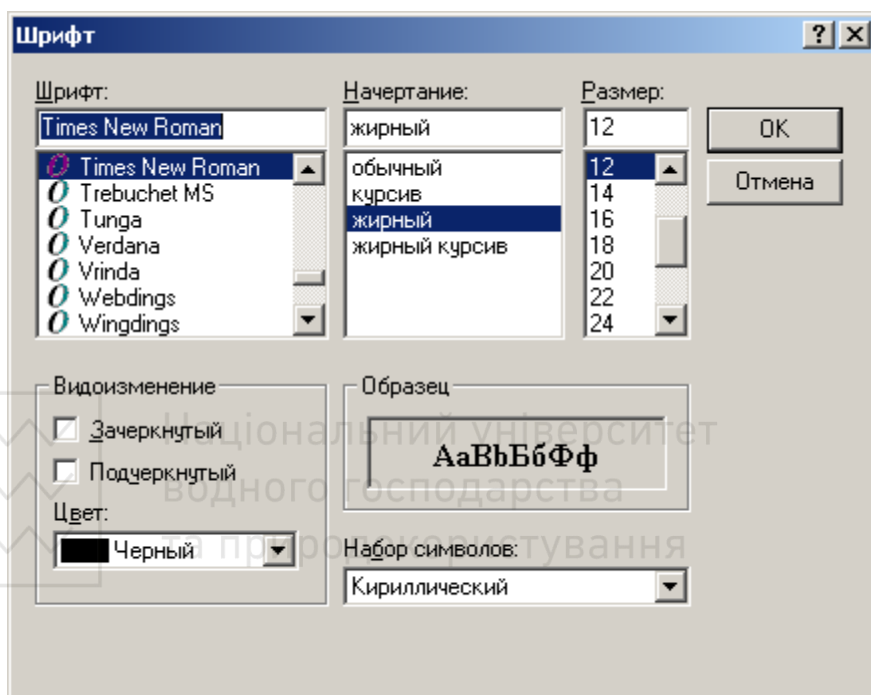


Рис.5.19. Діалогове вікно “Шрифт”

В результаті отримаємо текст з новими атрибутами (рис.5.20).

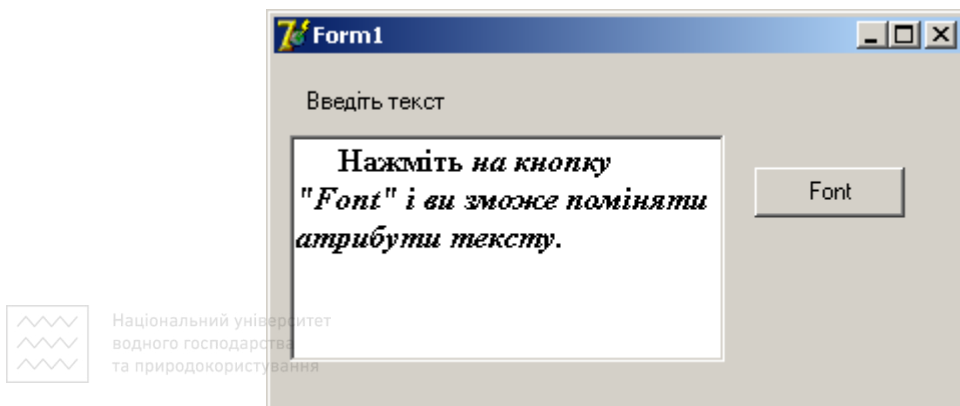


Рис.5.20. Текст з новими атрибутами

Наступні властивості притаманні як *компоненті TMemo*, так і *TRichEdit*.

Властивість *Alignment* визначає вирівнювання тексту, *Wordwrap* – допустимість перенесення довгих стрічок. Надання властивості *ReadOnly* значення *true* визначає текст тільки для читання. Властивість *MaxLength* визначає максимальну довжину тексту, що водиться (якщо *MaxLength = 0*, то довжина тексту необмежена). Властивості *WantReturns* та *WantTab* визначають допустимість вводу користувачем в текст символів переводу стрічки та табуляції. Властивість *ScrollBars* визначає наявність смуг прокрутки тексту у вікні. Основна властивість вікон *TMemo* та *TRichEdit* – *Lines*, що містить текст вікна у вигляді списку стрічок та має тип *TStrings*. Під час виконання можна заносити текст у вікно редагування за допомогою методів властивості *Lines* типу *TStrings*. Властивість *Clear* забезпечує видалення тексту із вікна.

Основними властивостями та методами *Lines* є:

- *Text* – весь текст, що представляється однією стрічкою типу *String*, включаючи роздільники;
- *Strings[Index: Integer]* – забезпечує доступ до окремої стрічки тексту (наприклад, *Memo1.Lines.Strings[0]* – це текст першої стрічки); при цьому слід зважати на значення властивості *WordWrap*);
- *Count* – вказує число стрічок в тексті;
- методи *Add* або *Append* дозволяють заносити нові стрічки в кінець файлу вікна редагування;
- *LoadFromFile* – використовується для завантаження тексту із файлу;
- *SaveToFile* – використовується для збереження у файлу.

Властивість *SelStart* компонент *TMemo* та *TRichEdit* вказує позицію курсору в тексті або початок виділеного користувачем тексту.

## Приклади розробки Delphi-додатків.

### Приклад 5.5

В середовищі **Delphi** розробити віконний додаток для табулювання значення функції  $y = \sqrt{x^3 + x + 1}$  на проміжку  $[a, b]$  із кроком  $h$ , що задається користувачем.

**Етап проектування** інтерфейсу проекту:

Під час проектування (рис.5.21) використовуємо компоненти класу **TLabelledEdit**, **TLabel**, **TButton**, **TMemo**:

**Label1.Caption:= 'Введіть проміжок [a, b]: '** ;

**Label2.Caption:= 'Введіть крок h: '** ;

**Label3.Caption:= 'Результат '** ;

**LabeledEdit1.LabeledEdit.Caption:= 'a'** ;

**LabeledEdit2.LabeledEdit.Caption:= 'b'** ;

**LabeledEdit3.LabeledEdit.Caption:= 'h'** ;

**Button1.Caption:= 'Табулювання'** .

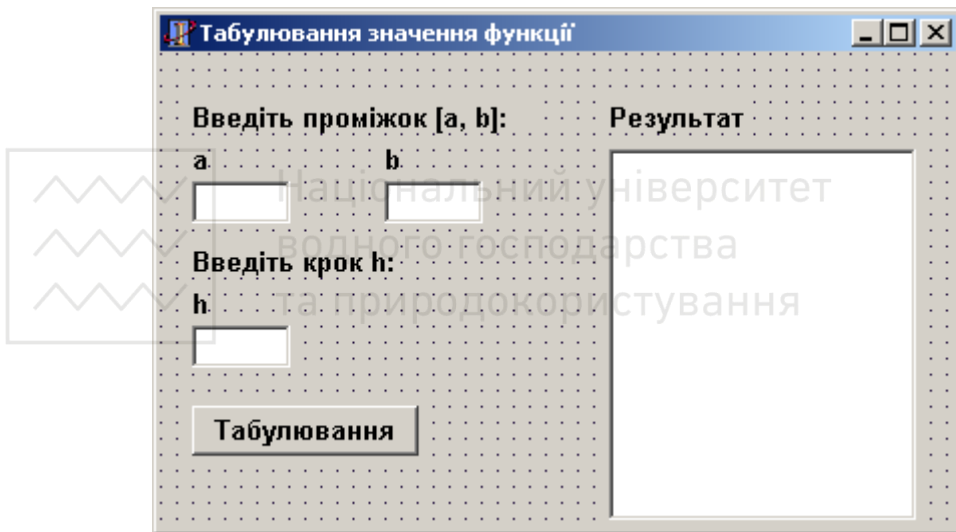


Рис.5.21. Вікно форми проекту прикладу 5.5

Для табулювання функції вводимо спочатку:

- проміжок, на якому потрібно провести табулювання;
- крок.

**Табулювання** проводиться при натисканні на кнопку „Табулювання”, результат заноситься до поля **TMemo**.

**Етап програмування** проекту:

Процедура обробки події **OnClick** матиме наступний вигляд:

```
procedure TForm1.Button1Click(Sender: TObject);
var x,y,a,b,h: real;
begin
    a:=StrToFloat(LabeledEdit1.Text);
    b:=StrToFloat(LabeledEdit2.Text);
    h:=StrToFloat(LabeledEdit3.Text);
    Memo1.Lines.Add(' x      y');
    x:=a;
    while x<=b do
    begin
        y:=sqrt(power(x,3)+x+1);
        Memo1.Lines.Add(FloatToStrF(x,ffFixed,7,2)+ ' '
            +FloatToStrF(y,ffFixed,7,2));
        x:=x+h;
    end;
end;
```

**Результат виконання** проекту:

На проміжку  $[1,2]$  з кроком  $0,09$  матимемо наступні результати, які зазначені на рис.5.22.

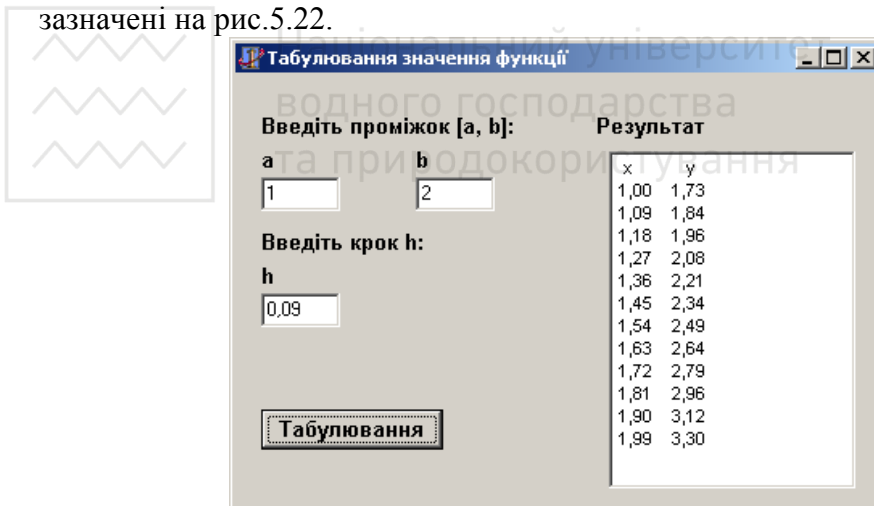


Рис 5.22. Вікно форми виконання проекту прикладу 5.5

### Приклад 5.6

В середовищі **Delphi** розробити віконний додаток для табулювання значення функції  $y = \sqrt{x^3 + x + 1}$  починаючи із заданої точки  $x_0$ , коли

заданий крок  $\Delta x$  та кількість кроків  $n$  (з використанням компоненти *TStringGrid*).

**Етап проектування** інтерфейсу проекту:

Під час проектування (рис.5.23) використаємо компоненти класу *TEdit*, *TLabel*, *TButton*, *TStringGrid* :

*Label1.Caption:= 'Введіть початкове значення: ' ;*

*Label2.Caption:= 'Введіть крок: ' ;*

*Label3.Caption:= 'Введіть кількість кроків: ' ;*

*Label4.Caption:= 'Таблювання' ;*

*Button1.Caption:= 'Таблювання' ;*

*SpinEdit1.MaxValue = 7;*

*SpinEdit1.MinValue = 1;*

*StringGrid1.ColCount = 3;*

*StringGrid1.RowCount = 2*

*StringGrid1.Options = [goFixedVertLine, goFixedHorzLine, goVertLine, goHorzLine, goRangeSelect, goEditing];*

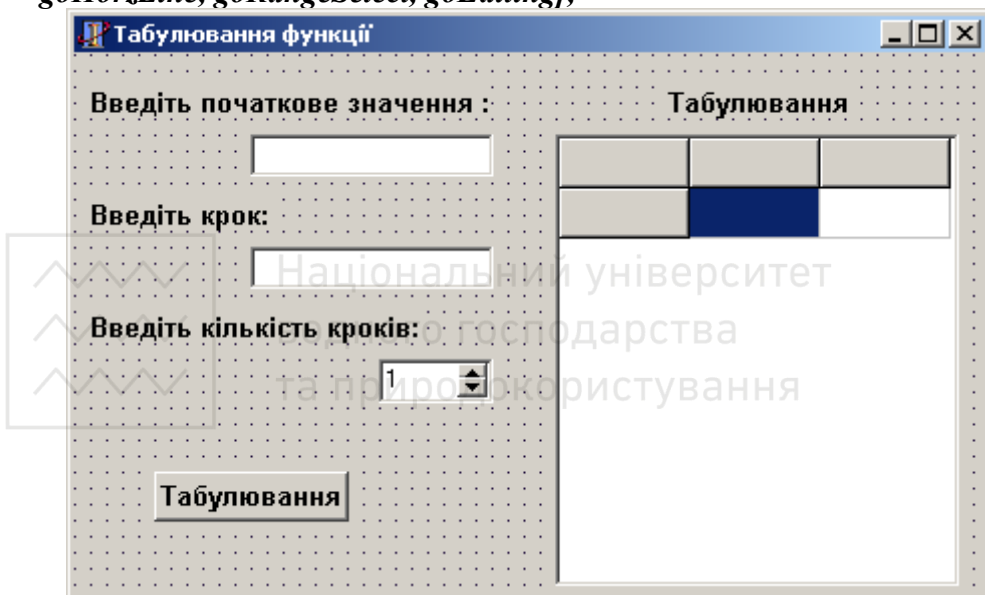


Рис.5.23. Вікно форми проекту прикладу 5.6

**Етап програмування** проекту:

Процедура обробки події *OnClick* матиме наступний вигляд:

```
procedure TForm1.Button1Click(Sender: TObject);  
var x,dx,y:real;  
    n,i:integer;
```

```

begin
  x:=StrToFloat(Edit1.Text);
  dx:=StrToFloat(Edit2.Text);
  n:=SpinEdit1.Value;
  StringGrid1.Cells[0,0]:= 'n';
  StringGrid1.Cells[1,0]:= 'x';
  StringGrid1.Cells[2,0]:= 'y';

  for i:=1 to n do
    begin
      StringGrid1.RowCount:=i+1;
      y:=sqrt(power(x,3)+x+1);
      StringGrid1.Cells[0,i]:= IntToStr(i);
      StringGrid1.Cells[1,i]:= FloatToStr(x);
      StringGrid1.Cells[2,i]:= FloatToStrF(y,ffFixed,7,3);
      x:=x+dx;
    end;
  end;
end;

```



Національний університет  
водного господарства  
та природокористування

**Результат виконання** проекту:

З кроком 0,09 матимемо результати, які зазначені на рис.5.24.

n	x	y
1	1	1,732
2	1,09	1,840
3	1,18	1,955
4	1,27	2,078
5	1,36	2,208
6	1,45	2,345
7	1,54	2,488

Рис 5.24. Вікно форми виконання проекту прикладу 5.6



## Завдання для виконання лабораторної роботи.

### Завдання 1.

В середовищі програмування **Delphi** розробити програму обчислення значення функції  $y = f(x)$  на вказаному проміжку із заданим кроком  $\Delta x$  для відповідного аргументу функції (провести табулювання функції).



#### Вимоги до проекту:

- значення  $x_0$ ,  $\Delta x$  вводяться на формі (використати компоненти **TLabeledEdit**, **TLabel**, **TEdit**;
- результати табулювання функції значення функції виводяться на форму (використати **TMemo** або **TRichEdit**);
- реалізувати контроль наявності факту введення даних;
- забезпечити контролювання правильності введення числових даних для меж табулювання та кроку;
- реалізувати контроль за межами табуляції ( $a < b$ ).

#### Варіанти завдань:

1)  $y = \frac{x + \cos 2x}{x + 2}$ ,  $0,2 \leq x \leq 10$ ,  $\Delta x = 0,8$ ;

2)  $y = \frac{\operatorname{tg} 0,5x}{x^3 + 7,5}$ ,  $0,1 \leq x \leq 1,2$ ,  $\Delta x = 0,1$ ;

3)  $y = \frac{6x + 4}{\sin 3x - x}$ ,  $2,3 \leq x \leq 7,8$ ,  $\Delta x = 0,9$ ;

4)  $y = \frac{2,5x^3}{e^{2x} + 2}$ ,  $0 \leq x \leq 0,5$ ,  $\Delta x = 0,1$ ;

5)  $y = \frac{1,5x - \ln 2x}{3x + 1}$ ,  $2,5 \leq x \leq 9$ ,  $\Delta x = 0,8$ ;

6)  $y = \frac{3x + 1}{\operatorname{arctg} x}$ ,  $0,1 \leq x \leq 1,5$ ,  $\Delta x = 0,2$ ;

7)  $y = \frac{\cos^2 x}{x^2 + 1}$ ,  $3,8 \leq x \leq 7,6$ ,  $\Delta x = 0,6$ ;

8)  $y = \frac{e^{2x} - 8}{x + 3}$ ,  $-1 \leq x \leq 2,3$ ,  $\Delta x = 0,7$ ;

9)  $y = \frac{x + \cos 2x}{3x}$ ,  $2,3 \leq x \leq 5,4$ ,  $\Delta x = 0,8$ ;

$$10) \quad y = \frac{\cos^3 t^2}{1,5t + 2}, \quad 2,3 \leq t \leq 7,2, \quad \Delta t = 0,8;$$

$$11) \quad y = \frac{x^3 + 2x}{3 \cos \sqrt{x} + 1}, \quad 0 \leq x \leq 2 \quad \Delta x = 0,4;$$

$$12) \quad z = \frac{t + \sin 2t}{t^2 - 3}, \quad 2,4 \leq t \leq 6,9 \quad \Delta x = 0,4;$$

$$13) \quad y = \frac{x^3 - 2}{3 \ln x}, \quad 4,5 \leq x \leq 16,4 \quad \Delta x = 2,2;$$

$$14) \quad y = \frac{\arccos x}{2x + 1}, \quad 0,1 \leq x \leq 0,9 \quad \Delta x = 0,1;$$

$$15) \quad y = \frac{5 \operatorname{tg}(x + 7)}{(x + 3)^2}, \quad 1,2 \leq x \leq 6,3 \quad \Delta x = 0,2;$$

$$16) \quad y = \frac{5 \lg x}{x^2 - 1}, \quad 1,2 \leq x \leq 3,8 \quad \Delta x = 0,4;$$

$$17) \quad z = \frac{2 \sin^2(x + 2)}{x^2 + 1}, \quad 7,2 \leq x \leq 12 \quad \Delta x = 0,5;$$

$$18) \quad y = \frac{(3x + 2)^2}{\sin x + 3}, \quad 4,8 \leq x \leq 7,9 \quad \Delta x = 0,4;$$

$$19) \quad y = \frac{\operatorname{tg} 2t - 3t}{t + 3}, \quad 0,2 \leq t \leq 0,8 \quad \Delta t = 0,1;$$

$$20) \quad y = \frac{(x + 2)^2}{\sqrt{x^2 + 1}}, \quad 2,3 \leq x \leq 8,3 \quad \Delta x = 0,6;$$

$$21) \quad y = \frac{t - \ln 2t}{3t + 1}, \quad 2,1 \leq t \leq 8,5 \quad \Delta t = 0,7;$$

$$22) \quad y = \frac{x^2 + 2x}{\cos 5x + 2}, \quad -2 \leq x \leq 4,5 \quad \Delta x = 0,5;$$

$$23) \quad y = \frac{\ln|x + 1| + 5}{2x + 3}, \quad 0,2 \leq x \leq 0,9 \quad \Delta x = 0,15;$$

$$24) \quad y = \frac{2t + 8}{|\cos 3t| + 1}, \quad 2 \leq t \leq 6,5 \quad \Delta x = 0,8;$$

$$25) \quad y = 1/\cos^3(2x) - \sqrt{x^2 - 4}, \quad 2 \leq x \leq 7, \quad \Delta x = 0,05.$$

### Завдання 2.

В середовищі програмування **Delphi** розробити програму обчислення значення функції, починаючи із заданої точки  $x_0$ , коли заданий крок  $\Delta x$  та кількість кроків  $n$ , для відповідного аргументу функції

(провести табулювання функції). Серед значень функції знайти мінімальне, максимальне та середнє арифметичне значення.

### Вимоги до проекту:

- значення  $x_0$ ,  $\Delta x$  вводяться на формі (*TLabeledEdit*, *TEdit*),
- значення  $n$  (кількість кроків) вводяться на формі за допомогою компоненти *TSpinEdit*,
- значення точності (значення  $t$  – величина може набувати цілі значення від 0 до 6, задає кількість цифр після коми для значення функції) вводиться за допомогою поєднання компонент *TEdit* та *TUpDown*;
- значення функції виводяться на форму з використанням компоненти *TStringGrid*;
- реалізувати контроль наявності факту введення даних;
- забезпечити контролювання правильності введення числових даних для меж табулювання та кроку
- реалізувати контроль за межами табуляції ( $a < b$ );
- мінімальне, максимальне та середнє арифметичне значення виводяться на форму (*TLabeledEdit*, *TLabel*, *TEdit*)

### Варіанти завдань:

$$1. \quad y = \frac{x + \cos 2x}{x + 2}, \quad x \geq 0,6, \quad \Delta x = 1,5, \quad n = 6;$$

$$2. \quad y = \frac{\operatorname{tg} 0,5x}{x^3 + 7,5}, \quad x \geq 0,5, \quad \Delta x = 0,05, \quad n = 8;$$

$$3. \quad y = \frac{6x + 4}{\sin 3x - x}, \quad x \geq 2,8, \quad \Delta x = 0,3, \quad n = 6;$$

$$4. \quad y = \frac{2,5x^3}{e^{2x} + 2}, \quad x \geq -0,8, \quad \Delta x = 0,25, \quad n = 6;$$

$$5. \quad y = \frac{1,5x - \ln 2x}{3x + 1}, \quad x \geq 0,8, \quad \Delta x = 1,2, \quad n = 6;$$

$$6. \quad y = \frac{3x + 1}{\operatorname{arctg} x}, \quad x \geq 0,4, \quad \Delta x = 0,1, \quad n = 5;$$

$$7. \quad y = \frac{\cos^2 x}{x^2 + 1}, \quad x \geq 0,5, \quad \Delta x = 0,1, \quad n = 9;$$

$$8. \quad y = \frac{e^{2x} - 8}{x + 3}, \quad x \geq 1,5, \quad \Delta x = 0,3, \quad n = 6;$$

$$9. \quad y = \frac{x + \cos 2x}{3x}, \quad x \geq 0,8, \quad \Delta x = 0,2, \quad n = 6;$$

$$10. y = \frac{\cos^3 t^2}{1,5t + 2}, t \geq 0, \Delta t = 0,3, n = 5;$$

$$11. y = \frac{x^3 + 2x}{3 \cos \sqrt{x} + 1}, x \geq 0,3, \Delta x = 0,8, n = 7;$$

$$12. z = \frac{t + \sin 2t}{t^2 - 3}, t \geq 3,1, \Delta t = 0,8, n = 6;$$

$$13. y = \frac{x^3 - 2}{3 \ln x}, x \geq 2, \Delta x = 1,5, n = 5;$$

$$14. y = \frac{\arccos x}{2x + 1}, x \geq 0, \Delta x = 0,2, n = 4;$$

$$15. y = \frac{5 \operatorname{tg}(x + 7)}{(x + 3)^2}, x \geq 0,2, \Delta x = 0,1, n = 5;$$

$$16. y = \frac{5 \lg x}{x^2 - 1}, x \geq 1,2, \Delta x = 1,5, n = 6;$$

$$17. z = \frac{2 \sin^2(x + 2)}{x^2 + 1}, x \geq 0, \Delta x = 0,1, n = 5;$$

$$18. y = \frac{(3x + 2)^2}{\sin x + 3}, x \geq 0,2, \Delta x = 0,7, n = 6;$$

$$19. y = \frac{\operatorname{tg} 2t - 3t}{t + 3}, t \geq -0,5, \Delta t = 0,2, n = 5;$$

$$20. y = \frac{(x + 2)^2}{\sqrt{x^2 + 1}}, x \geq 6,5, \Delta x = 0,3, n = 4;$$

$$21. y = \frac{t - \ln 2t}{3t + 1}, x \geq 0,6, \Delta x = 2,5, n = 5;$$

$$22. y = \frac{x^2 + 2x}{\cos 5x + 2}, x \geq 0,6, \Delta x = 0,1, n = 5;$$

$$23. y = \frac{\ln|x + 1| + 5}{2x + 3}, x \geq 5, \Delta x = 0,4, n = 6;$$

$$24. y = \frac{2t + 8}{|\cos 3t| + 1}, x \geq 0,1, \Delta t = 0,3, n = 7;$$

$$25. y = 1/\cos^3(2x) - \sqrt{x^2 - 4}, x \geq 2, \Delta x = 0,05, n = 8.$$

### Контрольні запитання:

1. Перерахуйте типи змінних, які використані в даній роботі.
2. Назвіть стандартні арифметичні функції, які використані в проекті.

3. Оператори циклів, їх призначення.
4. Дайте означення проекту в *Delphi*.
5. **Подія** – це ... .
6. Які існують процедури обробки подій?

## Лабораторна робота №4



Національний університет

**Тема: Розробка програм з використанням одновимірних масивів.  
Використання меню при створенні проектів.**

### **Мета роботи:**

- Навчитися працювати з візуальними та невізуальними компонентами сторінки палітри сторінки *Standard: TMainMenu, TPopupMenu, TComboBox, TListBox*.
- Повторити методи роботи з компонентами сторінки *Standard: TMemo, TEdit, TButton, TLabel* та сторінки *Additional: TBitBtn, TLabeledEdit, TStringGrid*.
- Вивчення нових компонент панелі *Standard* та *Additional* .
- Вивчити методи побудови головного та контекстного меню.

### **Теоретичні відомості:**

При реалізації завдань даної лабораторної роботи пропонується використовувати компоненти, які розглянуті в попередніх лабораторних роботах: *TMemo, TButton, TLabel, TBitBtn, TLabeledEdit, TStringGrid* та нові компоненти із сторінок *Standard* та *Additional: TMainMenu, TPopupMenu, TComboBox, TListBox*. Призначення, основні властивості та методи роботи з ними наведено нижче.

#### **Компонента TMainMenu**

**Призначення:** Головне меню форми.

**Основні властивості:**

Дана компонента є невізуальною, тобто під час виконання програми компонента не відображається на формі, користувач може побачити лише меню, створене під час проектування.

Зазвичай на формі розміщується тільки одна компонента *TMainMenu* і її ім'я автоматично заноситься у властивість форми *Menu*. Але існує можливість розмістити на формі і декілька компонент *TMainMenu* із різними наборами розділів, що відповідають різним режимам роботи.

Основна властивість компоненти – *Items*. Дана властивість заповнюється за допомогою конструктора меню, який можна викликати наступним чином: подвійним натисканням на компоненті *TMainMenu*, вибором *Menu Designer* у контекстному меню при натисненні правою кнопкою миші на компоненті, натисканням кнопки із багато крапкою поряд з властивістю *Items* у вікні Інспектора Об’єктів. Вікно конструктора меню має наступний вигляд (рис.5.25):

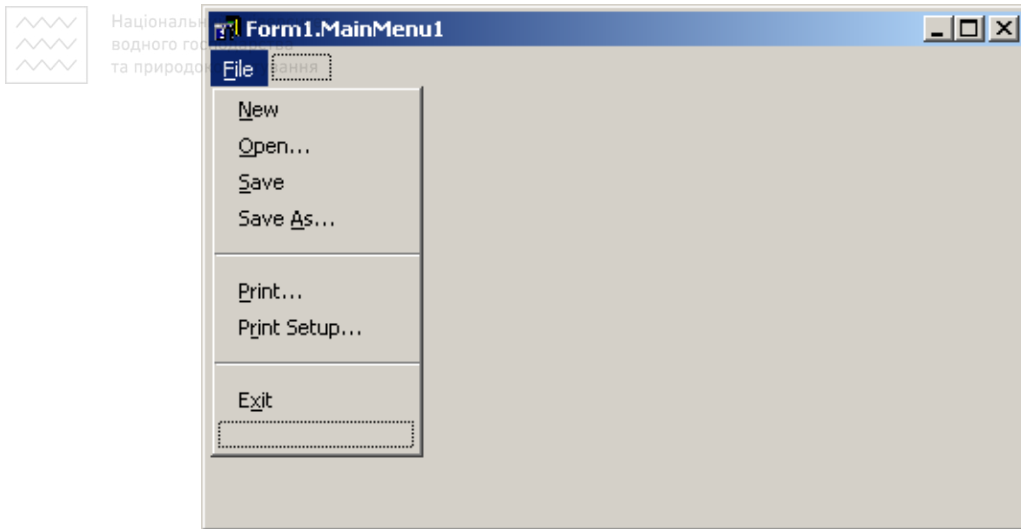


Рис. 5.25. Вікно конструктора меню

При роботі в конструкторі меню нові розділи створюються при розміщенні курсору в рамку із крапок, яка визначає місце розташування нового розділу. Розділи можна переміщувати, проводити в них редагування, вставляти нові розділи.

Інший шлях введення нового розділу – використання контекстного меню, що з’являється при натисканні правою кнопкою миші (команда *Insert* дозволяє вставляти рамку нового розділу перед виділеним розділом, команда *Delete* дозволяє видаляти виділений розділ).

В контекстному меню можна також виконати команду *Create Submenu*, що дозволяє ввести підменю у виділений розділ. Для кожного розділу в Інспекторі Об’єктів можна побачити ряд властивостей та подій даного розділу. Кожен розділ меню є об’єктом типу *TMenuItem*, що має свої властивості, методи, події.

Властивість *Caption* означає назву розділу. Заповнення даної властивості аналогічне заповненню властивості кнопок, включаючи використання символу амперсанд для позначення клавіш швидкого доступу. Якщо значення властивості *Caption* задати „—”, то замість розділу з’явиться розділювач (див. Рис.5.25 між розділами *Save As...* та *Print...*).

Властивість **Name** визначає ім'я об'єкта, що відповідає розділу меню (слід задавати змістовні імена для подальшого розуміння).

Властивість **Shortcut** визначає комбінацію клавіш, які дублюють виконання подій, що визначені розділами меню. Для визначення можливості потрібно відкрити випадаючий список властивості **Shortcut** у вікні **Інспектора Об'єктів** та вибрати потрібну комбінацію клавіш, дана комбінація з'явиться в стрічці розділу меню поряд із надписом.

Властивість **Default** визначає, чи є даний розділ розділом по замовчуванню свого підменю, тобто розділом, що виконується при подвійному натисканні користувачем на батьківському розділі.

Властивість **Break** використовується у довгих меню, щоб розбити список розділів на декілька стовпчиків (можливі значення: **mbNone** – відсутність розбиття, **mbBarBreak** – в меню вводиться новий стовпчик розділів, відділений від попереднього смугою, **mbBreak** в меню вводиться новий стовпчик розділів, відділений від попереднього пропусками).

Властивість **Checked**, що встановлена в **true**, вказує, що в розділі меню буде відображатися маркер прапорця, який показує, що даний розділ вибраний.

Якщо властивість **AutoCheck** встановлена в **true**, то при кожному виборі користувачем даного розділу маркер буде автоматично переключатися, вказуючи то на вибраний стан, то на відсутність вибору.

Властивість **RadioItem** визначає режим радіокнопки, якщо значення властивості встановлено в **true**, то даний розділ працюватиме в режимі радіокнопки разом з іншими розділами, що мають теж значення властивості **GroupIndex** (по замовчуванню значення **GroupIndex = 0**).

Для кожного розділу можуть бути встановлені під час проектування або програмно під час виконання властивість **Enabled** – доступний, **Visible** – видимий. Якщо встановити **Enabled = false**, то розділ буде відображатися сірим надписом та не буде реагувати на натискання миші, якщо задати **Visible = false**, то розділу не буде видно зовсім, а інші розділи зсунуться, зайнявши місце невидимого).

Властивість **Bitmap** дозволяє безпосередньо ввести зображення в розділ, вибравши із файлу, властивість **ImageIndex** дозволяє вказувати індекс зображення, що містяться у зовнішній компоненті **TImageList**, вказівку на яку слід вказати у властивості **TImages** компоненти **TMainMenu** (індекси починаються із 0, якщо вказати індекс -1 (значення по замовчуванню), то зображення не буде).

Основна подія розділу – **OnClick**, що виникає при натисканні користувачем на розділ або при натисканні клавіш швидкого доступу.

### Компонента TPopupMenu

Призначення: Контекстне меню.

Основні властивості:

Компонента **TPopupMenu** дозволяє створювати контекстне меню. Контекстне меню прив'язане до конкретних компонент. Вона з'являється при натисненні на ній правою кнопкою миші, коли компонента знаходиться у фокусі. Зазвичай до контекстного меню включають ті команди головного меню, які в першу чергу будуть потрібні при роботі з даною компонентою.

Віконні компоненти: панелі, вікна редагування, а також мітки та інші мають властивість **PopupMenu**, яка по замовчуванню пуста, але туди можна розмістити ім'я тієї компоненти **PopupMenu**, з якою буде пов'язана дана компонента.



Контекстне меню формується за допомогою конструктора меню, що викликається подвійним натисканням на **PopupMenu**, аналогічно як це проводилось для головного меню. Для даної компоненти існує можливість спрощеної роботи, що забезпечується командою **Select Menu**, за допомогою якої можна переходити із меню в меню, вибрати, копіювати (**Ctrl+C**) та вставляти (**Ctrl+V**) потрібний, вже створений розділ із іншого меню.

Інші властивості **PopupMenu** подібні до властивостей головного меню.

### Компонента TListBox

**Призначення:** Відображення стандартного вікна списку **Windows**, що дозволяє користувачу вибрати із нього пункти.

**Основні властивості:**

Властивість **Items** містить список всіх стрічок, який можна заповнити під час проектування в Інспекторі Об'єктів при натисканні кнопки із багатьма крапками біля даної властивості, або під час виконання додатку роботи із даною властивістю використовуючи властивості та методи класу **TStrings** – **Clear**, **Add** та інші. Під час виконання користувач може заповнити список, якщо в компоненті дозволено редагування. Існує в свою чергу можливість програмно присвоїти властивості **Items** деякого списку типу **TStrings**. Всі стрічки списку, що присвоюються, повинні мати вигляд „ім'я = значення”, тоді вміст списку буде відображатися в колонках компоненти.

В компоненті **ListBox** наявна властивість **MultiSelect**, що дозволяє користувачу вибрати декілька стрічок відразу. Якщо **MultiSelect = false** (значення по замовчуванню), то користувач може вибрати тільки один елемент списку.

Властивість **ItemIndex** дозволяє під час виконання дізнатися індекс вибраної стрічки, якщо **ItemIndex = -1**, то це означає, що ні одна стрічка не вибрана. Якщо допускається вибір багатьох стрічок (**MultiSelect = true**), то значення **ItemIndex** відповідає тому елементу списку, який знаходиться у фокусі.



При можливості вибору багатьох стрічок, перевірити, чи вибраний даний елемент, можна перевірити властивість *Selected[Index: Integer]* типу *Boolean*.

На спосіб можливості вибору багатьох стрічок при *MultiSelect = true* впливає ще властивість *ExtendedSelect*. Якщо *ExtendedSelect = true*, то користувач може виділити інтервал елементів, виділивши один із них, натиснувши клавішу *Shift* і перевівши курсор до іншого елементу. Виділити елементи, які не є сусідніми, можна, якщо під час вибору тримати натиснутою клавішу *Ctrl*. Слід зауважити, якщо *ExtendedSelect = false*, то клавіші *Shift* та *Ctrl* при виборі не працюють.

Властивість *Columns* визначає кількість стовпчиків, в яких буде відображатися список, якщо він не вміщується у вікні компоненти *ListBox*.

Властивість *Sorted* дозволяє впорядкувати список за алфавітом (при *Sorted = true* нові стрічки в список додаються не в кінець, а за алфавітом).

Якщо властивість *AutoComplete* приймає значення *false*, то користувачу надається можливість швидко знаходити стрічку в списку, натискаючи клавішу, що відповідає її першому символу.

Властивість *Style* при значенні *IbStandard* (значення по замовчуванню) відповідає списку стрічок. Інші значення *Style* дозволяють відображати у списку не тільки текст, але і зображення.

### Компонент *TComboBox*

*Призначення:* Випадаючий список із можливістю редагування. Об'єднує функції *TListBox* та *TEdit*. Користувач може ввести текст або вибрати його зі списку.

*Основні властивості:*

Стиль компоненти *TComboBox* визначається властивістю *Style*, яка може приймати наступні значення:

- *csDropDown* – список зі стрічками однакової висоти та з вікном редагування, що дозволяє користувачу вводити або редагувати текст;
- *csSimple* – розгорнутий список зі стрічками однакової висоти та вікном редагування, що дозволяє користувачу вводити або редагувати текст;
- *csDropDownList* – випадаючий список зі стрічками однакової висоти, що не містить вікна редагування;
- *csOwnerDrawFixed* – випадаючий список зі стрічками однакової висоти, в яких можуть відображатися зображення та текст;
- *csOwnerDrawVariable* – випадаючий список зі стрічками різної висоти, в яких можуть відображатися зображення та текст.

Властивість *Text* дозволяє визначити вибір користувача або введений ним текст.

Властивість **ItemIndex** використовується, якщо потрібно визначити індекс вибраного користувачем елемента списку. Значення даної властивості визначаються як і у властивості **ListBox**. Слід зауважити, якщо початкове значення **ItemIndex** не задане, то в момент запуску додатку користувач не побачить у вікні компоненти ні одного із можливих значень списку.

Властивість **MaxLength** визначає максимальне число символів, які користувач може ввести у вікно редагування, якщо **MaxLength = 0**, то число символів, які можна ввести, є необмеженим.

Властивість **DropDownCount** вказує число стрічок, що з'являються у випадаючому списку без наявності смуги прокрутки.

Властивість **Sorted** дозволяє впорядкувати список за алфавітом.

## Приклади розробки Delphi-додатків:

### Приклад 5.7

В середовищі програмування **Delphi** розробити віконний додаток для знаходження максимального елемента заданого масиву із десяти цілих чисел.

**Етап проектування** інтерфейсу проекту:

Під час проектування використаємо компоненти класу **TEdit**, **TLabel**, **TButton**, **TStringGrid**.

Основні властивості використаних компонент (рис.5.26):

**Label1.Caption:= 'Масив ';**

**Label2.Caption:= 'Максимальний елемент масиву ';**

**Button1.Caption:= 'Дані';**

**Button2.Caption:= 'Макс.ел.';**

**Button3.Caption:= 'Вийти';**

**Button4.Caption:= 'Очистити';**

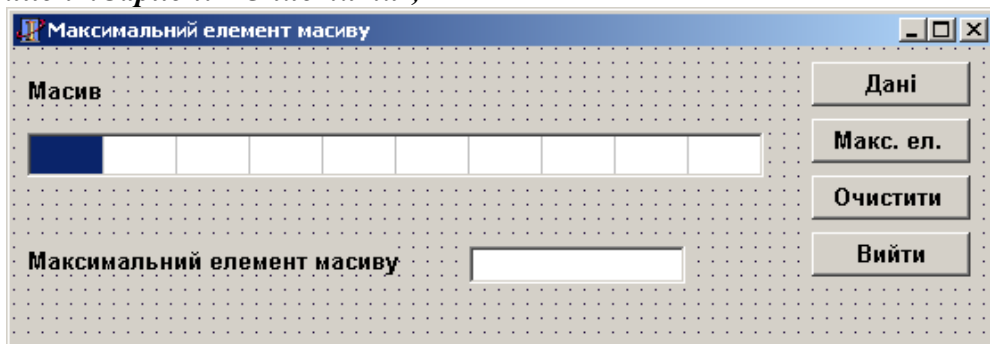


Рис. 5.26. Вікно форми проекту прикладу 5.7

Для відображення масиву використаємо компоненту *TStringGrid*, для якої під час проектування визначимо необхідну кількість комірок (властивість *ColCount*), заберемо фіксовані комірки, залишимо тільки один рядок (*RowCount:=1*).

### **Етап програмування проекту:**

При натисканні на кнопку **'Дані'**, випадковим чином генерується масив цілих чисел від -100 до 100. Процедура обробки даної події має вигляд:



```
procedure TForm1.Button1Click(Sender: TObject);  
var i:integer;  
begin  
Randomize;  
for i:=0 to 9 do  
StringGrid1.Cells[i,0]:=IntToStr( RandomRange(-100,100));  
end;
```

Максимальний елемент обчислюється при натисканні на кнопку **'Макс. ел.'**. Обробка події реалізується наступною функцією:

```
procedure TForm1.Button2Click(Sender: TObject);  
var a: array [0..9] of integer;  
    i:integer;  
begin  
for i:=0 to 9 do  
    a[i]:= StrToInt(StringGrid1.cells[i,0]);  
    Edit1.Text:=IntToStr(MaxIntValue(a));  
end;
```

Також на формі наявна кнопка очищення даних **'Очистити'** та виходу з програми **'Вийти'**.

### **Результат виконання проекту:**

Для заданого масиву, що генерується, випадковим чином визначено максимальний елемент 83 (рис.5.27).

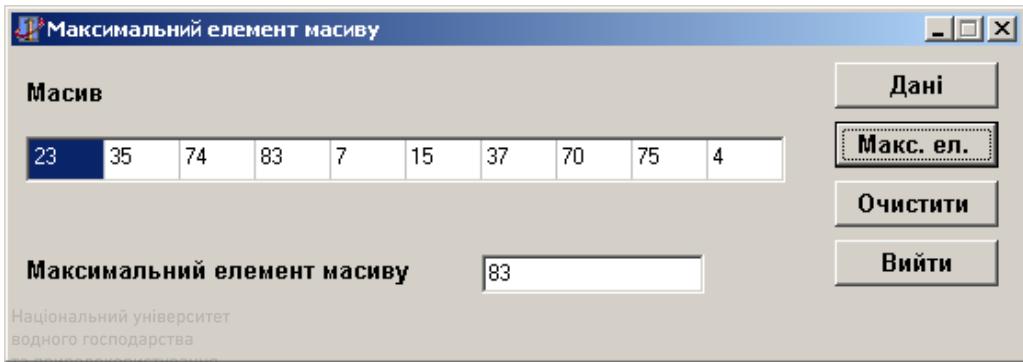


Рис.5.27. Результат виконання проекту прикладу 5.7

### Приклад 5.8

В середовищі програмування *Delphi* розробити програму, яка б для заданих масивів цілих чисел *A* та *B* розмірності *n* визначала:

- 1) кількість відповідних елементів масиву *A* та *B*, які рівні між собою;
- 2) кількість відповідних елементів масиву *A*, які є квадратами відповідних елементів масиву *B*;
- 3) кількість відповідних елементів масиву *A* та *B*, які нерівні між собою.

**Етап проектування** інтерфейсу проекту:

Під час проектування використаємо компоненти класу *TEdit*, *TLabel*, *TButton*, *TStringGrid*, *TSpinEdit*, *TMemo*.

Основні властивості використаних компонент (рис.5.28):

*Label1.Caption:= 'Розмірність масивів ';*

*Label2.Caption:= 'Завдання';*

*Label3.Caption:= 'A';*

*Label4.Caption:= 'B';*

*Button1.Caption:= 'Визначити';*

*SpinEdit1.MaxValue:=20;*

*SpinEdit1.MinValue:=1;*

При формуванні проекту використані три компоненти *CheckBox*, які визначають потрібне завдання:

*CheckBox1.Caption:= 'Кількість відповідних елементів масивів в A і B, рівних між собою';*

*CheckBox2.Caption:= 'Кількість відповідних елементів масивів в A, що є квадратами відповідних елементів у B';*

*CheckBox3.Caption:= 'Кількість відповідних елементів масивів в A і B, нерівних між собою'.*

Тобто за бажанням можна вибрати кожне із завдань окремо, або всі разом.

Для відображення масиву використаємо компоненту *TStringGrid*, для якої під час проектування заберемо фіксовані комірки, залишимо лише

один рядок, а під час виконання формування необхідної кількості комірок забезпечуватиметься за допомогою компоненти *SpinEdit1*.

На етапі проектування в проєкті сформовано наступне меню:

*Дані*

*Генеруються випадковим чином*

*Задаються користувачем*

*Редагування*

*Очистити*

*Вихід*



Національний університет  
водного господарства  
та природокористування

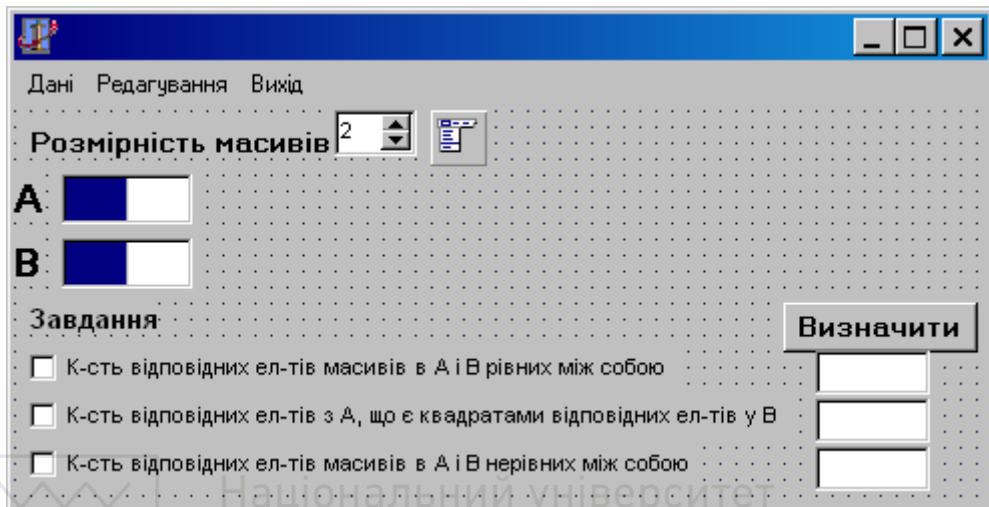


Рис.5.28. Вікно форми проєкту прикладу 5.8

**Етап програмування проєкту:**

Як глобальні описані наступні змінні:

```
var  
    size:integer;  
    a:array[1..20] of integer;  
    b:array[1..20] of integer;
```

Відповідно до зміни значення наступна процедура змінює кількість комірок для введення значень масиву:

```
procedure TForm1.SpinEdit1Change(Sender: TObject);  
begin  
    StringGrid1.ColCount:=SpinEdit1.Value;  
    StringGrid1.Width:=SpinEdit1.Value*31+10;  
    StringGrid2.ColCount:=SpinEdit1.Value;
```

```
StringGrid2.Width:=SpinEdit1.Value*31+10;  
end;
```

Процедура, яка дозволяє закривати додаток:

```
procedure TForm1.N4Click(Sender: TObject);  
begin  
Close;  
end;
```



Національний університет  
водного господарства  
та природокористування

Процедура, яка визначає можливість вибору першого завдання:

```
procedure TForm1.CheckBox1Click(Sender: TObject);  
begin  
if CheckBox1.Checked then Edit1.Visible:=true  
else Edit1.Visible:=false;  
end;
```

Процедура, яка визначає можливість вибору другого завдання:

```
procedure TForm1.CheckBox2Click(Sender: TObject);  
begin  
if CheckBox2.Checked then Edit2.Visible:=true;  
else Edit2.Visible:=false;  
end;
```



Процедура, яка визначає можливість вибору третього завдання:

```
procedure TForm1.CheckBox3Click(Sender: TObject);  
begin  
if CheckBox3.Checked then Edit3.Visible:=true  
else Edit3.Visible:=false;  
end;
```

Обробка події **OnClick** для кнопки **'Визначити'** забезпечується наступною процедурою:

```
procedure TForm1.Button1Click(Sender: TObject);  
var  
i,equal, nequal, square:integer;  
begin  
for i:=1 to SpinEdit1.Value do begin  
a[i]:=StrToInt(StringGrid1.Cells[i-1,0]);  
b[i]:=StrToInt(StringGrid2.Cells[i-1,0]); end;  
equal:=0;
```

```

nequal:=0;
square:=0;
for i:=1 to SpinEdit1.Value do
  begin
    if a[i]= b[i] then equal:=equal+1;
    if a[i]=b[i]*b[i] then square:=square+1;
    if a[i]<>b[i] then nequal:=nequal+1;
    end;
if CheckBox1.Checked=true then
  Edit1.Text:=IntToStr(equal);
if Checkbox2.Checked=true then
  Edit2.Text:=IntToStr(square);
if CheckBox3.Checked=true then
  Edit3.Text:=IntToStr(nequal);
end;

```



Національний університет  
водного господарства  
та природокористування

Процедура, що забезпечує роботу підрозділу меню ‘Геруються випадковим чином’, для заповнення масиву випадковими значеннями із діапазону від 0 до 100.

```

procedure TForm1.N2Click(Sender: TObject);
var i:integer;
begin
  Randomize;
for i:=1 to SpinEdit1.Value do
  begin
    StringGrid1.Cells[i-1,0]:=IntToStr(Random(100));
    StringGrid2.Cells[i-1,0]:=IntToStr(Random(100));
  end;
end;

```



Процедура, що забезпечує роботу підрозділу меню ‘Геруються випадковим чином’, для заповнення масиву випадковими значеннями із діапазону від 0 до 100.

```

procedure TForm1.N3Click(Sender: TObject);
var i:integer;
begin
  StringGrid1.Options:=[goFixedVertLine,goFixedHorzLine,
    goVertLine, goHorzLine, goRangeSelect, goEditing];
  StringGrid2.Options:=[goFixedVertLine, goFixedHorzLine,
    goVertLine, goHorzLine, goRangeSelect, goEditing];
for i:=1 to SpinEdit1.Value do
  begin

```

```

StringGrid1.Cells[i-1,0]:="";
StringGrid2.Cells[i-1,0]:="";
end;
end;

```

Процедура, що забезпечує роботу підрозділу меню 'Очистити', для очищення масиву від даних:

```

procedure TForm1.N5Click(Sender: TObject);
var i:integer;
begin
for i:=1 to SpinEdit1.Value do
begin
StringGrid1.Cells[i-1,0]:="";
StringGrid2.Cells[i-1,0]:="";
end;
CheckBox1.Checked:=false;
CheckBox2.Checked:=false;
CheckBox3.Checked:=false;
end;

```

**Результат виконання** проекту:

Приклад виконання програми подано на рис.5.29.

Дані Редагування Вихід

Розмірність масивів 15

<b>A</b>	43	92	44	13	64	82	9	3	73	20	53	23	82	8	75
<b>B</b>	91	86	46	13	8	31	12	28	80	20	60	53	49	32	90

**Завдання**

К-сть відповідних ел-тів масивів в А і В рівних між собою 2

К-сть відповідних ел-тів з А, що є квадратами відповідних ел-тів у В 1

К-сть відповідних ел-тів масивів в А і В нерівних між собою 13

**Визначити**

Рис.5.29. Результат виконання проекту прикладу 5.8

**Завдання для виконання лабораторної роботи:**

**Завдання 1.**



В середовищі програмування **Delphi** розробити програму, яка б для заданого масиву  $D(n)$  із  $n$  елементів виконувала наведені нижче варіанти завдань:

### Вимоги до проекту:

- Розробити головне меню (використати компоненту **TMainMenu**), яке б мало, наприклад, наступні розділи:



Національний університет  
водного господарства  
та природокористування

#### **Дані**

*Розмірність масиву*

*Генеруються випадковим чином*

*Задаються користувачем*

#### **Редагування**

*Очистити дані*

#### **Допомога**

*Інформація про користування програмою*

*Інформація про проект*

#### **Вихід**

- Розробити контекстні меню для кожної компоненти, що знаходиться на формі, та для форми (використати компоненту **TPopupMenu**); в контекстному меню мають бути команди, які стосуються окремої компоненти, або додатку в цілому (якщо меню для форми).
- Для введення масивів використати компонент **TStringGrid** або **TMemo**;
- Забезпечити правильність введення даних.

### Варіанти завдань:

1. Знайти суму від'ємних елементів, які стоять на парних місцях в масиві.
2. Знайти середнє арифметичне невід'ємних елементів масиву, які стоять на непарних місцях.
3. Обчислити добуток елементів масиву.
4. Знайти максимальний по модулю елемент масиву.
5. Визначити порядкові номери невід'ємних елементів масиву.
6. У масиві  $D(n)$  із  $n$  елементів є хоча б один нуль. Знайти мінімальний по модулю елемент у частині масиву після останнього нуля.
7. Знайти мінімальний серед додатних елементів масиву.
8. Знайти мінімальний серед від'ємних елементів та вказати його номер.

9. Знайти максимальний серед від'ємних елементів та вказати його номер.
10. Знайти різницю між максимальним та мінімальним серед від'ємних елементів.
11. Сформувати новий масив, замінивши елементи, які стоять на парних місцях на елементи, які стоять на непарних місцях та навпаки
12. У масиві  $D(n)$  із  $n$  елементів є хоча б один нуль. Визначити кількість елементів після першого нуля.
13. Визначити середнє арифметичне додатних елементів та суму від'ємних.
14. Знайти і надрукувати суму додатних елементів масиву.
15. Обчислити добуток додатних елементів масиву.
16. Визначити кількість від'ємних елементів масиву.
17. Визначити порядкові номери від'ємних елементів масиву.
18. Підрахувати кількість елементів масиву, значення яких більше 2,3.
19. Визначити номери додатних елементів масиву.
20. Знайти кількість додатних елементів масиву.
21. Визначити середнє арифметичне додатних елементів масиву.
22. Знайти середнє арифметичне елементів масиву і на місці від'ємних елементів поставити це значення.
23. Забезпечити заміну елементів, значення яких менші за 1 на середнє арифметичне. Підрахувати кількість проведених замін.
24. Кожен елемент дорівнює 1, 2 або 3. Впорядкувати масив таким чином, щоб всі одинички йшли першими, а трійки останніми.
25. Написати програму, яка для заданої послідовності  $B$ , розмірності  $n$ , де  $n > 0$ , обчислює довжину її максимальної зростаючої послідовності.

## Завдання 2.

В середовищі програмування **Delphi** розробити проект для реалізації завдань, які наведені нижче

### Вимоги до проекту:

- забезпечити створення головного меню;
- розмірність масивів має вводитись на окремій формі (використати компоненту **TSpinEdit**);
- для вибору пункту, за яким буде здійснюватись пошук, скористатись компонентою **TListBox** або **TComboBox**;

- для введення масивів використати компонент *TStringGrid* або *TMemo*;
- вивід результатів для кожного пунктів пошуку має здійснюватись на різних формах.

### **Варіанти завдань:**

1. В середовищі програмування *Delphi* розробити програму, яка б для заданих масивів цілих чисел *A* та *B* розмірності *n* визначала:

- 1) номери відповідних елементів масивів *A* та *B*, які рівні між собою;
- 2) номери відповідних елементів масиву *A*, які є кубами відповідних елементів масиву *B*;
- 3) номери відповідних елементів масиву *A*, які є коренями квадратними відповідних елементів масиву *B*.

2. В середовищі програмування *Delphi* розробити програму, яка б для заданих масивів цілих чисел *A* та *B* розмірності *n* визначала:

- 1) номери відповідних елементів масиву *A*, які є квадратами відповідних елементів масиву *B*;
- 2) номери відповідних елементів масиву *A*, які є більші відповідних елементів масиву *B*;
- 3) номери *i* такі, що  $A[i] = \sin(B[i])$ .

3. В середовищі програмування *Delphi* розробити програму, яка б для заданих масивів цілих чисел *A* та *B* розмірності *n* виконувала:

- 1) злиття двох масивів в один масив;
- 2) сортування новоствореного масиву у порядку зростання його елементів;
- 3) підрахунок суми елементів масиву.

4. В середовищі програмування *Delphi* розробити програму, яка б для заданих масивів цілих чисел *A* та *B* розмірності *n* виконувала:

- 1) злиття двох масивів в один масив, відсортований у порядку зростання;
- 2) підрахунок суми мінімального елемента масиву *A* та максимального елемента масиву *B*;
- 3) підрахунок суми елементів двох масивів.

5. В середовищі програмування *Delphi* розробити програму, яка б для двох фіксованих, впорядкованих по зростанню масиви *A* розмірності *n* та *B* розмірності *m*, де  $n > 0$ ,  $m > 0$  визначала

- 1) кількості значень, які зустрічаються як масиві *A*, так і в *B*;
- 2) кількість значень, які зустрічаються тільки в масиві *A*;

3) кількість значень, які зустрічаються тільки в масиві В.  
Відомо, що в межах масивів А та В немає однакових елементів.

6. В середовищі програмування *Delphi* розробити програму, яка б для масиву дійсних чисел розмірності  $n$  визначала:

- 1) максимальний елемент;
- 2) мінімальний елемент;
- 3) кількість перестановок елементів в масиві, якщо впорядкувати його за зростанням.



7. В середовищі програмування *Delphi* розробити програму, яка б для заданих масивів цілих чисел А та В розмірності  $n$  визначала:

- 1) кількість відповідних елементів масиву А, які більші відповідних елементів масиву В;
- 2) кількість відповідних елементів масиву А, які менші відповідних елементів масиву В;
- 3) кількість відповідних елементів масиву А, які кратні відповідним елементам масиву В.

8. В середовищі програмування *Delphi* розробити програму, яка б для заданих масивів цілих чисел А та В розмірності  $n$  визначала:

- 1) номери відповідних елементів масивів А та В, які не рівні між собою;
- 2) номери відповідних елементів масиву А, які є сумою цифр елементів масиву В;
- 3) номери  $i$  такі, що  $A[i] = \cos(B[i])$ .

9. В середовищі програмування *Delphi* розробити програму, яка б для заданих масивів цілих чисел А та В розмірності  $n$  визначала:

- 1) кількість відповідних елементів масиву А та В, сума цифр яких більша 5;
- 2) кількість відповідних елементів масиву А, які є менші відповідних елементів масиву В;
- 3) кількість відповідних елементів масиву А та В, які рівні між собою;

10. В середовищі програмування *Delphi* розробити програму, яка б для заданої таблиці розмірності  $(n \times m)$ , елементи якої можуть бути набори довільних символів, визначала:

- 1) кількість комірок  $a_{ij}$ , в яких довжина відповідного слова більша за  $i + j$ ;
- 2) кількість комірок  $a_{ij}$ , в яких довжина відповідного слова менша за  $i + j$ ;

- 3) кількість комірок  $a_{ij}$ , в яких довжина відповідного слова дорівнює сумі цифр, що зустрічаються в даному слові.

11. В середовищі програмування **Delphi** розробити програму, яка б для заданих масивів А та В, елементами яких є стрічки (розмірності  $n$ ), визначала:

- 1) кількість взаємних пар  $A[i]$  та  $B[i]$ , де довжина стрічки  $A[i]$  дорівнює сумі цифр, що зустрічаються в елементі  $B[i]$ ;
- 2) кількість взаємних пар  $A[i]$  та  $B[i]$ , де символи, що стоять на  $i$ -му місці в  $A[i]$  та  $B[i]$ , рівні між собою.
- 3) кількість взаємних пар  $A[i]$  та  $B[i]$ , де кількість символів  $A[i]$  дорівнює кількості символів в  $B[i]$ .

12. В середовищі програмування **Delphi** розробити програму, яка б для заданих масивів цілих чисел А та В розмірності  $n$  визначала:

- 4) номери відповідних елементів масивів А та В, які рівні між собою;
- 5) номери відповідних елементів масиву А, які є кубами відповідних елементів масиву В;
- 6) номери відповідних елементів масиву А, які є коренями квадратними відповідних елементів масиву В.

13. В середовищі програмування **Delphi** розробити програму, яка б для заданої таблиці розмірності  $(n \times m)$ , елементами якої можуть бути набори довільних символів, визначала:

- 1) кількість комірок  $a_{ij}$ , в яких довжина відповідного слова більша за  $i * j$ ;
- 2) кількість комірок  $a_{ij}$ , в яких довжина відповідного слова менша за  $i + j$ ;
- 3) кількість комірок  $a_{ij}$ , в яких довжина відповідного слова більша суми цифр, що зустрічаються в даному слові.

14. В середовищі програмування **Delphi** розробити програму, яка б для заданих масивів цілих чисел А та В розмірності  $n$  визначала:

- 1) номери відповідних елементів масиву А, які є добутками цифр відповідного значення масиву В;
- 2) кількість елементів масиву А, які є більші відповідних елементів масиву В;
- 3) номери  $i$  такі, що  $A[i] = \sin(2 * B[i])$ .

15. Середовищі програмування **Delphi** розробити програму, яка б для заданих масивів цілих чисел  $A$  та  $B$  розмірності  $n$  виконувала:

- 1) злиття двох масивів в один масив із видаленням елементів, які менші 10;
- 2) сортування новоствореного масиву у порядку спадання його елементів;
- 3) підрахунок суми елементів масиву із парними номерами.



16. В середовищі програмування **Delphi** розробити програму, яка б для заданих масивів цілих чисел  $A$  та  $B$  розмірності  $n$  виконувала:

- 1) злиття двох масивів в один масив, відсортований у порядку зростання ;
- 2) підрахунок середньо арифметичного значення елементів масивів  $A$  та  $B$ ;
- 3) підрахунок суми елементів двох масивів, які кратні 2.

17. В середовищі програмування **Delphi** розробити програму, яка б для двох фіксованих, впорядкованих по зростанню масиви  $A$  розмірності  $n$  та  $B$  розмірності  $m$ , де  $n > 0$ ,  $m > 0$  визначала

- 1) суму значень, які зустрічаються як масиві  $A$ , так і в  $B$ ;
- 2) кількість значень, які зустрічаються тільки в масиві  $A$  та стоять на парних місцях;
- 3) кількість значень, які зустрічаються тільки в масиві  $B$  та стоять на непарних місцях.

Відомо, що в межах масивів  $A$  та  $B$  немає однакових елементів.

18. В середовищі програмування **Delphi** розробити програму, яка б для масиву дійсних чисел розмірності  $n$  визначала:

- 1) максимальний елемент;
- 2) мінімальний елемент, серед елементів які мають парні номери;
- 3) кількість перестановок елементів в масиві, якщо впорядкувати його за зростанням.

19. В середовищі програмування **Delphi** розробити програму, яка б для заданих масивів цілих чисел  $A$  та  $B$  розмірності  $n$  визначала:

- 1) кількість відповідних елементів масиву  $A$ , які більші відповідних парних елементів масиву  $B$ ;
- 2) кількість відповідних елементів масиву  $A$ , які менші відповідних непарних елементів масиву  $B$ ;
- 3) кількість відповідних елементів масиву  $A$ , які кратні відповідним елементам масиву  $B$ .

20. В середовищі програмування **Delphi** розробити програму, яка б для масиву цілих чисел розмірності  $n$  визначала:

- 1) максимальний елемент, серед парних елементів масиву із непарними номерами;
- 2) мінімальний елемент, серед елементів які мають непарні номери;
- 3) кількість елементів в масиві, які кратні 3.

21. В середовищі програмування **Delphi** розробити програму, яка б для заданих масивів цілих чисел  $A$  та  $B$  розмірності  $n$  визначала:

- 1) кількість відповідних елементів масиву  $A$ , які діляться на відповідні елементи масиву  $B$ ;
- 2) номери відповідних елементів масиву  $A$ , в яких добуток цифр у числі співпадає із кубами відповідних елементів масиву  $B$ ;
- 3) номери відповідних елементів масиву  $A$ , які є коренями квадратними відповідних елементів масиву  $B$ .

22. В середовищі програмування **Delphi** розробити програму, яка б для заданих масивів цілих чисел  $A$  та  $B$  розмірності  $n$  визначала:

- 1) кількість відповідних елементів масиву  $A$  та  $B$ , які кратні 2;
- 2) номери відповідних елементів масиву  $A$ , які більші відповідних елементів масиву  $B$ ;
- 3) кількість відповідних елементів масиву  $A$  та  $B$ , які не рівні між собою;

23. В середовищі програмування **Delphi** розробити програму, яка б для заданої таблиці розмірності  $(n \times m)$ , елементи якої можуть бути набори довільних символів, визначала:

- 1) кількість комірок  $a_{ij}$ , в яких довжина відповідного слова більша за суму цифр, які зустрічаються у слові;
- 2) кількість комірок  $a_{ij}$ , в яких кількість символів переважає над кількістю цифр;
- 3) кількість комірок  $a_{ij}$ , в яких довжина кратна 3.

24. В середовищі програмування **Delphi** розробити програму, яка б для заданих масивів  $A$  та  $B$ , елементами яких є стрічки (розмірності  $n$ ), визначала:

- 1) кількість взаємних пар  $A[i]$  та  $B[i]$ , де довжина стрічки  $A[i]$  дорівнює сумі цифр, що зустрічаються в елементі  $B[i]$ ;
- 2) кількість взаємних пар  $A[i]$  та  $B[i]$ , де довжини стрічок рівні;

3) кількість взаємних пар  $A[i]$  та  $B[i]$ , де кількість символів  $A[i]$  дорівнює кількості цифр в  $B[i]$ .

25. В середовищі програмування **Delphi** розробити програму, яка б для двох фіксованих, впорядкованих по зростанню масиви А розмірності n та В розмірності m, де  $n > 0$ ,  $m > 0$  визначала

1) добуток елементів, які зустрічаються тільки в масиві А;

2) номери елементів масиву А, які менші відповідних елементів масиву В;

3) кількість значень, які зустрічаються тільки в масиві В та стоять на непарних місцях.

Відомо, що в межах масивів А та В немає однакових елементів.

### **Контрольні запитання:**

1. Як в загальному описуються масиви?
2. Який масив називають одновимірним?
3. Що таке динамічний масив?
4. Як описується процедура введення та виведення масивів?
5. Назвіть властивості компоненти **StringGrid**.
6. Методи роботи із **StringGrid**.

### **Лабораторна робота №5**

**Тема: Розробка програм з обробки багатовимірних масивів.  
Створення багатовіконних додатків.**

#### **Мета роботи:**

- Навчитись працювати з візуальними компонентами сторінки **Standard: TPanel**, сторінки **Win32: TTabControl** та **TStatusBar**.
- Закріпити навички роботи з компонентами сторінки **Samples: TSpinEdit**, сторінки **Standard: TMemo, TEdit, TButton, TLabel** та сторінки **Additional: TBitBtn, TLabelEdit, TStringGrid**.
- Вивчити призначення, основні властивості та методи роботи з компонентами сторінки **Standard: TPanel**, сторінки **Win32: TTabControl** та **TStatusBar**.
- Вивчити елементи створення багатовіконного **Windows**-додатку.

#### **Теоретичні відомості:**



В процесі виконання завдань даної лабораторної роботи пропонується використати компоненти *TSpinEdit*, *TMemo*, *TEdit*, *TButton*, *TLabel*, *TBitBtn*, *TLabelledEdit*, *TStringGrid*, призначення, основні властивості та методи роботи, що розглянуті в попередніх лабораторних роботах, та компоненти *TPanel*, *TTabControl* та *TStatusBar*, основна інформація про які буде викладена нижче.

### Компонента TPanel

**Призначення:**

Компонента *TPanel* використовується як контейнер для групування різних елементів інтерфейсу, які функціонально пов'язані один з одним, для організації інструментальних панелей, ліній стану, для відображення тексту з можливостями об'ємного оформлення.

**Основні властивості:**

До основних властивостей, які стосуються зовнішнього вигляду панелі, можна віднести:

- **BevelInner** – стиль внутрішньої частини панелі,
- **BevelOuter** – стиль зовнішньої частини панелі,
- **BevelWidth** – ширина зовнішньої частини панелі,
- **BorderStyle** – стиль рамки,
- **BorderWidth** – ширина рамки.

Для відображення тексту використовується властивість **Caption**. Значення властивості **Caption** можна задавати під час проектування або програмним чином:

**Panel1.Caption:= 'Текст, який буде відображатися на панелі';**

Колір фону панелі визначається властивістю **Color**, а колір тексту – підвластивістю **Color** властивості **Font**. Розміщення компоненти на формі визначається властивостями **Top**, **Left**, **Height**, **Width**, **Align**, **Anchors**, **Constraints**.

Властивість **Anchors** визначає прив'язку компоненти *TPanel* до форми при зміні її розмірів. Дана властивість є множиною та може містити наступні елементи, які визначають прив'язаність до певного краю форми (**akTop** – до верхнього краю, **akLeft** – до лівого краю, **akRight** – до правого краю, **akBottom** – до нижнього краю).

Властивість **Align** компоненти *TPanel* визначає, чи будуть проводитись зміни щодо розташування компоненти на формі. Властивість **Align** може приймати наступні значення:

- **alNone** – компонента залишається там, де була розташована під час проектування, і розміри компоненти не змінюються;
- **alTop** – компонента займає всю верхню частину форми, при цьому під час виконання ширина компоненти змінюється при зміні ширини форми, висота залишається незмінною;



- ***alButtom*** – компонента займає всю нижню частину форми, при цьому під час виконання ширина компоненти змінюється при зміні ширини форми, висота залишається незмінною;
- ***alLeft*** – компонента займає всю ліву частину форми, при цьому під час виконання висота компоненти змінюється при зміні висоти форми, ширина залишається незмінною;
- ***alRight*** – компонента займає всю праву частину форми, при цьому під час виконання висота компоненти змінюється при зміні висоти форми, ширина залишається незмінною;
- ***alClient*** – компонента займає всю клієнтську частину форми, при цьому під час виконання розміри компоненти змінюється при зміні розмірів форми, висота залишається незмінною;

Властивість ***Top*** визначає координату верхнього краю компоненти в пікселях. За початок відліку береться верхня границя клієнтської частини форми.

Властивість ***Left*** визначає координату лівого краю компоненти в пікселях. За початок відліку береться ліва границя клієнтської частини форми.

Властивість ***Height*** визначає висоту компоненти в пікселях.

Властивість ***Width*** визначає ширину компоненти в пікселях.

Властивість ***Constraints*** дозволяє задавати обмеження на допустиму зміну розмірів компоненти при зміні розмірів вікна додатку.

Серед методів компоненти ***TPanel*** можна зазначити метод ***Repaint***, який дозволяє перемальовувати компоненту. При зміні розмірів вікна додатку потрібно перемальовувати компоненту ***TPanel*** у випадку, коли вона відображає текст, оскільки вирівнювання тексту може порушитись. Таким чином, потрібно забезпечити створення процедури обробки події ***OnResize***:

```
procedure TForm1.FormResize(Sender: TObject);  
begin  
Panel1.Repaint;  
end;
```

Слід зазначити, що на панелі можна розмістити текст тільки в один рядок.

### **Компонента TTabControl**

**Призначення:** Дозволяє створювати панель із закладками, які може вибирати користувач.

**Основні властивості:**

Дана компонента має єдину панель для всіх закладок і всі дії стосовно зміни інформації на даній панелі залежно від вибраної користувачем закладки повинні здійснюватися програмним чином.

Стиль відображення компоненти **TTabControl** визначає властивість **Style**, яка може приймати наступні значення: **tsTabs** – стандартні закладки об'ємного виду; **tsButtons** – закладки у вигляді кнопок; **tsFlatButtons** – закладки у вигляді плоских кнопок.

Властивість **Tabs** має тип **String** та містить список всіх закладок.

Наприклад, для створення програми, яка б дозволяла переглядати інформацію про успішність студентів різних груп (рис.5.30):

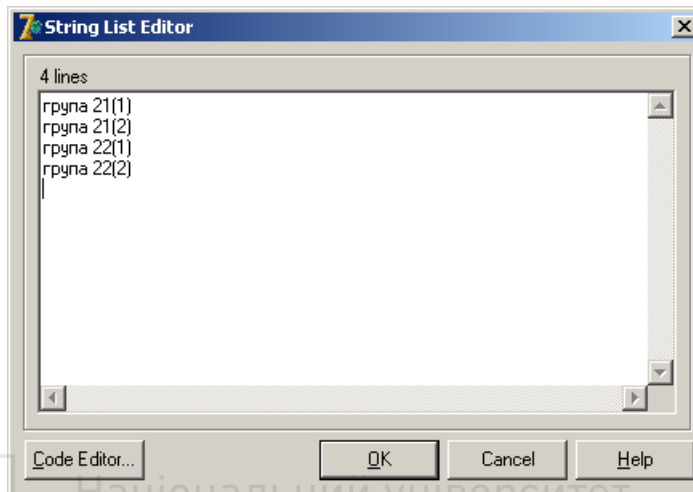


Рис.5.30. Редактор формування закладок

Залежно від вибраної закладки, користувачу буде подана відповідна інформація (рис.5.31).

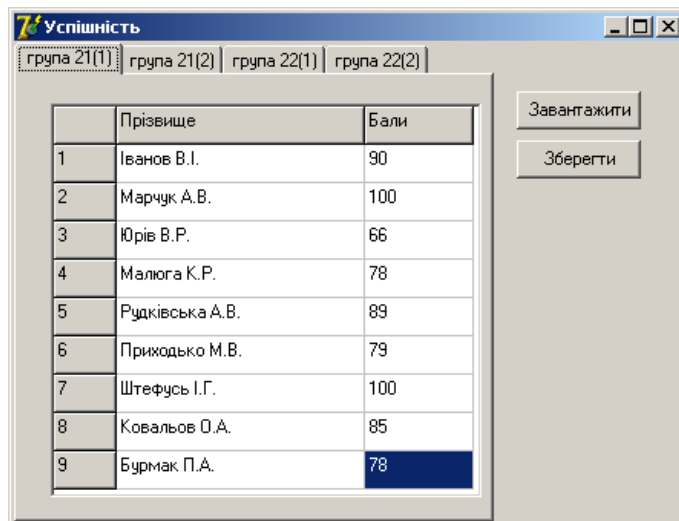


Рис.5.31. Приклад перегляду інформації відповідно до вибраної закладки.

Приклади розробки Delphi-додатків.

### Приклад 5.9

В середовищі програмування **Delphi** реалізувати проект для знаходження  $C = A + B$  та  $D = A - B$ , де  $A, B$  – матриці розмірності  $n \times n$ .

Державний університет  
водного господарства  
та природокористування

**Етап проектування** інтерфейсу проекту:

Під час проектування інтерфейсу проекту використаємо компоненти класу **TMemo, TEdit, TLabel, TButton**.

Для задання розмірності матриць сформуємо першу форму (рис.5.32):

**Form2.Caption:= 'Розмірність матриць';**

**Label1.Caption:= 'Кількість рядків';**

**Label2.Caption:= 'Кількість стовпчиків';**

**Button1.Caption:= 'Задати';**

**SpinEdit1.MaxValue:=10;**

**SpinEdit1.MinValue:=1;**

**SpinEdit2.MaxValue:=10;**

**SpinEdit2.MinValue:=1;**

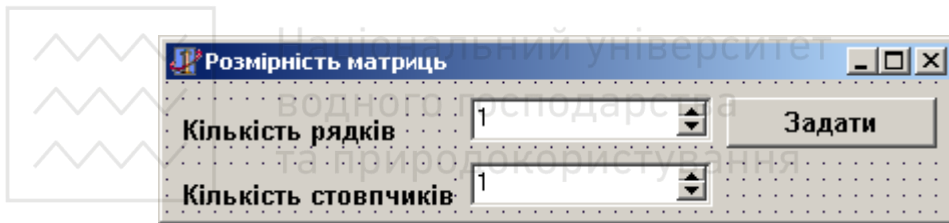


Рис.5.32. Вікно форми для задання розмірності

На другій формі (рис.5.33) здійснюється введення матриць:

**Label1.Caption = 'введіть матрицю A'**

**Label2.Caption = 'Введіть матрицю B'**

**Label4.Caption = 'Знайти'**

**Button1.Caption = 'Ввести матрицю A'**

**Button2.Caption = 'Ввести матрицю B'**

**Button5.Caption = 'A+B'**

**Button5.Caption = 'A-B'**

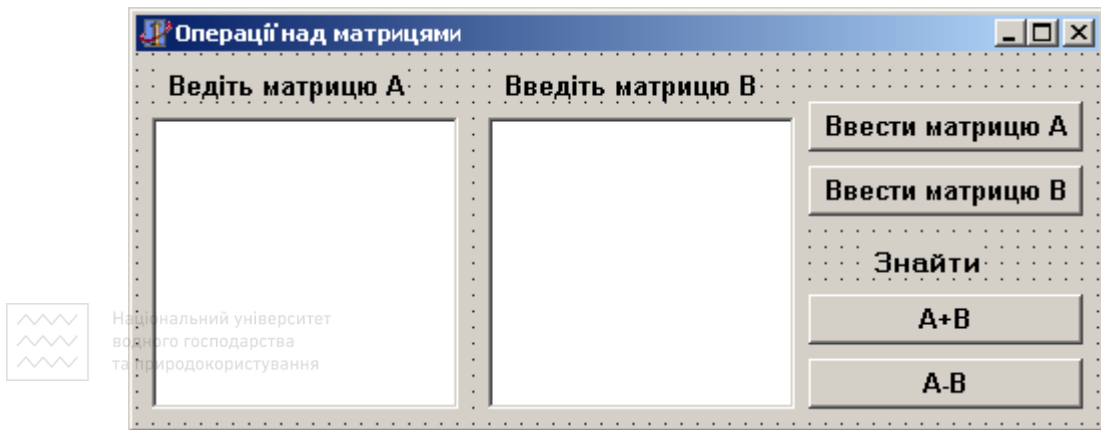


Рис.5.33. Вікно форми для введення даних

Третя форма (рис.5.34) створена для виведення результату програми: *Label1.Caption = 'Label1'*

*Label1.Visible = False*

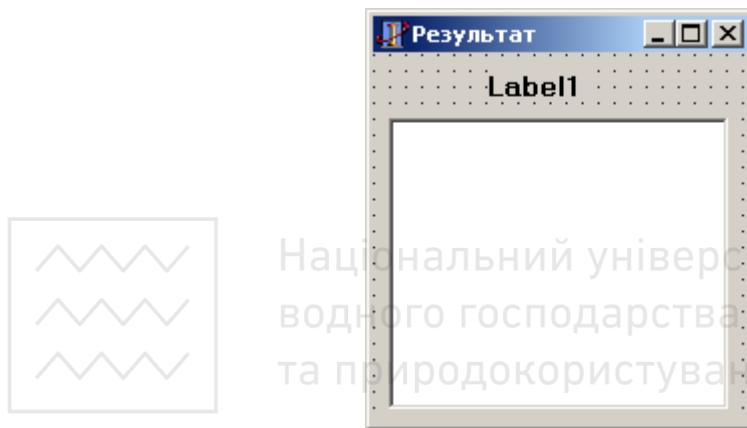


Рис. 5.34 Форма для виведення результату прикладу 5.9

*Етап програмування проекту:*

***unit Unit1;***

***interface***

***uses***

***Windows, Messages, SysUtils, Variants, Classes, Graphics,***

***Controls, Forms,***

***Dialogs, StdCtrls;***

***type***

***TForm1 = class(TForm)***

***Memo1: TMemo;***

***Label1: TLabel;***

***Label2: TLabel;***

```

Button1: TButton;
Button2: TButton;
Button3: TButton;
Memo2: TMemo;
Button5: TButton;
Label4: TLabel;
procedure Button3Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
end;
var
  Form1: TForm1;
  rowcount,colcount:integer;
implementation

uses ElemMas, Unit3;
const N=10;
  M=10;
var t,s,r,q: array [1..N,1..M] of real;
{SR *.dfm}
// Процедура підрахунку суми двох матриць:
procedure TForm1.Button3Click(Sender: TObject);
var ij:integer;
temp:string;
begin
  Form3.Memo3.Clear;
  for i:=1 to rowcount do
    begin
      temp:=' ';
      for j:=1 to colcount do
        begin
          s[i,j]:=r[i,j]+q[i,j];
          temp:=temp + floattostr(s[i,j])+' ';
        end;
      Form3.Label1.Visible:=true;
      Form3.Label1.Caption:='A+B';
      Form3.Show;
      Form3.Memo3.Lines.Add(temp);
    end;
  end;

```



```

    end;
end;
//Процедура, яка забезпечує введення даних матриці B:
procedure TForm1.Button2Click(Sender: TObject);
var i,j:integer;
buf,st :string;
begin
    i:=1;
    repeat st:=getline(memo2.Text,i);
    if length(st)<>0
    then
    begin
        j:=1;
        repeat buf:=getsubstr(st,j);
        if length(buf)<>0 then
        try r[i,j]:=strtofloat(buf)
        except on EConvertError do
        begin
            showmessage('потрібно ввести число');
            memo2.SetFocus;
            exit;
        end
        end
        else begin
            showmessage('потрібно
            ввести'+inttostr(colcount)+'стовпців.');
            memo2.SetFocus;
            exit;
        end;
        j:=j+1;
        until (j>colcount) or (length(buf)=0)
        end
        else begin
            showmessage('потрібно ввести'+inttostr(rowcount)+'рядків.');
            memo2.SetFocus;
            exit;
        end;
        i:=i+1
        until (i>rowcount) or (length(st)=0);
    end;
//Процедура, яка забезпечує введення даних матриці A:
procedure TForm1.Button1Click(Sender: TObject);
var i,j:integer;

```



Національний університет  
водного господарства  
та природокористування



Національний університет  
водного господарства  
та природокористування

```

    buf,st :string;
begin
    i:=1;
    repeat st:=getline(memo1.Text,i);
    if length(st)<>0
    then
    begin
        j:=1;
        repeat buf:=getsubstr(st,j);
        if length(buf)<>0 then
            try q[i,j]:=strtofloat(buf)
            except on EConvertError do
            begin
                showmessage('потрібно ввести число');
                memo1.SetFocus;
                exit;
            end
            end
        else begin
            showmessage('потрібно ввести'+inttostr(colcount)+'стовпців. ');
            memo1.SetFocus;
            exit;
            end;
        j:=j+1;
        until (j>colcount) or (length(buf)=0)
    end
    else begin
        showmessage('потрібно ввести'+inttostr(rowcount)+'рядків. ');
        memo1.SetFocus;
        exit;
        end;
        i:=i+1
        until (i>rowcount) or (length(st)=0);
    end;
// Процедура, яка визначає різницю двох матриць:
procedure TForm1.Button5Click(Sender: TObject);
var i,j:integer;
temp:string;
begin
    Form3.Memo3.Clear;
    for i:=1 to rowcount do
        begin
            temp:=' ';

```





```

for j:=1 to colcount do
begin
s[i,j]:=q[i,j]-r[i,j];
temp:=temp + floattostr(s[i,j])+ ' ';
end;
Form3.Label1.Visible:=true;
Form3.Label1.Caption:='A-B';
Form3.Show;
Form3.Memo3.Lines.Add(temp);
end;
end;
end.

unit Unit2;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, Spin;
type
  TForm2 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Button1: TButton;
    SpinEdit1: TSpinEdit;
    SpinEdit2: TSpinEdit;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
  var
    Form2: TForm2;
implementation
uses Unit1;
  {$R *.dfm}
  // Процедура, яка дозволяє задавати розмірність матриць:
  procedure TForm2.Button1Click(Sender: TObject);
  var k:integer;
  begin
    Form1.Show;
    unit1.rowcount:=spinedit1.Value;
    unit1.colcount:=spinedit2.Value;
  end;

```



Національний університет  
водного господарства  
та природокористування



Національний університет  
водного господарства  
та природокористування

*end.*  
**unit ElemMas;**  
**interface**  
**function GetSubStr(st:string; n:integer):string ;**  
**function GetLine(st:string; n:integer):string ;**  
**implementation**  
 {видаляє з рядка *st* підрядок (послідовність символів між пропусками) із вказаним номером *n*, результат – підрядок або порожній рядок}



Національний університет  
 водного господарства  
 та природокористування

**function GetSubStr(st:string; n:integer):string ;**  
**var**  
**p:integer;**  
**begin** //видалення початкових пропусків у *st*  
**while (pos (' ',st)=1) do delete(st,1,1);**  
**if n>1 then**  
**repeat p:=pos(' ',st);**  
**if p<>0 then**  
**begin**  
 //видалення підрядка через часткове копіювання  
**st:=copy(st,p+1,length(st)-p);**  
 //видалення початкових пропусків у частині рядка, що  
 //залишається  
**while (pos (' ',st)=1) do delete(st,1,1);**  
**n:=n-1;**  
**end;**  
**until (n=1) or (p=0);** //формування результату  
**if n>1 then**  
**result:=""**  
**else begin**  
**p:=pos(' ',st);**  
**if p<>0 then**  
**result:=copy(st,1,p-1)**  
**else result:=st;**  
**end;**  
**end;**  
 {повертає підрядок (послідовність символів з #13 на кінці) з вказаним номером *n*, результат – підрядок або порожній рядок}  
**function GetLine(st:string; n:integer):string ;**  
**var**  
**p:integer;**  
**begin** //видалення початкових пропусків у *st*  
**while (pos (' ',st)=1) do delete(st,1,1);**



Національний університет  
 водного господарства  
 та природокористування

```

if n>1 then
  repeat p:=pos(#13,st);
  if p<>0 then
    begin
      st:=copy(st,p+2,length(st)-p);
      // роздільник – символи з кодом 13 та 10
      // видалення початкових пропусків у рядку-залишку
      while (pos(' ',st)=1) do delete(st,1,1);
      n:=n-1;
    end;
  until (n=1) or (p=0); //формування результату
if n>1
  then result:=""
  else
    begin p:=pos(#13,st);
    if p<>0 then
      result:=copy(st,1,p-1)
    else result:=st;
    end;
end;
end.

```

Результат виконання проекту:

Задаємо розмірність матриць(рис.5.35):

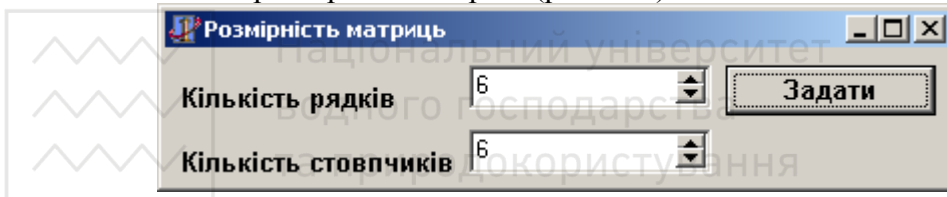


Рис.5.35. Введення розмірності матриць

Введемо значення для матриць (рис.5.36):

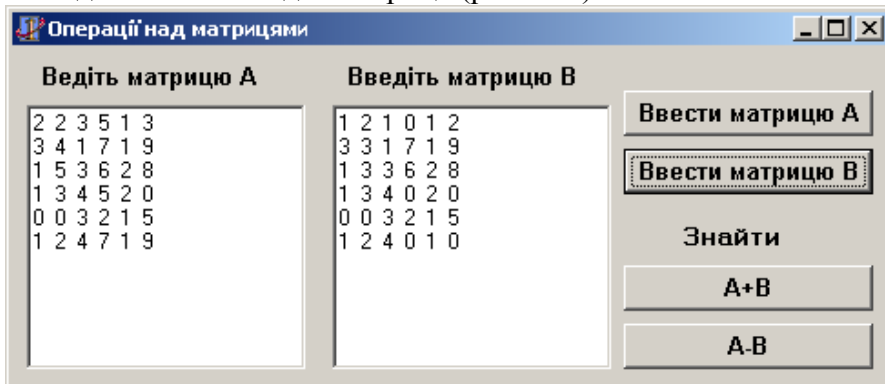


Рис. 5.36. Введення вхідних даних

При натисканні на кнопку 'А-В' у окремому вікні з'являється результат операції (рис.5.37).

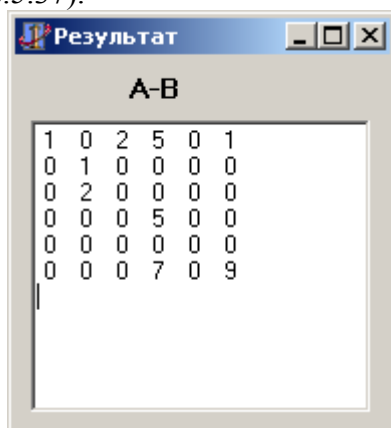


Рис. 5.37. Результат виконання проекту прикладу 5.9

### Завдання для виконання лабораторної роботи:

В середовищі програмування **Delphi** розробити програми, що реалізують нижченаведені варіанти завдань. Забезпечити виконання наступних вимог:

- створити головне меню (використати головне меню **TMainMenu**);
- розмірність матриць вводити на окремій формі (компонента **TSpinEdit**);
- для введення матриць використати компоненти **TMemo** та **TStringgrid** (для вибору способу введення використати компоненту **TTabControl**);
- виведення результатів має здійснюватися на окрему форму (відповідно до способу введення даних).

### Варіанти завдань:

1. Дано дві матриці:  $A = \{a_{ij}\}$ ,  $B = \{b_{ij}\}$ . Створити матрицю  $C = \{c_{ij}\}$  за

$$\text{правилом } C_{ij} = \begin{cases} 1, & a_{ij} = b_{ij} \\ 0, & a_{ij} \neq b_{ij} \end{cases}.$$

2. Дано дві матриці  $A$  і  $B$  ( $m \times n$ ). Знайти  $C = A \cdot B$ .

3. Дана матриця  $A$  та вектор-стовпець  $B$  з  $n$ -вимірного простору. Знайти  $C = A \cdot B$ .

4. Дано дві матриці  $A$  і  $B$  ( $m \times n$ ). Якщо матриця  $C = A - B$  буде симетричною, то зафарбувати її комірки в зелений колір, в протилежному випадку – в червоний.

5. Дано дві квадратні матриці  $A$  і  $B$ . Знайти  $C = (A + E) \cdot (B + E)$ , де  $E$  – одинична матриця.

6. Дана квадратна матриця  $A$  розмірності  $n \times n$ . Визначити: скільки її стовпців впорядковані по зростанню та скільки її стовпців впорядковані за спаданням.

7. Дана матриця  $B (n \times m)$  дійсних чисел. Транспонувати дану матрицю та сформуванати масив  $C = \{c_i\}, i = \overline{1, n}$ , де кількість додатних елементів  $i$  – го стовпця транспонованої матриці.

8. В квадратній матриці  $C (n \times n)$  підрахувати та вивести на екран окремо суму додатних і від’ємних елементів, підрахувати при цьому їх кількість.

9. Обчислити елементи матриці  $C = \{c_{ij}\}$ , якщо

$$c_{ij} = \begin{cases} a_{ij}, & |a_{ij}| > |b_{ij}| \\ b_{ij}, & |a_{ij}| \leq |b_{ij}| \end{cases}, \text{ де } a_{ij} \text{ – елементи матриці } A \text{ та } b_{ij} \text{ –}$$

елементи матриці  $B$ .

10. Провести перетворення матриці  $A$  таким чином: додатні елементи замінити на число 2, від’ємні – на 1, нульові – на 3.

11. Дано матриця  $B (n \times n)$ . Перетворити матрицю, замінивши додатні елементи – номером рядка, в якому вони знаходяться, від’ємні елементи – номером стовпчика, в якому вони знаходяться, нульові елементи – сумою відповідного рядка та стовпчика.

12. Дана матриця  $A = \{a_{ij}\} (n \times n)$ . Знайти  $c = \frac{a_{\min} + a_{\max}}{2}$ , де  $a_{\min} = \min \{a_{ij}\}$ ,  $a_{\max} = \max \{a_{ij}\}$ .

13. Дана матриця  $A (m \times n)$ . Знайти мінімальний елемент для даної матриці та вивести його. Замінити елементи матриці, що лежать нижче головної діагоналі, на мінімальний елемент.

14. Дана матриця  $A (m \times n)$ . Знайти кількість від’ємних елементів, що лежать нижче головної діагоналі.

15. Дана матриця  $A (m \times n)$ . Отримати нову матрицю шляхом ділення всіх її елементів на найбільший по модулю елемент.

16. Дана матриця  $A (m \times n)$ . Замінити всі елементи даної матриці, сума індексів яких парна, добутками відповідних індексів.

17. Дана квадратна матриця  $A$  розмірності  $n \times n$  ( $n$  – парне число), яка складається з цілих чисел. Якщо в парних стовпчиках кількість непарних чисел парна, то дані в стовпчиках замінити на дані непарних рядків, а якщо ні – то парних рядків.

18. Дана матриця  $A (m \times n)$ . Здійснити перетворення наступним чином: якщо сума індексів елементу матриці більша за сам елемент – то замінити значення даного елементу на одиницю, рівна значенню елементу – то замінити значення даного елементу на два, менша – на три. Підрахувати

кількість 1, 2, 3 в матриці. На формі зобразити круг і зафарбувати його наступним чином, якщо найбільшу кількість в матриці складають

$$\begin{cases} 1, & \text{червоним;} \\ 2, & \text{жовтим;} \\ 3, & \text{зеленим.} \end{cases}$$

Якщо максимум складають декілька номерів, то круг замалювати синім кольором.



19. Дана квадратна матриця  $A$  розмірності  $n \times n$ . Передбачити введення двох чисел  $k$  та  $j$ , де  $k, j \leq n$ . Якщо добуток елементів  $k$ -го стовпчика менше добутку елементів  $j$ -го рядка, то поміняти їх місцями, інакше – провести заміну елементів  $k$ -го стовпчика на добуток елементів  $j$ -го рядка.

20. Дана квадратна матриця  $A$  розмірності  $n \times n$  ( $n$  – парне число), яка складається із цілих чисел. Якщо в непарних рядочках кількість парних чисел непарна, то дані рядки поміняти з непарними стовпчиками, інакше – з парними.

21. Дано дві матриці  $A$  і  $B$  ( $m \times n$ ). Сформувати матрицю  $C = \{c_{ij}\}$  за правилом

$$C_{ij} = \begin{cases} 0, & a_{ij} = b_{ij}, a_{ij} - \text{непарне} \\ 1, & a_{ij} > b_{ij}, b_{ij} - \text{парне} \\ -1, & a_{ij} < b_{ij}, b_{ij} - \text{непарне} \\ 10, & \text{в протилежних випадках} \end{cases}$$

22. Дана матриця  $A$  ( $m \times n$ ). Сформувати матрицю  $B = \{b_{ij}\}$ , де  $b_{ij}$  дорівнює остачі від ділення числа  $a_{ij}$  на число  $i$ , та матрицю  $C = \{c_{ij}\}$ , де  $c_{ij}$  дорівнює сумі парних елементів матриці  $B$  в  $j$ -му стовпці.

23. Дана квадратна матриця  $A$  розмірності  $n \times n$ . Сформувати два масиви  $B = \{b_i\}$  та  $C = \{c_j\}$  за правилом

$$b_i = \begin{cases} 1, & \text{якщо } i\text{-й рядок матриці } A \text{ утворює арифметичну прогресію;} \\ 0, & \text{в протилежному випадку;} \end{cases}$$

$$c_j = \begin{cases} 1, & \text{якщо } j\text{-й стовпчик матриці } A \text{ утворює геометричну прогресію} \\ 0, & \text{в протилежному випадку.} \end{cases}$$

24. Дана квадратна матриця  $A$  розмірності  $n \times n$ . Сформувати матрицю  $C = \{c_{ij}\}$ , де  $c_{ij}$  – кількість цифр в числі, записаному в  $a_{ij}$ . Якщо кількість цифр перевищує 4, то  $c_{ij} = 0$ .

25. Дана квадратна матриця  $A$  розмірності  $n \times n$ . Сформувати два масиви  $B = \{b_i\}$  та  $C = \{c_j\}$  за правилом

$$b_i = \begin{cases} 1, & \text{якщо відповідна паралельна діагональ до головної діагоналі матриці } A \\ & \text{є зростаючою послідовністю;} \\ -1, & \text{якщо спадною;} \\ 0, & \text{в протилежному випадку;} \end{cases}$$

$$c_j = \begin{cases} 1, & \text{якщо відповідна паралельна діагональ до побічної діагоналі матриці } A \\ & \text{є спадною послідовністю;} \\ -1, & \text{якщо зростаючою;} \\ 0, & \text{в протилежному випадку;} \end{cases}$$

#### Контрольні запитання:

1. Назвіть принципи роботи з масивами?
2. Який масив називають багатовимірним?
3. Який масив називають статичним?
4. Що таке динамічний масив?
5. Вкажіть властивості компонент *TStringGrid* та *TMemo*.
6. Назвіть основні властивості компоненти *TTabControl*.

#### Лабораторна робота №6

**Тема: Розробка додатків з використанням графічних можливостей Delphi.**

#### Мета роботи:

- Навчитись працювати з візуальними та невізуальними компонентами сторінки *Additional: TImage, TChart*, сторінки *Dialogs: TColorDialog, TOpenPictureDialog, TSavePictureDialog*, сторінки *System: TPaintBox*.
- Закріпити навички роботи з візуальними компонентами сторінки *Standard: TEdit, TButton, TLabel*, сторінки *Additional: TBitBtn, TLabeledEdit*.
- Розглянути призначення, основні властивості та методи роботи з компонентами *TImage, TChart, TColorDialog, TOpenPictureDialog, TSavePictureDialog, TPaintBox*.
- Навчитись будувати графіки та діаграми.

### Теоретичні відомості:

Для виконання завдань даної лабораторної роботи пропонується використати компоненти *TImage*, *TChart*, *TColorDialog*, *TOpenPictureDialog*, *TSavePictureDialog*, *TPaintBox*. Призначення, основні властивості та методи роботи з ними викладені нижче. При написанні програм пропонується також використати компоненти, які були розглянуті в теоретичних відомостях попередніх лабораторних робіт.



#### Компонента *TColorDialog*

##### Призначення:

Компонента *TColorDialog* використовується для створення вікна діалогу „Колір” – вибір кольору.

##### Основні властивості:

Компонента *TColorDialog* викликає стандартне діалогове вікно вибору кольору, в якому користувач може вибрати колір із базової палітри, чи розкрити додаткову панель (натиснувши кнопку „Определить цвет>>”), на якій можна проводити синтез кольорів (Рис.5.38).

Основна властивість компоненти *TColorDialog* – *Color*, яка відповідає кольору, який вибрано користувачем в діалоговому вікні. Дану властивість можна задати також під час проектування або програмним чином, якщо потрібно задати початкове наближення кольору. Властивість *CustomColors* типу *TStrings* дозволяє задати колір на замовлення із додаткової палітри.

Кожний колір визначається стрічкою виду

*<Ім'я кольору>=<шістнадцяткове представлення кольору>*.

Імена кольорів задаються від *ColorA* (перший колір) до *ColorP* (шістнадцятий колір).

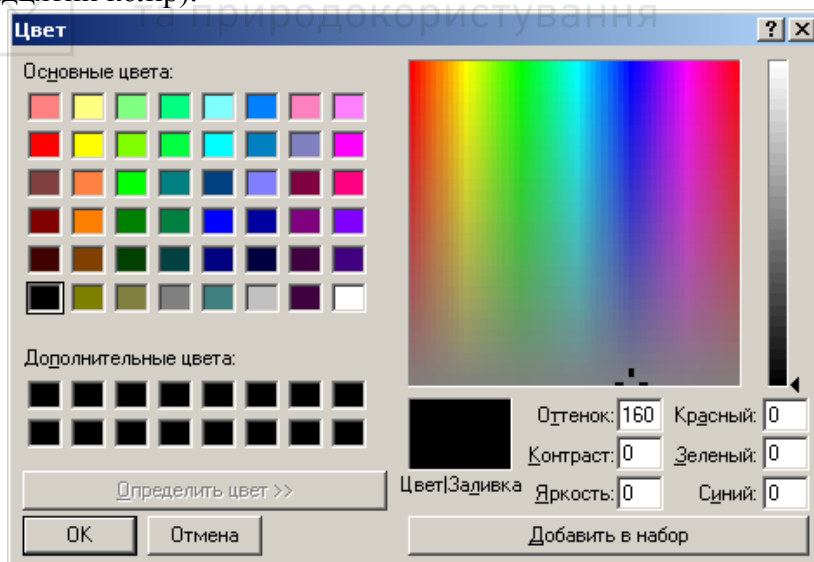


Рис.5.38. Вікно палітри кольорів



Властивість **Options** має наступні опції, які по замовчуванню виключені:

- **cdFullOpen** – надає можливість відображати відразу при відкритті діалогового вікна додаткову панель визначення кольорів за замовленням;
- **cdPreventFullOpen** – дозволяє заборонити появу в діалоговому вікні кнопки „Определить цвет>>”;
- **cdShowHelp** – дозволяє додати в діалогове вікно кнопку „Справка”;
- **cdSolidColor** – вказує Windows використовувати суцільний колір, найближчий до вибраного;
- **cdAnyColor** – дозволяє користувачу вибирати несуцільні кольори (кольори можуть бути нерівними).

Наприклад, потрібно змінювати колір форми. Це можна реалізувати оператором

```
if ColorDialog1.Execute then Form1.Color:=ColorDialog1.Color;
```

### Компоненти **TOpenPictureDialog**, **TSavePictureDialog**

Призначення:

Компонента **TOpenPictureDialog** призначена для створення діалогового вікна „Відкрити малюнок”, що відкриває графічний файл; компонента **TSavePictureDialog** призначена для створення діалогового вікна „Зберегти малюнок”, яке забезпечує збереження зображення у графічному файлі.

Основні властивості:

Компоненти **TOpenPictureDialog** та **TSavePictureDialog** мають аналогічні властивості, як і компоненти **TOpenDialog** та **TSaveDialog**. Відрізняються тільки властивістю **Filter**, де визначаються всі типи графічних файлів, з якими може працювати діалогове вікно ( задається по замовчуванню). В редакторі фільтрів (Рис.5.39) можна редагувати визначені типи файлів, видаляти, перейменовувати:



Національний університет  
водного господарства  
та природокористування

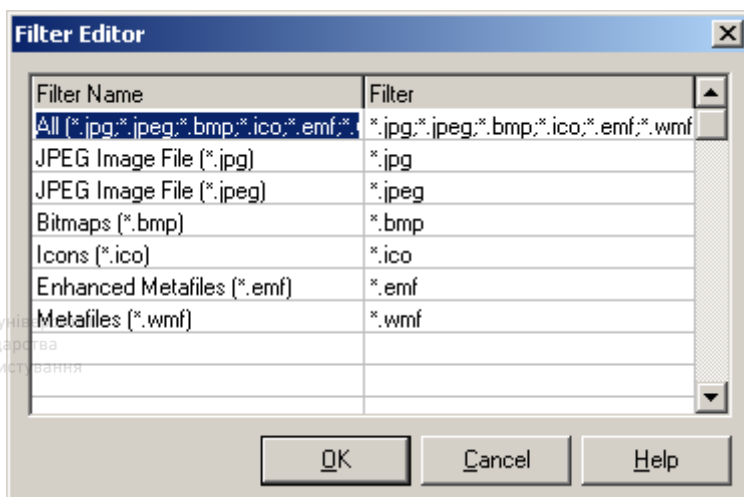


Рис.5.39. Редактор фільтрів

### Компонент TColorBox

#### Призначення:

Випадаючий список для вибору кольору.

#### Основні властивості:

Інколи користувачу зручніше вибирати колір із випадаючого списку, що реалізується компонентою **TColorBox**. Властивість даної компоненти **Style** є множиною, елементи якої визначають, які категорії кольору представлені у списку (дана властивість може приймати наступні значення: **cbStandardColors**, **cbExtendedColors**, **cbSystemColors**, **cbIncludeNone**, **cbIncludeDefault**, **cbCustomColor**, **cbPrettyNames**).

В дані списки, окрім різних кольорів, можуть включати стрічки такі, як **clDefault** – колір компоненти по замовчуванню, **clNone** – колір, що залежить від версії Windows (наприклад, білий колір для Windows 98 та чорний для Windows 2000\XP\NT).

Властивість **DefaultColorColor** визначає, який колір матиме квадратик в списку стрічки **clDefault**.

Властивість **NoneColorColor** визначає, який колір матиме квадратик в списку стрічки **clNone**.

Властивість **Selected** дозволяє дізнатися, який колір вибраний користувачем програми (використовується подія компоненти **OnSelect**, яка виникає в момент вибору користувачем кольору).

Властивість **Colors** є масивом індексів кольорів в списку (індекси починаються з 0).

Властивість **ColorName** – масив стрічок із назвами кольорів.

Властивість **Items** типу **TStrings** містить список всіх стрічок.

Властивість **ItemIndex** визначає індекс стрічки кольору, яка буде показана користувачу в момент початку виконання додатку. Якщо потрібно в перший момент виконання програми показати користувачам

деякий визначений колір, то це можна зробити при реалізації події форми **OnCreate** (*ItemIndex* визначається за допомогою методу **IndexOf**).

### Компонента **TPaintBox**

**Призначення:** Вікно для малювання графічних зображень (область для малювання).

**Основні властивості:**

Основна властивість **Canvas** (канва, полотно). Канва – це область компоненти, на якій можна малювати. Для того, щоб вивести на поверхню об'єкта графічний елемент (прямокутник або відрізок), потрібно застосовувати до властивостей **Canvas** відповідний метод. Властивість **Canvas** є об'єктом **TCanvas**.



### Компонента **TImage**

**Призначення:**

Використовується для відображення графіки: піктограм, бітових матриць та метафайлів.

**Основні властивості:**

Однією з властивостей даної компоненти є властивість **Picture**, яка може містити зображення. Коли зображення завантажується в процесі проектування із файлу до компоненти **TImage**, то воно не просто відображається, але і включається до файлу проекту **\*.DPR** та потім приєднується до файлу виконання, таким чином файл **\*.EXE** може бути достатньо великим.

Щоб під час проектування завантажити зображення, потрібно біля властивості **Picture** натиснути на багатокрапку, в результаті чого з'явиться вікно **Picture Editor** (рис.5.40).

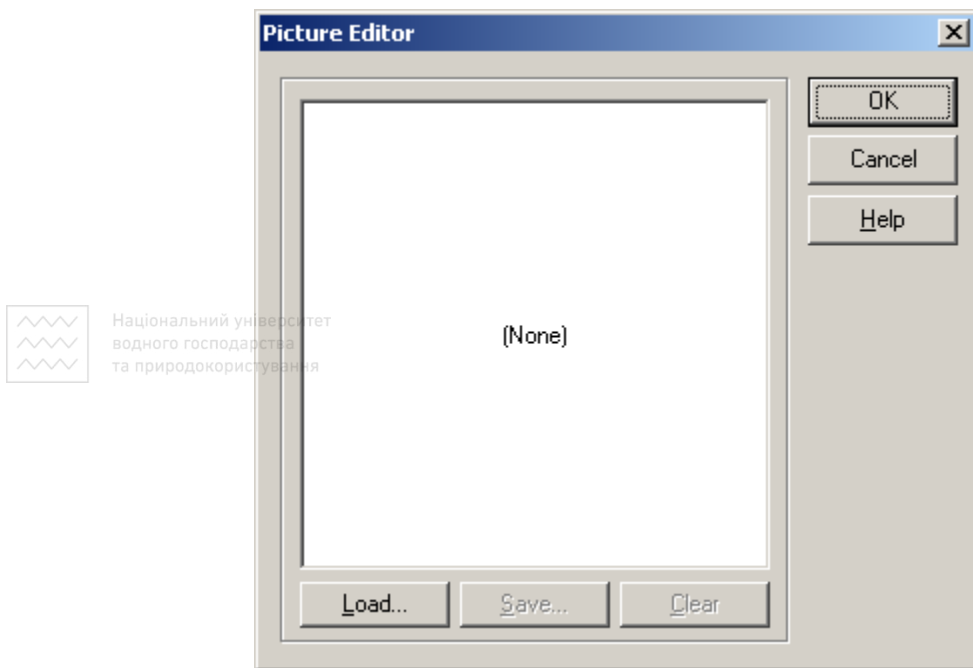


Рис.5.40. Вікно *Picture Editor*

В даному вікні при натисканні на кнопку *Load* відкриється вікно *Load Picture*, де можна завантажити зображення (рис.5.41).

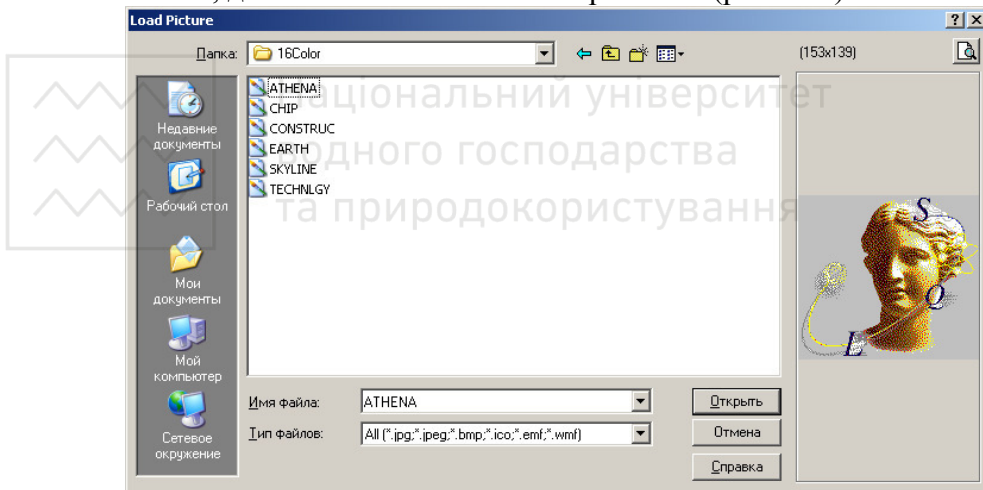


Рис.5.41. Вікно завантаження файлів

Після завантаження файлу потрібно натиснути на кнопці ОК. В результаті в компоненті *Image* буде відображено вибране зображення (рис.5.42).

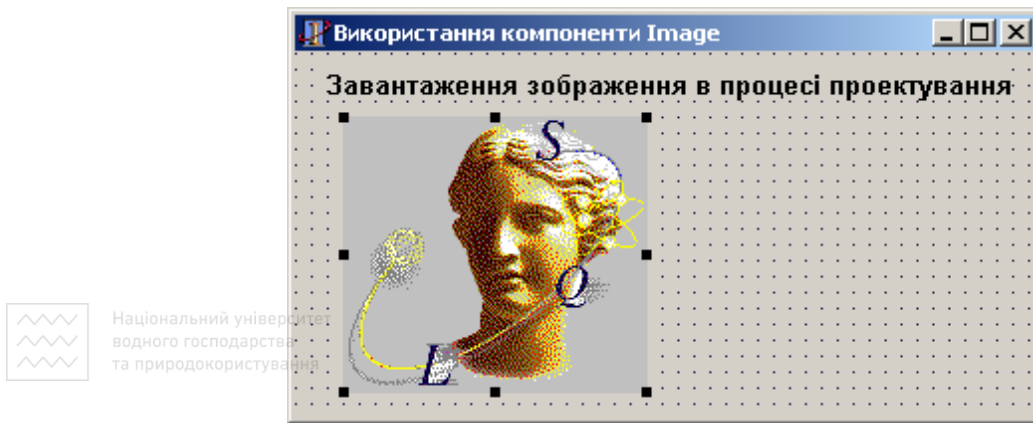


Рис.5.42. Зображення, завантажене в процесі проектування

Це дає можливість розроблювати додатки без окремих графічних файлів. Хоча існує можливість завантажувати в *TImage* і зовнішні графічні файли в процесі виконання додатку.

Основна властивість є *Canvas* (канва, полотно), за допомогою якої можна виводити різноманітну графічну інформацію безпосередньо на зображення.

Властивість *AutoSize* компоненти *TImage* може примати значення *true* або *false*. Якщо *AutoSize=true*, то при невідповідності розмірів зображення та компоненти *TImage*, автоматично розміри компоненти підгоняться під розміри зображення. Якщо властивість *AutoSize=false*, то зображення може не поміститися в компоненту або, навпаки, площа компоненти може бути на багато більша за площу зображення.

Властивість *Stretch* визначає, чи дозволяється підігнати рисунок під розмір компоненти (дана властивість не діє на піктограми, які не можуть змінювати свої розміри). Якщо властивості *Stretch* надати значення *true*, при умові, що *AutoSize=false*, то побачимо, що зображення займе всю площу компоненти *TImage*.

Зазначимо, що *Stretch* слід надавати значення *true* не для рисунків, а для узорів. Якщо ж дві останні властивості встановлені в *false*, то для зображення, що завантажуються до компоненти, та переважає за розмірами, його частина, яка не потрапила до області компоненти *TImage*, відсікається

Властивість *Center*, якщо приймає значення в *true*, то відбувається центрування зображення відносно площі *Image*, якщо розмір компоненти більший за розмір рисунка.

Досить важливим є питання про формати графічних файлів. Файли зображень повинні мати формат *bmp* (bitmap), *wmf* (windows metafile), *emf* (enhanced metafile), *ico* (icon). Також *Delphi* може працювати із файлами у форматі *jpeg* або *jpg*.

**Бітова матриця** (файл с розширенням *.bmp*) відображає колір кожного пікселю в зображенні.

**Піктограми** (файли із розширенням *.ico*) – це маленькі бітові матриці. Використовуються для позначення значків в додатках, в кнопках швидкого доступу, в пунктах меню, в різноманітних списках.

**Метафайли (Metafiles)** зберігають не послідовність бітів, із яких складається зображення, а інформацію про спосіб створення зображення (послідовність команд малювання).

Властивість **Picture** є об'єктом із своїм набором властивостей та методів. Зображення можна завантажувати із різних джерел, а саме:

- із графічного файлу;
- із іншої об'єкта, який містить зображення;
- із файлу ресурсів.

Для завантаження зображення із файлу використовується метод **LoadFromFile (const FileName: String)**, де **FileName** визначає ім'я файлу зображення. Завантаження файлів можна здійснювати із графічних файлів, які мають розширення *\*.bmp*, *\*.wmf*, *\*.emf*, *\*.icon*, *\*.jpg*

Для завантаження файлу на вибір користувача застосовують компоненту **TOpenPictureDialog**, яка є не візуальною. Таким чином, через діалогове вікно можна вибрати та завантажити зображення. Наприклад, тіло процедури обробки події фіксації миші на кнопці матиме вигляд:

```
if OpenPictureDialog1.Execute then
```

```
begin
```

```
Image1.Picture.LoadFromFile(OpenPictureDialog1.FileName)
```

```
Form1.ClientHeight := Image1.Height+10;
```

```
Image1.Top := Form1.ClientRect.Top +  
(Form1.ClientHeight-Image1.Height) div 2;
```

```
Form1.ClientWidth := Image1.Width+10;
```

```
Image1.Left := Form1.ClientRect.Left +  
(Form1.ClientWidth-Image1.Width) div 2;
```

```
end;
```

Таким чином, завантажується у властивість **Picture** компоненти **Image1** файл, який вибраний користувачем. Діалогове вікно відкриття файлу відкривається при фіксації миші. Реалізовано також автоматичне налаштування форми під розміри зображення. Властивість **AutoSize** встановлюють як **true**.

Властивість **Transparent** визначає прозорість зображення (діє тільки для бітових матриць). Якщо **Transparent= true**, то зображення компоненти

**Image** стає прозорим. Дана властивість використовується для накладання зображень один на одного.

Наприклад, розташуємо на формі дві компоненти **Image** та розмістимо їх так, щоб вони перекривали один одного. Якщо **Transparent=false** та компонента **Image2** накриває компоненту **Image1**, то зображення другої компоненти закриває частину зображення першої компоненти (рис.5.43).

Якщо **Transparent=true** та компонента **Image2** накриває компоненту **Image1**, то зображення другої компоненти стає прозорим і не закриває зображення першої компоненти (рис.5.44).

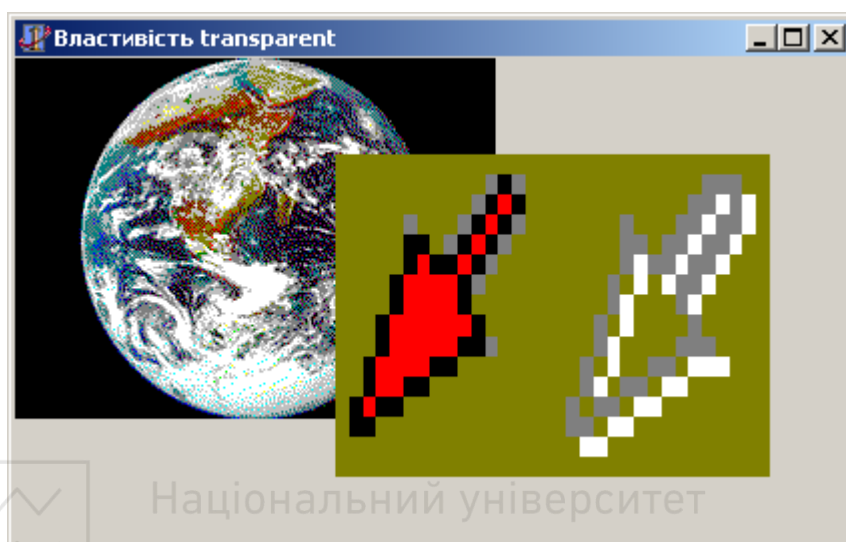


Рис.5.43. Властивість **Transparent=false**

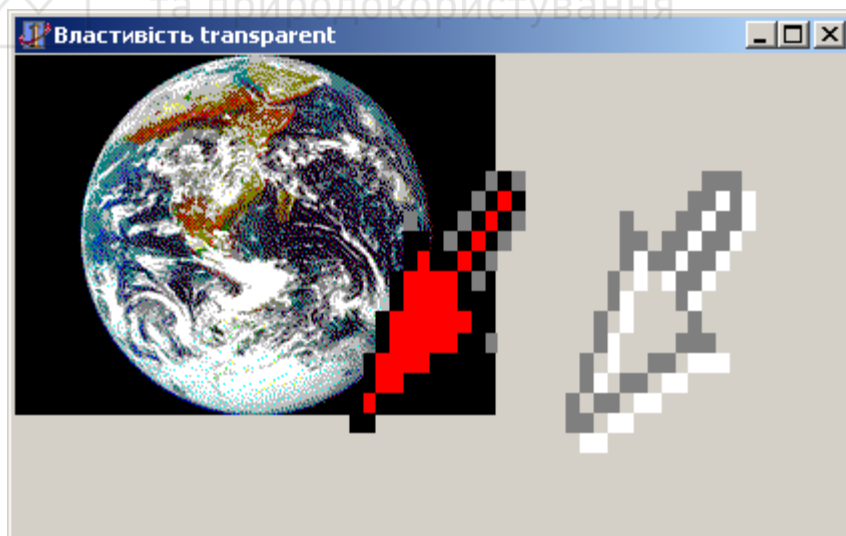


Рис.5.44. Властивість **Transparent=true**

## Компонента *TChart*

### Призначення:

Компонента дозволяє будувати різні діаграми та графіки для відображення різного роду інформації в графічній формі

### Основні властивості:

Компонента *TChart* є контейнером об'єктів *Series* нащадків класу *TChartSeries*. Кожний такий об'єкт – це є серія даних, що характеризуються певним стилем відображення: тим чи іншим графіком або діаграмою. Кожна компонента *TChart* може включати декілька серій. При відображенні графіка кожна серія буде відповідати одній кривій на графіку. При відображенні діаграм декілька серій можуть відповідати різним типам діаграм. Таке зображення використовується при відображенні однакових даних в різному представленні. Таким чином, користувачу буде надана можливість вибрати тип діаграми для перегляду даних.

Властивість *AllowPanning* визначає можливість користувача, нажимаючи праву кнопку миші, прокручувати частину графіка, що ним переглядається, під час виконання (можливі значення: *pmNone* – прокручування заборонено, *pmHorizontal*, *pmVertical*, або *pmBoth* – дозволено відповідно прокручування тільки в горизонтальному напрямку, тільки у вертикальному або в обох напрямках).

Властивість *AllowZoom* дозволяє змінювати під час виконання масштаб зображення, вирізаючи фрагменти діаграми або графіка курсором миші (якщо при виділенні фрагмента рамка малюється вправо та вниз, то фрагмент розтягнеться на все поле графіка; якщо ж рамка малюється доверху та вліво, то відновлюється вихідний масштаб).

Властивість *Title* визначає заголовок діаграми.

Властивість *Foot* дозволяє визначити підпис під діаграмою (по замовчуванню текст відсутній). Текст підпису визначається підвластивістю *Text*.

Властивість *Frame* визначає рамку навколо діаграми.

Властивість *Legend* визначає легенду діаграми (список позначень).

Властивості *Margin Left*, *MarginRight*, *MarginTop*, *MarginBottom* – значення лівого, правого, верхнього та нижнього полів.

Властивості *BottomAxis*, *LeftAxis*, *RightAxis* визначають характеристики відповідної нижньої, лівої та правої осей.

Властивості *LeftWal*, *BottomWall*, *BackWall* визначають характеристики відповідно лівої, нижньої та задньої граней області трьохвимірного зображення графіка.

Властивість *SeriesList* визначає список серій даних, що відображаються в компоненті.



Властивість **ViewSd** визначає дозвіл на трьохвимірне зображення діаграми.

Властивість **ViewSDOptions** визначає характеристики трьохвимірного зображення.

Дані властивості можна задавати в редакторі діаграм, який з'являється при натисканні на три крапки біля деяких вище згаданих властивостей в Інспекторі Об'єктів. Також існує можливість виклику редактора діаграм при подвійній фіксації миші на компоненті **Chart** або при натисканні на ній правою кнопкою миші та виборі команди **Edit Chart** у контекстному меню. Редактор діаграм (Рис.5.45) має дві сторінки **Chart** та **Series**.

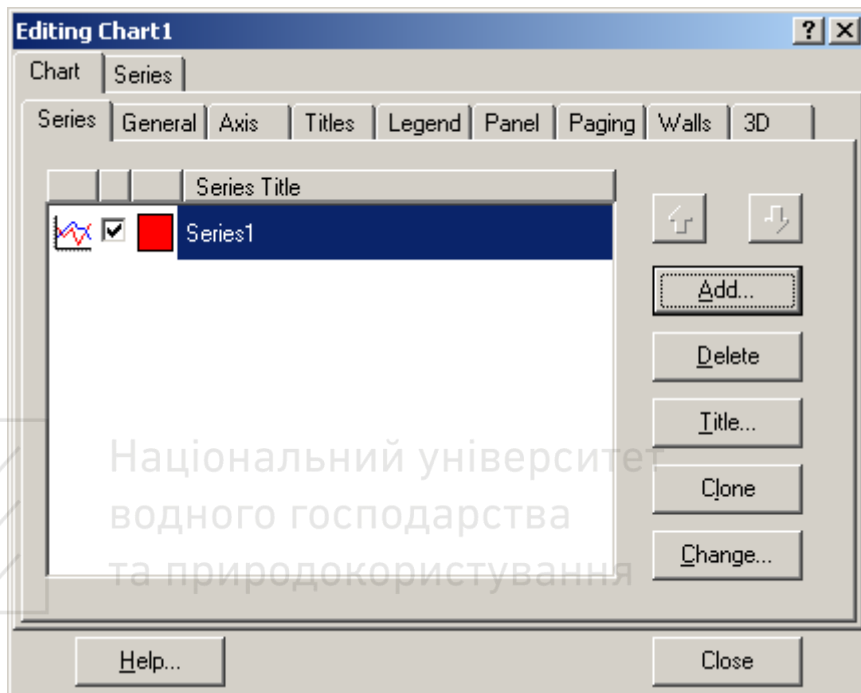


Рис.5.45. Редактор діаграм

Сторінка **Chart** в свою чергу має декілька закладок: закладка **Series**, на якій забезпечується додавання (наприклад, при натисканні на кнопку **Add**, що реалізує додавання серій, з'явиться вікно, в якому можна вибрати тип діаграми, або графіка), видалення, створення клону (серію із ідентичними властивостями) та редагування серій; закладка **Titles**, на якій можна задати заголовок діаграми, закладка **Legend** дозволяє задавати параметри відображення легенди діаграми (списку позначень) або дозволяє зовсім не відображати її; закладка **Panel** визначає вид панелі, на якій відображається діаграма; закладка **3D** дає можливість змінювати зовнішній вигляд діаграми: нахил, зсув, товщина і т.д.

При роботі із редактором діаграм, вибираючи тип діаграми (рис.5.46), в компоненті **TChart** буде відображатися вид діаграми потрібного (вибраного) типу із занесеними в неї умовними даними. Таким чином надається можливість переглядати результати використання різних опцій діаграми з умовними даними.

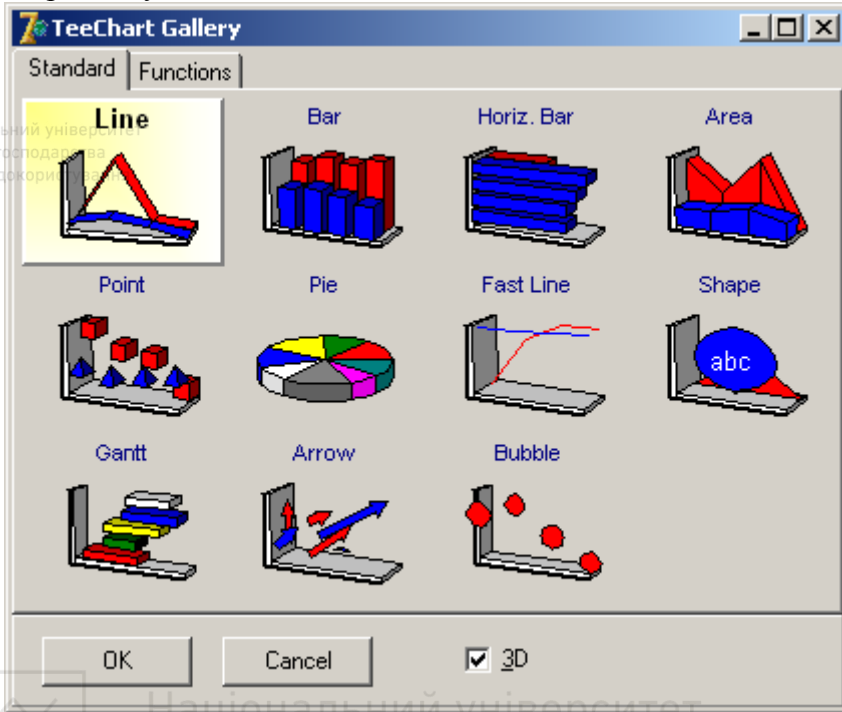


Рис.5.46. Тип діаграм

Сторінка **Series**, що також має ряд закладок, дає можливість вибирати додаткові характеристики відображення серій.

Щоб задати значення, які будуть відображатися на діаграмі, потрібно використовувати методи серій **Series** (табл.5.1).

Таблиця 5.1

Основні методи **Series**

<i>Метод</i>	<i>Призначення</i>
<b>Clear</b>	Очищає серію від занесених раніше даних (потрібно, якщо в процесі роботи додатку дані оновлюються)
<b>Add(Const AValue:Double; Const ALabel: String; AColor:TColor)</b>	Дозволяє додати в діаграму нову точку ( <b>AValue</b> відповідає значенню, що додається, <b>ALabel</b> – мітка, яка буде відображатися на діаграмі в легенді (даний параметр не є обов'язковим, його можна задати пустим), <b>AColor</b> – колір)

Метод	Призначення
<i>AddXY</i> (Const <i>AXValue</i> , <i>AYValue</i> :Double; Const <i>ALabel</i> :String; <i>AColor</i> :TColor)	Дозволяє додати нову точку в графіку функції ( <i>AXValue</i> – аргумент, <i>AYValue</i> відповідає значенню функції, <i>ALabel</i> – мітка, яка буде відображатися на діаграмі в легенді, <i>AColor</i> – колір).
<i>AddY</i> (Const <i>AYValue</i> :Double; Const <i>ALabel</i> :String; <i>AColor</i> :TColor)	Дозволяє додати нову точку в графіку функції ( <i>AYValue</i> відповідає значенню функції, <i>ALabel</i> – мітка, яка буде відображатися на діаграмі в легенді, <i>AColor</i> – колір).

### Приклади розробки Delphi-додатків:

#### Приклад 5.10

В середовищі програмування *Delphi* розробити віконний додаток, за допомогою якого можна було б переглядати графічні файли.

**Етап проектування** інтерфейсу проекту:

Під час проектування (рис.5.47) використовуємо компоненти класу *TButton*, *TImage*, *TOpenPictureDialog*.

Основні властивості використаних компонент:

*Button1.Caption:= 'Файл';*

*Image1.AutoSize:= true;*

*Form1.AutoSize:= true;*

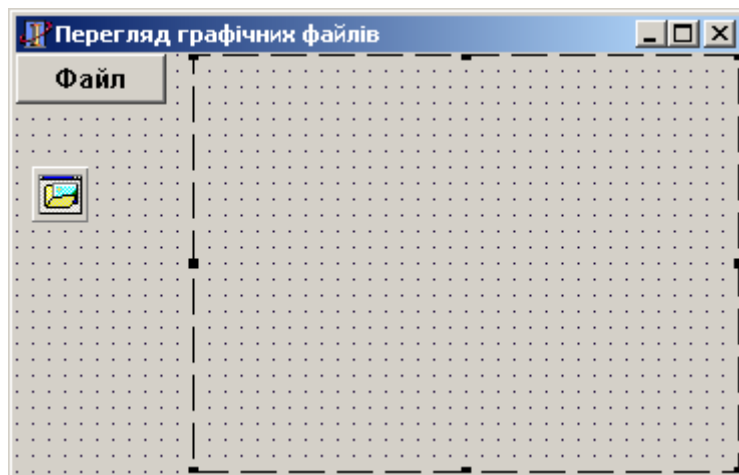


Рис.5.47. Вікно форми проекту прикладу 5.10

### Етап програмування проекту:

Процедура обробки події натискання кнопки „Файл” забезпечує завантаження графічного файлу до компоненти *TImage* та визначає розміри форми залежно від розміру зображення.

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  if OpenPictureDialog1.Execute then
    begin
      Image1.Picture.LoadFromFile(OpenPictureDialog1.FileName);
      Form1.ClientHeight := Image1.Height+10;
      Image1.Top := Form1.ClientRect.Top +
        (Form1.ClientHeight-Image1.Height) div 2;
      Form1.ClientWidth := Image1.Width+80;
      Image1.Left := Form1.ClientRect.Left+
        (Form1.ClientWidth-Image1.Width);
    end;
```

### Результат виконання проекту:

В результаті при використанні програми можна переглядати графічні файли. Відповідно до розміру зображення змінюється розмір форми (рис.5.48, рис.5.49).

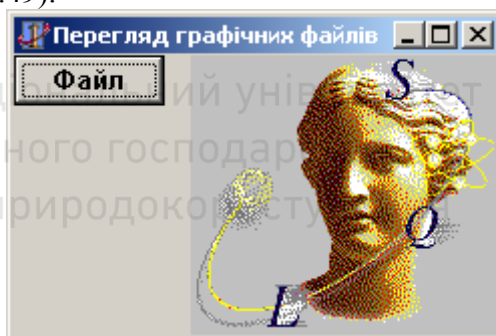


Рис.5.48. Зменшення розмірів форми



Рис.5.49. Збільшення розмірів форми

### Приклад 5.11

В середовищі програмування **Delphi** розробити програму побудови графіка функції на відрізку  $[a,b]$  за допомогою компоненти **TChart**

**Етап проектування** інтерфейсу проекту:

Під час проектування використовуємо компоненти класу **TTabControl**, **TChart**.

Основні властивості використаних компонент:

**Form1.Caption** = 'Побудова графіка функції';

**Form1.Tabs.Strings** = ('y=cosx'  
'y=sinx');

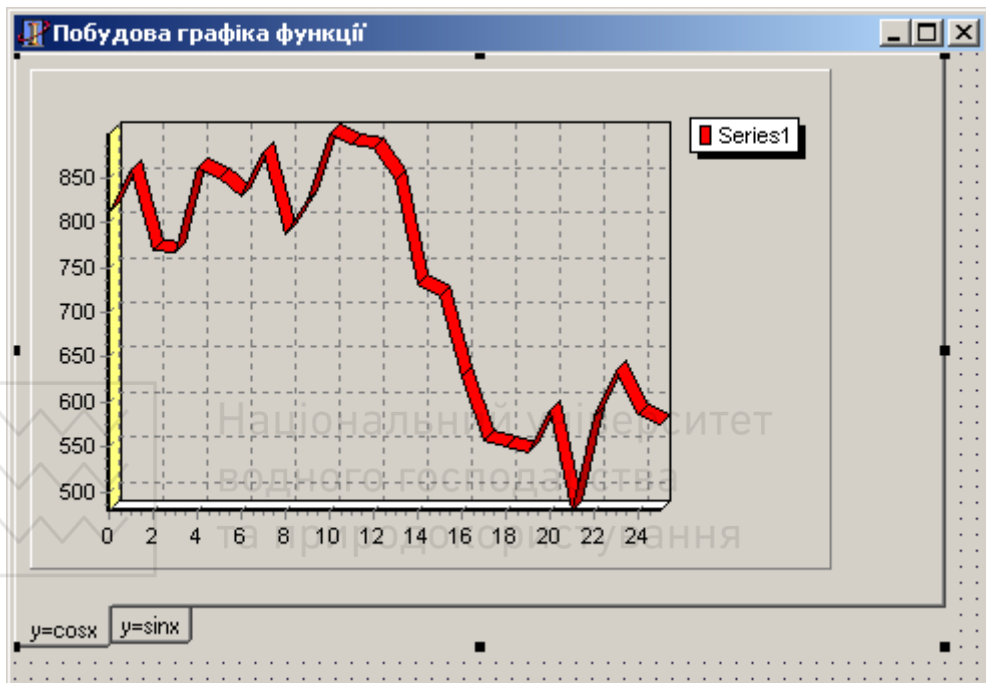


Рис.5.50. Вікно форми проекту прикладу 5.11

**Етап програмування** проекту:

```
procedure TForm1.TabControl1Change(Sender: TObject);
```

```
var i:integer;
```

```
begin
```

```
Chart1.Visible:=true;
```

```
Series1.Clear;
```

```
if TabControl1.TabIndex=0 then
```

```
begin
```

```
Series1.Title:='y=cosx';
```

```
for i:=0 to 100 do
```

```

Series1.AddXY(0.02*2*pi*i,cos(0.02*2*pi*i),
floattostrf(0.02*2*pi*i,ffixed,6,2),clred);
end;
if TabControl1.TabIndex=1 then
begin
Series1.Title:='y=sinx';
for i:=0 to 100 do
Series1.AddXY(0.02*2*pi*i,sin(0.02*2*pi*i),
floattostrf(0.02*2*pi*i,ffixed,6,2),clred);
end
end;

```

**Результат виконання** проекту:

В результаті виконання програми буде побудовано графік функції  $y = \cos x$  (рис.5.51) та графік функції  $y = \sin x$  (рис.5.52).

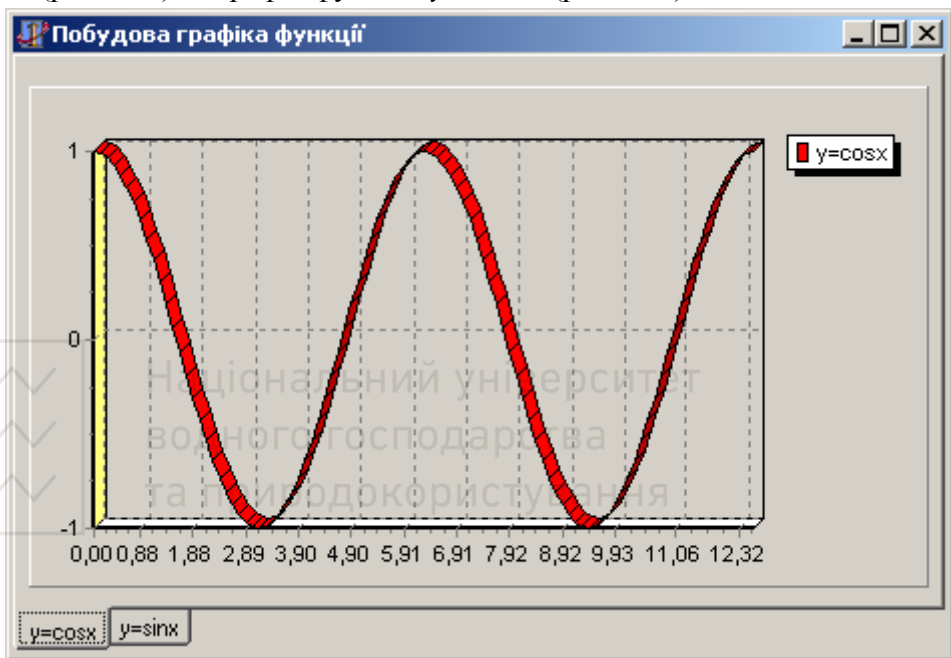


Рис.5.51. Побудова графіка функції  $y = \cos x$

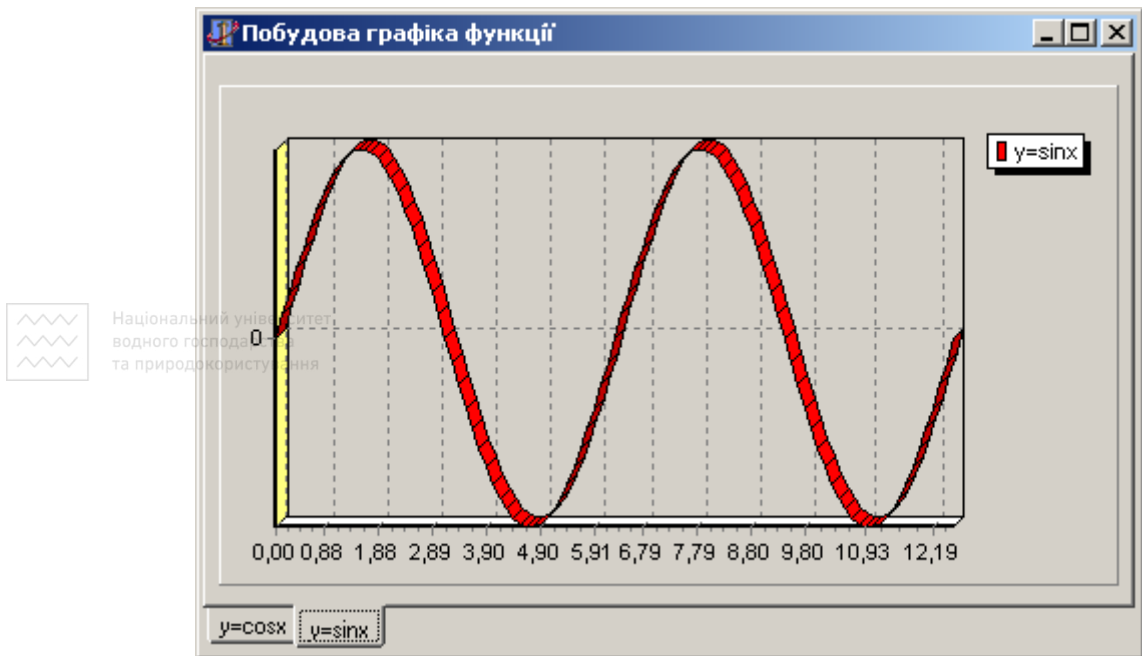


Рис.5.52. Побудова графіка функції  $y = \sin x$

### Завдання для виконання лабораторної роботи:

#### Завдання 1

В середовищі програмування **Delphi** розробити програму побудови графіка функції  $y = f(x)$  на відрізку  $[a, b]$ .

#### Вимоги до проекту:

- всі вхідні дані вводяться на окремих формах (наприклад, задання відрізка  $[a, b]$ );
- розробити масштабування по осях;
- програма повинна містити головне меню;
- передбачити вибір кольору графіка, фону, осей координат (використати компоненту **TColorDialog**);
- графік намалювати не на формі, а на окремій компоненті (використати компоненту **TImage**, або **TPaintBox**).

#### Варіанти завдань:

1.  $y = ||x| - 1|$ ;
2.  $y = \frac{1}{2}(|\sin x| - \sin x)$ ;
3.  $y = \cos x + |\cos x|$ ;

$$4. y = \sin^{\frac{4}{3}} x;$$

$$5. y = \sqrt{x} - \cos x;$$

$$6. y = e^{-x} \sin x;$$

$$7. y = x - \sin x;$$

$$8. y = \arcsin(\sin x)$$

$$9. y = \sin(\sin x);$$

$$10. y = |\cos(2x)|;$$

$$11. y = \frac{1}{3} \sin |x|;$$

$$12. y = 2|\sin x| - 1;$$

$$13. y = 2 + 3|\cos x|;$$

$$14. y = x + \sin |x|;$$

$$15. y = \sin x + \cos x;$$

$$16. y = x \sin x;$$

$$17. y = \frac{\ln x}{x};$$

$$18. y = 2^{\cos x};$$

$$19. y = \sin^3 x;$$

$$20. y = \frac{\sin x}{e^x};$$

$$21. y = \sqrt{x} - \cos x;$$

$$22. y = \frac{1}{2} (|\sin x| - \sin x);$$

$$23. y = \cos x + |\cos x|;$$

$$24. y = \sqrt{2x - x^2};$$

$$25. y = 3 \sin\left(\frac{x}{2}\right) e^{\frac{x}{2}}.$$

### Завдання 2.

В середовищі програмування **Delphi** розробити програму побудови діаграми (використати компоненту **TChart**), яка б відображала інформацію про розподіл доходу вашої сім'ї.

### Контрольні запитання:

1. Які існують методи для малювання?
2. Як зобразити зафарбований прямокутник?
3. Як зробити текстовий надпис?



4. Охарактеризуйте змінні в побудові прямокутника.
5. Як намалювати чорну крапку на екрані?

## Лабораторна робота №7

**Тема:** Робота з файлами в Delphi.



Національний університет  
застосування науки та технологій

**Мета роботи:**

- Навчитись працювати з візуальними та невізуальними компонентами сторінки **Additional: TSpeedButton**, сторінки **Win32: TToolBar**, сторінки **Standard: TEdit, TButton, TLabel, TPanel**, сторінки **Dialogs: TOpenDialog, TSaveDialog**.
- Повторити методи роботи з візуальними та невізуальними компонентами сторінки **Additional: TBitBtn, TLabeledEdit**, сторінки **Standard: TEdit, TButton, TLabel, TPanel**, сторінки **Dialogs: TColorDialog, TOpenPictureDialog, TSavePictureDialog** сторінки **System: TPaintBox**
- Розглянути призначення, основні властивості та методи роботи з новими компонентами.

**Теоретичні відомості:**

В процесі виконання завдань лабораторної роботи пропонується використати компоненти, теоретичні відомості про які подано було в попередніх лабораторних роботах, а також компоненти сторінки **Additional: TSpeedButton**, сторінки **Win32: TToolBar**, сторінки **Standard: TEdit, TButton, TLabel, TPanel**, сторінки **Dialogs: TOpenDialog, TSaveDialog**. Призначення, основні властивості та методи роботи з новими компонентами наведено нижче.

**Компонента TSpeedButton**

**Призначення:**

Створення кнопки із фіксацією та графікою. Використовуються переважно для створення кнопок швидкого доступу, що дублюють різні команди меню, та при створенні інструментальних панелей.

**Основні властивості:**

Основна властивість **Glyph** визначає зображення, що відображається на кнопці. Надпис на кнопці можна задати властивістю **Caption**, але здебільшого дана властивість залишається пустою. При використанні надпису слід звернути увагу на такі властивості, як **Layout**,

**Marging**, що визначають розташування зображення, та **Spacing**, що визначає відстань між зображенням та надписом.

Певну кількість кнопок можна об'єднувати в групу взаємопов'язаних кнопок, з яких в натисненому стані може бути тільки одна. Для цього використовують властивість **GroupIndex**, що визначає індекс групи (ненульове значення). Якщо властивість **GroupIndex=0**, то поведінка кнопки аналогічна до поведінки **Button** та **BitBtn** (при натисканні мишею кнопка переходить в натиснений стан, при відпусканні миші кнопка переходить до нормального стану).

Початковий стан кнопки може бути натисненим, якщо для властивості **Down** задавати значення **true** (рис.5.53).

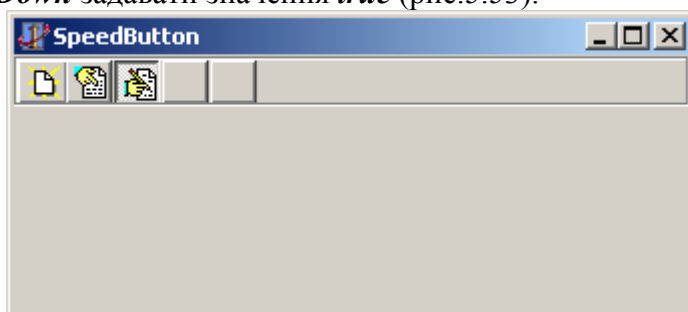


Рис.5.53. Приклад об'єднання кнопок **TSpeedButton** на панелі **TCoolBar**

Властивість **AllowAllUp** дозволяє віджати стан для всіх кнопок групи. Якщо **GroupIndex>0** та **AllowAllUp=true**, то якщо натиснути на кнопку, то вона залишиться в натисненому стані, якщо здійснюється повторне натиснення мишею, то кнопка повертається до початкового стану.

### Компоненти **TOpenDialog**, **TSaveDialog**

*Призначення:*

Компонента **TOpenDialog** призначена для створення діалогового вікна „Відкрити файл”, **TSaveDialog** – для створення діалогового вікна „Зберегти файл”.

*Основні властивості:*

Дані компоненти мають однакові властивості. Основна властивість **FileName**, в якій визначається вибраний користувачем файл. Значення даної властивості можна задати і під час проектування, тоді воно з'явиться в діалоговому вікні як значення по замовчуванню у вікні ім'я файлу.

Властивість **Filter** задає тип файлів, які будуть з'являтися у діалоговому вікні у випадаючому списку. Під час проектування дана властивість задається за допомогою редактора фільтрів (рис.5.54), який відкривається при натисканні кнопки з багатокрапкою біля імені даної властивості в **Інспекторі Об'єктів**.

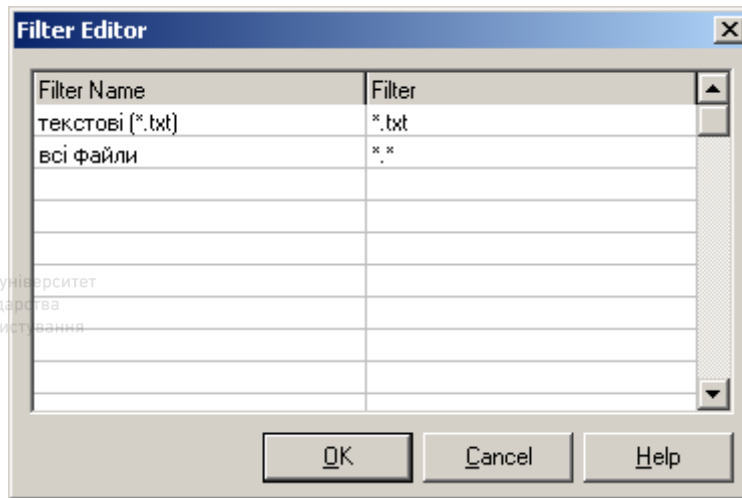


Рис.5.54. Редактор фільтрів

Таким чином, після виходу з редактора фільтрів задані шаблони з'являться у властивості **Filter** у вигляді стрічки: текстові (\*.txt)|\*.txt; всі файли|\*.\*. Існує також можливість задати дану властивість програмно під час виконання додатку.

В свою чергу властивість **FilterIndex** визначає номер фільтру, який буде відображатися по замовчуванню у момент відкриття діалогового вікна. Для властивості **InitialDir** слід задавати початковий каталог, який буде відкритим в момент роботи користувача з діалоговим вікном. Якщо дану властивість не визначити, то під час роботи з діалоговим вікном відкриється поточний каталог, або каталог, який був відкритим при останньому зверненні користувача до відповідного діалогового вікна в процесі виконання додатку.

Властивість **Title** дозволяє задати заголовок діалогового вікна. Тобто, існує можливість задавати заголовок діалогового вікна відмінним від стандартного, наприклад, заголовок може використовуватися для користувача підказкою очікуваних дій.

Властивість **DefaultExt** визначає значення розширення файлів по замовчуванню. Якщо властивість **DefaultExt** задана, то користувач може писати в діалоговому вікні ім'я без розширення, тобто буде прийняте розширення визначене по замовчуванню. Якщо дану властивість не вказати, то користувачу в діалоговому вікні слід зазначити повне ім'я файла разом із розширенням.

Властивість **Options** – це множина опцій, які визначають умови вибору файлів, дані опції можна задавати програмно або під час проектування:

Властивість **OptionsEx** містить одну опцію **ofExNoPlacesBar**, яка забороняє появу в діалозі смуги, що забезпечує доступ до папок **Робочий**

*стіл, Історія, Мої документи* і т.п., таким чином змінює вигляд діалогового вікна. Властивість **Files** типу **TStrings** визначає список вибраних користувачем файлів (якщо включена опція ofAllowMultiSelect).

Для компонент **TOpenDialog, TSaveDialog** передбачена можливість обробки ряду подій. Подія **OnCanClose** виникає при нормальному закритті діалогового вікна після вибору файлу, якщо користувач виходить із діалогового вікна не зробивши вибору, тобто нажимає кнопку „Отмена” або клавішу Esc, то подія **OnCanClose** не настає. Можна також реалізувати обробку подій **FolderChange** – зміна каталогу, **OnSelectionChange** – зміна ім'я файлу, **OnTypeChange** – зміна типу файла.

## Приклади розробки Delphi-додатків:

### Приклад 5.12

В середовищі програмування **Delphi** створити проект, який би дозволяв для файла *g*, компоненти якого є натуральні числа від 96 до 158, знаходити серед його компонент:

- кількість парних чисел;
- кількість подвоєних непарних чисел.

**Етап проектування** інтерфейсу проекту:

При розробці інтерфейсу (рис.5.55) використаємо візуальні компоненти: **TMemo, TButton, TLabel, TSpinEdit** та невізуальні: **TOpenDialog, TSaveDialog**.

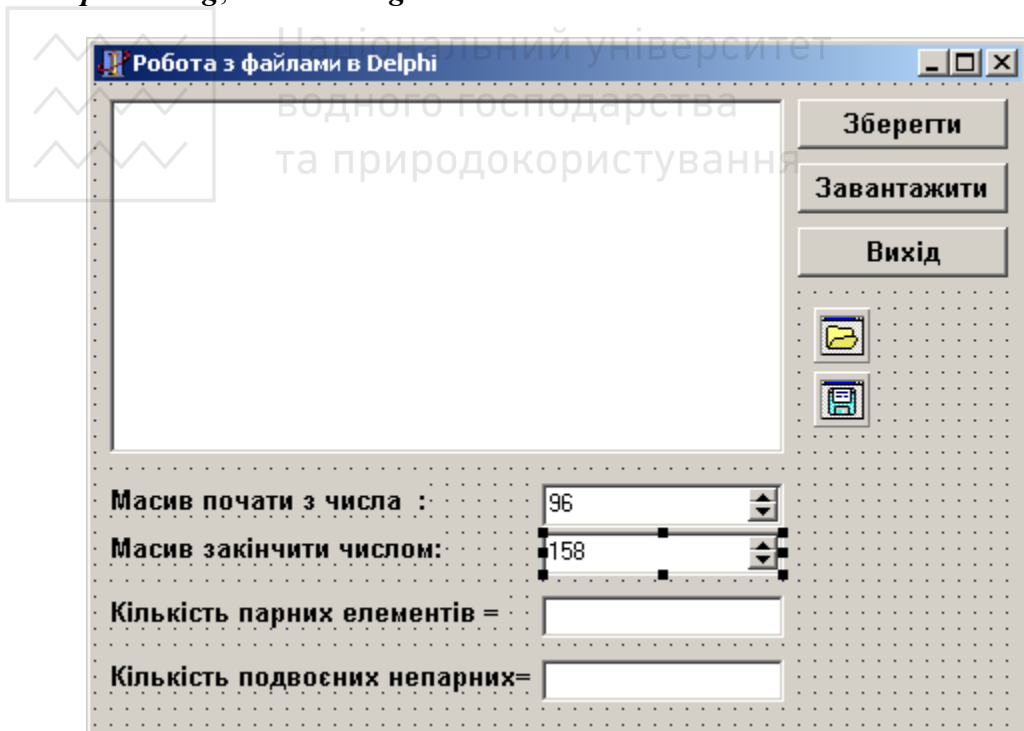


Рис.5.55. Форма проекту

Зазначимо основні властивості та зміни, які були проведені під час проектування:

```
Form1.Caption:= 'Робота з файлами в Delphi';  
Label1.Caption:= 'Масив почати з числа :'  
Label2.Caption:= 'Масив закінчити числом:'  
Label3.Caption:= 'Кількість парних елементів ='  
Label4.Caption:= 'Кількість подвоєних непарних='  
Memo1.ReadOnly:= True;  
SpinEdit1.Value:= 96  
SpinEdit2.Value := 158  
Button1.Caption := 'Зберегти'  
Button2.Caption := 'Завантажити'  
Button3.Caption := 'Вихід'  
Edit1.Text:= ' ';  
Edit2.Text:= ' '.
```

**Етап програмування** проекту:

Слід зазначити, що завдання виконано із використанням типізованих файлів

```
var  
f:file of integer;
```

В даному проекті існує можливість збереження сформованого файлу, компоненти якого є натуральними числами від 96 до 158. При натисканні на кнопку **'Зберегти'** реалізується дана можливість. Обробку події натискання мишею даної кнопки виконує наступна процедура:

```
procedure TForm1.Button2Click(Sender: TObject);  
var i,a1,an:integer;  
begin  
i:=0;  
if opendirialog1.Execute then  
begin  
assignfile(f,opendirialog1.filename);  
reset(f);  
while not eof(f) do  
begin  
i:=i+1;  
read(f,an);  
if i=1 then a1:=an;  
end;  
spinedit1.Value:=a1;
```

```

    spinedit2.Value:=an;
    end;
end;

```

Проміжок чисел, з якого формується файл визначається на формі за допомогою компонент *SpinEdit1* та *SpinEdit2*.

В проекті підрахунок кількості парних чисел та кількість подвоєних непарних чисел відбувається відразу при зміні значень проміжку, з якого формується файл. При зміні значень проміжку виникає подія *OnChange* для компонент *SpinEdit1* та *SpinEdit2*, обробка яких здійснюється наступною процедурою:

```

procedure TForm1.SpinEdit2Change(Sender: TObject);
var i,p,pn:integer; s:string;
begin
for i:=spinedit1.value to spinedit2.value do
begin
if i mod 2=0 // перевіряється умова на парність чисел
then p:=p+1; // підраховується кількість парних чисел
if (i mod 2=0) and (i mod 4<>0) // перевіряється умова , чи є
// число подвоєним непарним числом
then pn:=pn+1; // підраховується кількість подвоєних
// непарних чисел.
s:=s+IntToStr(i)+' ';
end;
memo1.Text:=s;
edit1.Text:=IntToStr(p); // виведення результатів до вікон
// редагування
edit2.text:=IntToStr(pn);
end;

```

Також дана процедура є обробкою події *OnCreate* для форми, оскільки таким чином забезпечує підрахунок визначених у завданні значень відразу після запуску проекту для файла, компонентами якого є натуральні числа від 96 до 158, оскільки проміжок визначено під час проектування.

Підрахувати кількість парних чисел та кількість подвоєних непарних чисел можна на основі вже існуючого файла, який можна завантажити, натиснувши на кнопку *'Завантажити'*. В результаті натиснення мишею кнопки виникає подія *OnClick*, обробка якої реалізується наступною процедурою:

```

procedure TForm1.Button1Click(Sender: TObject);
var i:integer;

```



```
begin  
if savedialog1.Execute then  
  begin  
    assignfile(f,savedialog1.filename);  
    rewrite(f);  
    for i:=spinedit1.value to spinedit2.value do  
      write(f,i);  
    closefile(f);  
  end;  
end;
```

В проєкті реалізовано також вихід з програми за допомогою кнопки **'Вихід'** та запезпечується процедурою:

```
procedure TForm1.Button3Click(Sender: TObject);  
begin  
close;  
end;
```

#### **Результат виконання** проєкту:

В результаті виконання програми у файлі, компоненти якого є натуральні числа від 96 до 158, підраховується кількість парних чисел та кількість подвоєних непарних чисел (рис.5.56).

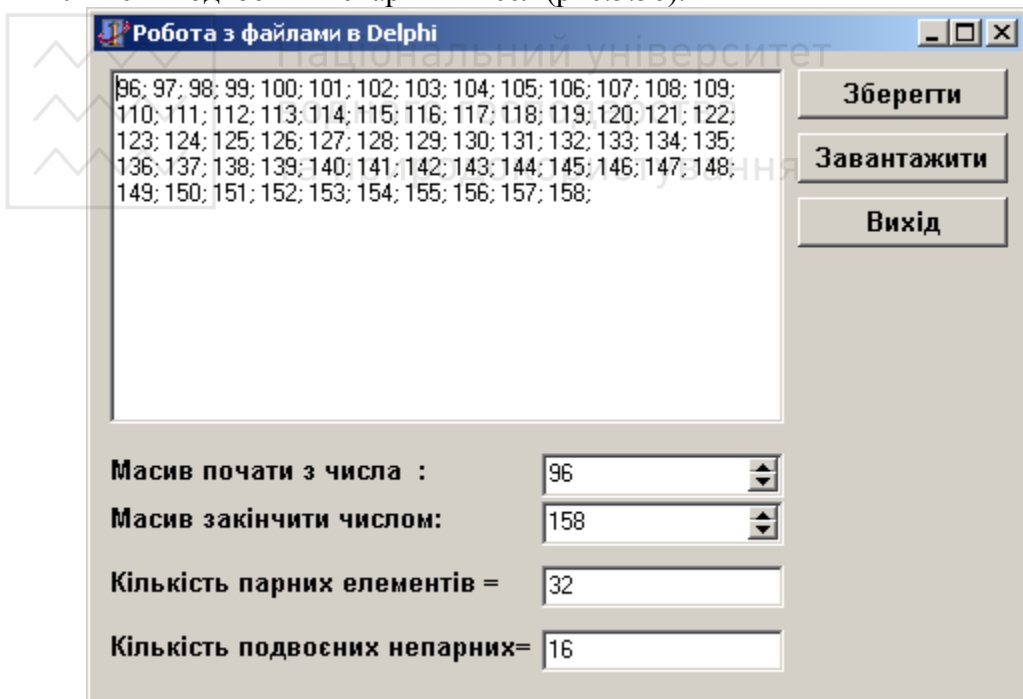


Рис.5.56. Результат виконання проєкту

### Приклад 5.13

В середовищі програмування **Delphi** створити проект, який би дозволяв підраховувати всі знаки пунктуації для заданого тексту.

**Етап проектування** інтерфейсу проекту:

При розробці інтерфейсу (рис.5.57) використаємо візуальні компоненти: *TMemo*, *TMainMenu*, *TButton*, *TLabel*, *TToolBar*, *TImageList* та невізуальні: *TOpenDialog*, *TSaveDialog*.

Зазначимо основні властивості компонент:

```
Form1.Caption:= 'Кількість знаків пунктуації';  
ToolButton1.Hint:= 'Створити новий файл';  
ToolButton1.Caption:= 'Create_new';  
ToolButton2.Hint := 'Відкрити файл';  
ToolButton2.Caption := 'Open_file';  
ToolButton3.Hint := 'Зберегти файл';  
ToolButton3.Caption := 'Save_file';  
ToolButton4.Hint := 'Провести обчислення';  
ToolButton4.Caption := 'Calculate';  
OpenDialog1.Filter := 'Текстові (*.txt)|*.txt';  
SaveDialog1.Filter := 'Текстові (*.txt)|*.txt';
```

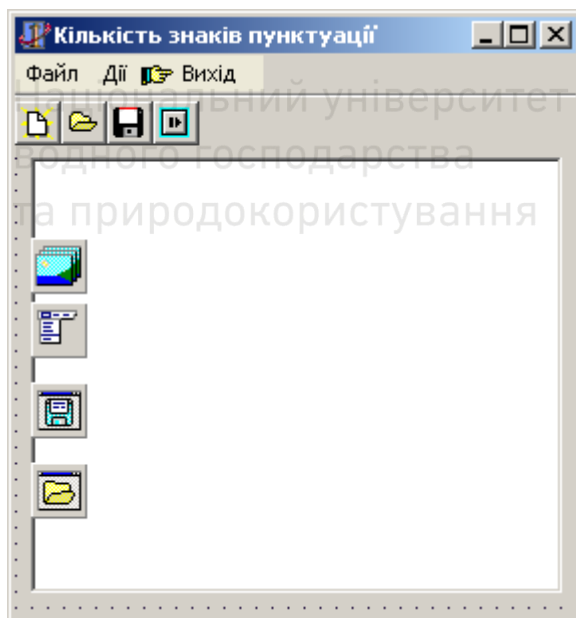


Рис.5.57. Вікно форми проекту прикладу 5.13

При розробці меню використано наступні розділи:

**Файл**

*Новий файл*



Відкрити файл

Зберегти файл

**Дії**

Кількість зн. пункт.

**Вихід**

Всі дії, що забезпечуються в меню, дублюються кнопками швидкого доступу на панелі **ToolBar**.



**Етап програмування** проекту:

водного господарства  
та природокористування

//збереження файлу

**procedure TForm1.SaveClick(Sender: TObject);**

**begin**

**if SaveDialog1.Execute then**

**Memo1.Lines.SaveToFile(SaveDialog1.FileName);**

**Memo1.Clear;**

**Memo1.ReadOnly:=true;**

**end;**

//відрутує файл

**procedure TForm1.OpenClick(Sender: TObject);**

**begin**

**if OpenFileDialog1.Execute then**

**Memo1.Lines.LoadFromFile(opendialog1.FileName);**

**end;**

//створення нового файлу (очищення поля Memo)

**procedure TForm1.NewClick(Sender: TObject);**

**begin**

**Memo1.Clear;**

**Memo1.ReadOnly:=False;**

**end;**

//закривається проект

**procedure TForm1.ExitClick(Sender: TObject);**

**begin**

**Close;**

**end;**

// процедура підрахунку кількості знаків пунктуації

**procedure TForm1.CalculateClick(Sender: TObject);**

**var**

**a:set of char;**

**k,i:integer; t:string;**

**begin**



Національний університет  
водного господарства  
та природокористування

```

a:=[',','!','?',';','-'];
k:=0;
t:=memo1.text;
for i:=1 to length(t) do
  if t[i] in a then inc(k);
  if k>0 then
showmessage('Кількість знаків пунктуації:'+inttostr(k))
else
showmessage('Знаків пунктуації в тексті не знайдено.');
```



Національний університет водного господарства та природокористування

Події *OnClick* для компонент *ToolButton1*, *ToolButton2*, *ToolButton3*, *ToolButton4* визначаються видно процедурами *TForm1.NewClick(Sender: TObject)*, *TForm1.OpenClick(Sender: TObject)*, *TForm1.SaveClick(Sender: TObject)*, *TForm1.CalculateClick(Sender: TObject)*.

**Результат виконання проекту:**

Результат виконання програми подано на рис.5.58 та рис.5.59.

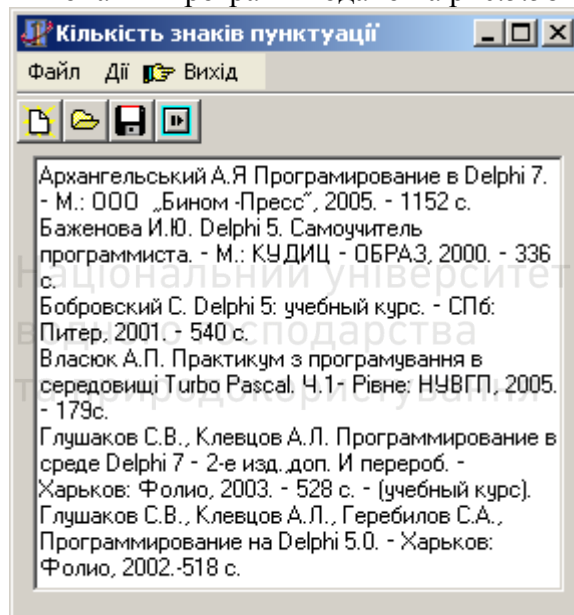


Рис.5.58. Завантаження файлу, для якого буде здійснюватися пошук

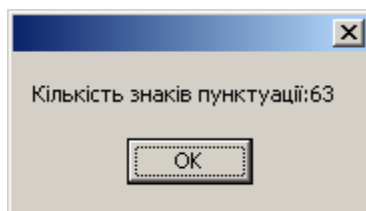


Рис.5.59. Результат виконання проекту

## Завдання для виконання лабораторної роботи:

### Завдання1.

В середовищі програмування **Delphi** розробити проекти для реалізації завдань, варіанти яких наведені нижче.

#### Вимоги до проекту:

- для створення, відкриття та збереження файлів використати компоненти **TOpenDialog**, **TSaveDialog**;
- дані, які записуються у файл або зчитуються з нього, відображати за допомогою візуальних компонент **TEdit**, **TStringGrid**, **TMemo** та ін.;
- створити головне меню (**TMainMenu**) та відповідну йому панель інструментів (**TToolBar** або **TPanel + TSpeedButton**);
- завдання сформувані так, ніби початковий файл вже існує на диску, але в програмі потрібно передбачити можливість створення початкового файлу відповідно до вимог та запитів користувача розробленою програмою;
- вхідні дані брати з файла;
- результати зберігати до файла.

#### Варіанти завдань:

1. Розробити проект, який би дозволив обробляти інформацію результатів модуля із курсу „Програмування ” для груп 21,22,23 (Прізвище, кількість балів). Забезпечити перегляд, редагування, збереження інформації.
2. Розробити проект сортування типізованого файлу (впорядкування його компонент за зростанням), що містить цілі числа.
3. Задано текст, який складається зі слів. Слова розділені пропусками. Знайти порядковий номер першого найдовшого слова.
4. Задано два типізованих файли, які містять цілі числа. Відсортувати файли за спаданням, забезпечити злиття файлів та впорядкувати отриманий файл за зростанням.
5. Задано два типізовані файли  $f$  та  $g$ , які містять цілі числа та мають однакову довжину. Створити третій файл  $h$  цілих чисел, що містить компоненти  $h_i = \max(f_i, g_i)$ .
6. Розробити спосіб шифрування та дешифрування інформації в текстових файлах.
7. Розробити проект для підрахунку кількості слів у тексті.
8. Розробити проект для підрахунку кількості знаків пунктуації.

9. Розробити проект для підрахунку в тексті кількості слів, які є паліндромами.
10. Розробити проект для визначення в тексті максимальну довжину послідовності символів, що не є буквами.
11. Розробити проект, який би визначав всі слова тексту, що складаються з тих самих букв, що останнє слово цього тексту.
12. Розробити проект, який забезпечував знаходження добутку елементів файлу.
13. Розробити проект, який би дозволяв визначати всі слова тексту, що складаються із тих самих букв, що і перше слово цього тексту.
14. Розробити проект, який би дозволяв читати з файла три послідовні речення, порядковий номер першого речення при цьому задається із клавіатури, і виводила їх на екран у зворотному напрямку.
15. Розробити проект, який би забезпечував читання з файла і виведення на екран тільки тих речень, які містять задане слово.
16. Розробити проект, який би забезпечував читання з файла і виведення на екран тільки тих речень, які містять двозначні числа.
17. Розробити проект, який би забезпечував читання з файла та виведення на екран тільки тих речень, які не містять ком.
18. Розробити проект, який би забезпечував знаходження середнього арифметичного елементів файлу.
19. Розробити проект, який би забезпечував читання тексту з файла та виведення на екран тільки тих речень, які мають визначену кількість слів.
20. Розробити проект, який би забезпечував читання з файла і виведення на екран тільки тих речень, які мають визначену кількість слів.
21. Розробити проект, який забезпечував читання з файла та виведення на екран тільки тих речень, які мають максимальну кількість знаків пунктуації.
22. Розробити проект, який забезпечував читання з файла та виведення на екран спочатку запитання, а потім речень зі знаком оклику.
23. Розробити проект, який би забезпечував читання з файла та виведення на екран спочатку однобуквені слова, а потім решту слів.
24. Розробити проект, який би забезпечував знаходження суми елементів файлу.
25. Розробити проект, який би дозволяв визначити всі слова тексту, що складаються із тих самих букв, що і друге слово цього тексту.

## **Завдання 2.**

*В середовищі програмування **Delphi** розробити віконний додаток для реалізації варіантів завдань лабораторної роботи №5. Забезпечити завантаження даних з файла та збереження результатів.*

## Контрольні запитання:

1. Дайте визначення файлу.
2. Пояснити технологію роботи з файлами.
3. Які існують типи файлів?
4. Що являють собою типізовані файли?
5. Що являють собою файли без типу?
6. Що являють собою текстові файли?
7. Які ви знаєте процедури та функції для роботи з файлами?



Національний університет  
водного господарства  
та природокористування

## Лабораторна робота №8

**Тема:** Створення баз даних засобами Delphi.

### **Мета роботи:**

- Навчитись використовувати можливості *Delphi* для створення баз даних.
- Навчитись працювати з компонентами сторінки *DataAccess: TDataSource*, сторінки *BDE: TTable*, сторінки *DataControls: TDBGrid, TDBNavigator, TDBText, TDBEdit*.
- Розглянути призначення, основні властивості та методи роботи з новими компонентами.

### **Теоретичні відомості:**

Для виконання завдань даної лабораторної роботи пропонується використати компоненти *TDataSource*, *TDBGrid*, *TDBNavigator*, *TDBText*, *TDBEdit*, *TTable*. Призначення, основні властивості та методи роботи з ними викладені нижче. При написанні програм пропонується також використати компоненти, які були розглянуті в теоретичних відомостях в попередніх лабораторних роботах.

### **Компонента TTable**

#### **Призначення:**

Це є набір даних, які в певний момент часу можуть бути зв'язані з однією таблицею бази даних. Забезпечує безпосередній зв'язок з фізичною таблицею бази даних, що розміщена на диску. Ця компонента розміщена на сторінці *BDE* палітри компонент.

#### **Основні властивості:**

Властивість *TableName* дозволяє задати ім'я файла, що містить таблицю, з якою потрібно працювати.

Властивість **Action** визначає активність компоненти **TTable**. Якщо дана властивість приймає значення **true**, то компонента **TTable** містить набір даних таблиці, що пов'язана з нею властивістю **TableName**. У протилежному випадку набір даних закритий і його зв'язок з базою даних розірвано.

Якщо властивість **ReadOnly** приймає значенням **true**, то користувачу забороняється змінювати записи таблиці, а дозволяється тільки їх переглядати.

Властивість **Exclusive** забезпечує для додатку монопольний доступ до таблиць (якщо властивість приймає значення **true**).

Властивість **IndexName** дозволяє задавати ім'я індексу, що визначений раніше при створенні таблиці. Властивість **IndexFieldName** дозволяє задавати ім'я полів, на основі яких був створений індекс. Якщо індекс створений для декількох полів, то вони перераховуються через крапку з комою.

Властивість **IndexFieldCount** дозволяє задавати кількість полів, на основі яких створюється індекс.

Властивість **IndexFields[Index: Integer]** дає можливість звертатися до полів поточного індексу за номером, що визначається параметром **Index** (нумерація починається з нуля).

Властивість **RecNo** визначає номер поточного запису.

Властивість **RecordCount** містить загальну кількість записів у таблиці.

Серед методів компоненти **TTable** виділимо наступні:

- **AddIndex** – визначає новий індекс для таблиці;
- **Add** – визначає новий індекс для таблиці (але не потребує переведення таблиці до монопольного режиму);
- **DeleteIndex** – видаляє індекс;
- **GetItemNames** – повертає імена індексів;
- **IndexOf** – виконує пошук індексу за іменем (повертає номер індексна);
- **Find** – використовується для пошуку індексу за іменем поля.

### Компонента TDataSource

*Призначення:*

Джерело даних **TDataSource** є проміжною ланкою між наборами даних і візуальними компонентами, за допомогою яких користувач може безпосередньо переглядати та редагувати ці набори даних. Ця компонента розміщена на сторінці **DataAccess** палітри компонентів.

*Основні властивості:*

Властивість **DataSet** використовується для визначення набору даних, з яким зв'язане джерело даних. В свою чергу, візуальні компоненти для зв'язку з джерелом даних використовують властивість **DataSource**.

Властивість **State** визначає поточний стан, в якому знаходиться набір даних.

Властивість **AutoEdit** визначає можливість редагування набору даних користувачем через візуальні компоненти з використанням певних властивих цим компонентам дій для переведення їх в режим модифікації. По замовчуванню властивість **AutoEdit** має значення **true**, тобто редагування записів дозволено. В протилежному випадку користувач не матиме можливості редагувати набір даних, пов'язаних з конкретним джерелом даних.

Властивість **Enabled** визначає, чи будуть візуальні компоненти, які пов'язані з конкретним джерелом даних, відображати дані.

### Компонента **TDBGrid**

*Призначення:* Компонента **TDBGrid** використовується для відображення вмісту набору даних у вигляді таблиці, де стрічки відповідають записам, а стовпчики – полям. Компонента **TDBGrid** надає можливість користувачу переглядати та змінювати дані.

*Основні властивості:*

Найголовнішою для компоненти **TDBGrid** є властивість **Columns**, що містить список стовпців компоненти. Для того, щоб викликати *Редактор стовпців*, необхідно натиснути правою кнопкою миші на компоненті **TDBGrid**, а потім в контекстному меню вибрати пункт **Columns Editor** (*Редактор стовпців*). Крім того, *Редактор стовпців* можна викликати при подвійному натисненні на компоненті **TDBGrid**, чи використати при цьому *Інспектор Об'єктів*. Вікно *Редактора стовпців* показано на рис.5.61

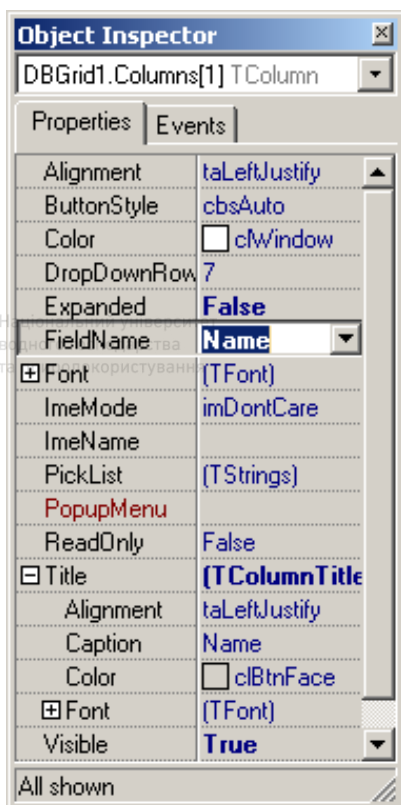


Рис.5.60. Інспектор Об'єктів при визначенні стовпчика з ім'ям Name

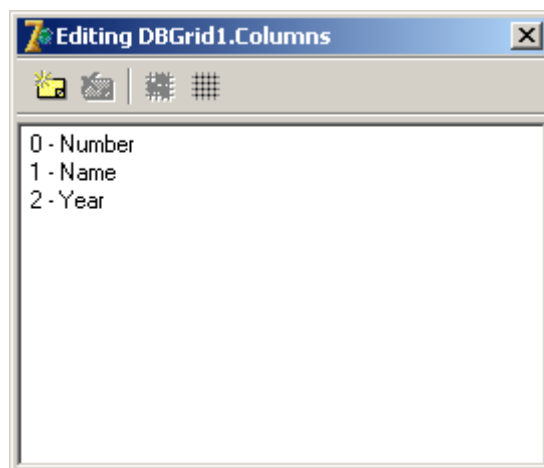


Рис.5.61. Редактор стовпців

Спочатку список стовпців пустий. Редактор дозволяє додавати новий або видалити стовпець, а також змінити їх порядок. Підміні порядку стовпців автоматично змінюється порядок зв'язаних з ним полів набору даних.

Властивість **Alignment** визначає спосіб вирівнювання значень в комірках стовпця.

Властивість **PickList** містить список для вибору значень, що вводяться в поле.

Властивість **Title** є об'єктом заголовка таблиці і в свою чергу має такі властивості: **Caption**, **Alignment**, **Color** і **Font**, які визначають відповідно назву, спосіб вирівнювання, колір і шрифт заголовка стовпця.

### Компонента TDBNavigator

**Призначення:** Компонента **TDBNavigator** – це є набір кнопок, який забезпечує інтерфейс керування набором даних, та дозволяє користувачу здійснювати переміщення по набору даних, додавання, редагування та видалення записів.

**Основні властивості:**



Властивість **VisibleButtons** визначає структуру видимих кнопок та може приймати комбінацію наступних значень:

- **nbFirst** – перехід до першого запису;
- **nbPrior** – перехід до попереднього запису;
- **nbNext** – перехід до наступного запису;
- **nbLast** – перехід до останнього запису;
- **nbInsert** – вставка нового запису;
- **nbDelete** – видалення поточного запису;
- **nbEdit** – редагування поточного запису;
- **nbPost** – підтвердження результату зміни запису;
- **nbCancel** – відміна зміни поточного запису;
- **nbRefresh** – оновлення інформації в наборі даних.

По замовчуванню всі кнопки навігатора є видимими.

Властивість **ConfirmDelete** при значенні **true** забезпечує вивід на екран діалогового вікна, що підтверджує видалення запису.

Якщо властивість **Flat** приймає значення **true**, то форма кнопок є плоскою, у протилежному випадку форма кнопок є об'ємною.

Властивість **Hints** визначає підказку для кожної окремої кнопки. При чому, для відображення підказок властивість **ShowHint** має приймати значення **true**.

### Компоненти **DBText** та **DBEdit**

**Призначення:** Компонента **DBText** дозволяє відображати значення визначеного поля поточного запису таблиці БД у вигляді надпису. Компонента **DBEdit** виводить значення заданого поля поточного запису в однорядковому редакторі.

**Основні властивості:**


Властивість **DataSource** – визначає джерело даних.

Властивість **DataField** – визначає поле таблиці, для відображення чи редагування якого використовується компонента.

## **Завдання для виконання лабораторної роботи:**

### **Варіанти завдань:**

1. Створити інформаційно-пошукову систему для станції технічного обслуговування.
2. Створити інформаційно-пошукову систему для автоматизації роботи приймальної комісії.

- 
- Національний університет водного господарства та природокористування
3. Створити інформаційно-пошукову систему контролю виконання доручень для деякої організації.
  4. Створити інформаційно-пошукову систему готельного комплексу.
  5. Створити інформаційно-пошукову систему агенства продажу нерухомості.
  6. Створити інформаційно-пошукову систему обліку витрат пального.
  7. Створити інформаційно-пошукову систему оплати праці підприємства.
  8. Створити інформаційно-пошукову систему рекламного відділу газети.
  9. Створити інформаційно-пошукову систему рекламної агенції.
  10. Створити інформаційно-пошукову систему телефонної станції.
  11. Створити інформаційно-пошукову систему туристичної фірми.
  12. Створити інформаційно-пошукову систему рекламної агенції.
  13. Створити інформаційно-пошукову систему фірми з продажу автомобілів.
  14. Створити інформаційно-пошукову систему бібліотеки інституту.
  15. Створити інформаційно-пошукову систему обліку товарів у продуктовому магазині.
  16. Створити інформаційно-пошукову систему обліку товарів у промисловому магазині.
  17. Створити інформаційно-пошукову систему для аналізу плати за користування дитячими закладами.
  18. Створити інформаційно-пошукову систему гуртожитку.
  19. Створити інформаційно-пошукову систему роботи автостоянки.
  20. Створити інформаційно-пошукову систему для обліку товарно-матеріальних цінностей на складі.
  21. Створити інформаційно-пошукову систему для медичної частини інституту.
  22. Створити інформаційно-пошукову систему для профспілкової організації інституту.
  23. Створити інформаційно-пошукову систему для магазину з продажу комп'ютерних комплектуючих.
  24. Створити інформаційно-пошукову систему, за допомогою якої можна було б здійснювати контроль знань студентів.
  25. Створити інформаційно-пошукову систему реєстратури поліклініки.

**Вимоги до завдань:**

- 1) забезпечити правильність введення даних;
- 2) забезпечити можливість додавання інформації у зручному форматі;
- 3) забезпечити редагування інформації;
- 4) забезпечити пошук потрібної інформації.

### Контрольні запитання:

1. Дайте визначення бази даних.
2. Як здійснюється доступ до баз даних з додатку *Delphi*?
3. Що таке індекс?
4. Які бувають індекси?
5. Що таке ключ?
6. Як створюються таблиці за допомогою *Database Desktop*?
7. Які компоненти використовуються для візуалізації та управління даними в додатках *Delphi*?
8. Які Ви знаєте методи створення баз даних?



Національний університет  
водного господарства  
та природокористування



Національний університет  
водного господарства  
та природокористування

## Контрольні завдання по курсу

### I. *Понятійний рівень:*

1. Проект
2. Форма
3. Модуль
4. Палітра компонент
5. Компоненти палітри
6. Об'єкти проекту
7. Метод
8. Подія в Delphi
9. Object Pascal
10. Клас
11. Середовище програмування Delphi
12. База даних
13. Індекс
14. Ключ
15. Записи
16. Поле

### II. *Змістовний рівень:*

1. Основні характеристики програмного продукту Delphi.
2. Характеристика ICP Delphi:
  - 1) Головне вікно середовища.
  - 2) Головне меню.
  - 3) Призначення вкладок палітри компонент.
  - 4) Інспектор об'єктів.
  - 5) Візуальне проектування робочої форми.
  - 6) Редактор коду програми.
3. Характеристика проектів Delphi:
  - 1) Основні файли проекту.
  - 2) Структура проекту Delphi.
  - 3) Структура файла модуля.
  - 4) Структура файла проекту.
  - 5) Розширення основних файлів: проекту, модуля і форми.
4. Методи формування та виконання проекту в Delphi.
5. Особливості програмування в Object Pascal:
  - 1) Типи даних, що використовуються в Object Pascal.
  - 2) Організація введення та виведення даних.

- 3) Основні математичні процедури та функції.
- 4) Масиви в Delphi, їх виведення та обробка.
- 5) Методика роботи із таблицями в Delphi, основні компоненти.

6. Графічні можливості Delphi:

- 1) Компоненти Delphi, що використовуються при роботі з графікою, основні властивості.
- 2) Процедури (методи) для рисування графічних примітивів.
- 3) Характеристика методів малювання.
- 4) Побудова найпростіших геометричних фігур.

7. Робота з файлами в Delphi:

- 1) Поняття файлу, типи файлів.
- 2) Загальний принцип роботи з файлами.
- 3) Стандартний підхід при роботі з файлами.
- 4) Процедури та функції для роботи із текстовими файлами.
- 5) Файли типізовані та файли без типу, процедури та функції.
- 6) Процедури та функції для роботи з файлами, які використовують дескриптори.
- 7) Об'єктний підхід при роботі з файлами.
- 8) Характеристика основних компонент Delphi, що використовуються при роботі з файлами та принцип роботи з ними.

8. Створення баз даних в Delphi:

- 1) Як здійснюється доступ до баз даних з додатку Delphi?
- 2) Як створюються таблиці за допомогою Database Desktop?
- 3) Дайте характеристику компонентам, що використовуються для візуалізації та управління даними в додатках Delphi?



## Тематика курсових робіт

### Вибір теми курсової роботи

Тема кожної курсової роботи визначається керівником роботи з урахуванням побажань студента, його інтересів та теми майбутньої дипломної роботи.

Всі теми відповідають проблемно-орієнтованим задачам, а тому вимагають від студента творчого підходу до їх розв'язання.

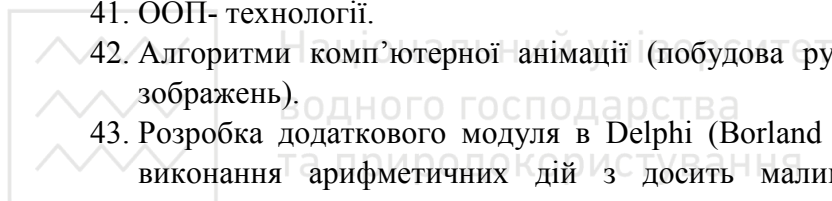


національний університет  
водного господарства  
та природокористування

### Приклади тем курсових робіт

1. База даних “Деканат”.
2. База даних “Кафедра”.
3. Електронний довідник.
4. Інформаційно-довідкова система “Медпункт”.
5. Інформаційно-довідкова система “Аналіз ринку попиту”.
6. Інформаційно-довідкова система: “Рейтингова таблиця”.
7. Інтелектуальна система “Запитання-відповідь” (із заданої предметній області).
8. Аналіз дорожньо-транспортних пригод на дорогах області.
9. Інформаційно-довідкова система “Курси валют”.
10. Інформаційно-довідкова система “Облік роботи приватних таксобусів”.
11. Інформаційно-довідкова система “Облік товарів у промисловому магазині”.
12. Інформаційно-довідкова система “Облік товарно-матеріальних цінностей на складі”.
13. База даних „Екологія” ( про екологічні санкції з підприємств-забруднювачів навколишнього середовища).
14. Програма перегляду графічних РСХ-файлів.
15. Програма запису і перегляду графічних BMP-файлів.
16. Моделювання маятника Капіці.
17. Розв'язання диференціальних рівнянь методом Монте-Карло.
18. Програма для відтворення звукових WAV-файлів.
19. Трьохвимірний лабіринт.
20. Кодування і декодування файлів для пересилки по електронній пошті.
21. Дослідження геометричних властивостей фракталів.
22. Управління космічним апаратом на орбіті і при польоті на Місяць.
23. Перетворення вихідного тексту програми на мові Pascal в RTF-файл.

24. Чисельне розв'язання нелінійних рівнянь (з графічною інтерпретацією уточнення коренів).
25. Чисельне розв'язання СЛАР.
26. Інтерполяція функцій.
27. Математична обробка результатів експерименту.
28. Наближене обчислення визначених інтегралів (звичайних, подвійних, потрійних).
29. Комп'ютерне моделювання розв'язку ЗЛП графічним методом.
30. Моделювання деяких фізичних процесів за допомогою ЕОМ.
31. Побудова гідродинамічних та ізотермічних полів за допомогою ЕОМ.
32. QR-алгоритми розв'язання СЛАР (шляхом перетворення матриці до 3-х діагональної).
33. Математичні моделі переслідування.
34. Порівняльний аналіз алгоритмів пошуку в масивах.
35. Порівняльний аналіз алгоритмів пошуку в текстах.
36. Порівняльний аналіз алгоритмів сортування масивів.
37. Порівняльний аналіз сортування файлів та послідовностей.
38. Алгоритми з використанням динамічних структур даних.
39. Алгоритми шифрування та дешифрування текстів.
40. Алгоритми архівування файлів.
41. ООП- технології.
42. Алгоритми комп'ютерної анімації (побудова рухомих графічних зображень).
43. Розробка додаткового модуля в Delphi (Borland C++ Builder) для виконання арифметичних дій з досить малими та великими числами.
44. Стрічковий компілятор.
45. Розробка бази даних графічних зображень.
46. Комп'ютерна ідентифікація кривих та поверхонь.
47. Драйвери-кирилізатори клавіатур.
48. Дис-асемблерування ехе-файлів, написаних на C++ та Pascal.
49. Віруси - антивіруси.
50. Програми - захисники від копіювання.
51. Робота з відеопам'яттю ПК.
52. Виконання музичних творів на ПК (програмування музичних творів на внутрішній динамік).
53. Програмування розпаралелення процесів на ПК.
54. Розпізнавання образів: автоматичне розпізнавання тексту.
55. Алгоритми перенумерації вузлів трикутної сітки для мінімізації



- ширини стрічки матриці жорсткості.
56. Автоматизація побудови кросвордів.
  57. Прямі методи розв'язання розрідження систем лінійних алгебричних рівнянь.
  58. Ітераційні методи розв'язання розріджених систем лінійних алгебричних рівнянь.
  59. Алгоритми розбиття плоскої кривої на скінчені елементи.
  60. Навчально контролююча програма "C++Builder".
  61. Моделювання руху автотранспорту та пішоходів на перехресті.
  62. Навчально-контролююча програма "Основи об'єктно-орієнтованого програмування" ( Object Pascal та C++).
  63. Навчально-контролююча програма "Visual C++".
  64. Алгоритми розбиття плоского многокутника на скінчені трикутні елементи.
  65. Програмування структур даних з використанням орієнтованих графів та їх застосування.
  66. Програмування структур даних з використанням неорієнтованих графів та їх застосування.
  67. Програмування алгоритмів для зовнішньої пам'яті.
  68. Програмування алгоритмів управління оперативною пам'яттю.
  69. Порівняльний аналіз виконання різних програм різними процесорами.
  70. Порівняльний аналіз виконання різних програм в різних середовищах програмування (GW Basic, Turbo Basic, Turbo , Borland Pascal, Borland C++, Delphi, Borland C++ Builder).

### ***Вимоги до виконання курсової роботи***

### **Оформлення курсової роботи**

***Курсова робота повинна містити наступне:***

1. Титульний лист (див. приклад курсової роботи).
2. Зміст (із вказанням сторінок).
3. Вступ (короткий виклад важливості розв'язуваного класу задач та доцільність її реалізації на ПК, опис предметної області).
4. Змістовна постановка задачі (приведення загального вигляду вхідних документів та вихідних форм, аналіз поставленої задачі, опис інтерфейсу).
5. Алгоритм розв'язання задачі (словесний або у вигляді блок-схеми).
6. Текст програми (написаний від руки або надрукований).
7. Зображення та описання всіх форм, які є в проекті.



8. Тестовий приклад (з наведеною повною вхідною інформацією, яка дозволяє перевірити всі режими роботи програми, всіма вихідними формами) з аналізом отриманих результатів.
9. Інструкція користувачу, в якій міститься пояснення щодо правил користування програмою.
10. Висновки.
11. Список використаної літератури.

Курсова робота оформляється на листах формату А4, що зверху нумеруються і зшиваються з лівої сторони. Ліве поле повинне бути від 25 до 30 мм, праве - не менше 10 мм; верхнє – 20 мм; нижнє – 20 мм.

Текст повинен відповідати змісту. Всі пункти змісту викладаються чітко і коротко.

### Вимоги до розробки програми

Програма для реалізації курсової роботи розробляється на мовах програмування *Object Pascal* з використанням систем розробки програм *Delphi*.

При написанні програми потрібно дотримуватися наступних вимог:

- Початкові дані та, можливо, результати обчислень зберігати у типованих файлах.
- На носії інформації (Гнучкому Магнітному Диску) обов'язково повинен знаходитися текст програмного проекту та текст звіту.
- Програма повинна містити зручний для користування інтерфейс.
- Тестування програми провести на прикладі, що дозволяє перевірити всі режими роботи програми.

### Захист курсової роботи

Оформлена курсова робота подається студентом разом з текстом програми на дискеті до захисту. Захист роботи проходить у формі співбесіди. Під час захисту потрібно коротко розповісти про зміст задачі, основні етапи її розв'язання, аргументувати вибір того чи іншого методу розв'язання задачі, пояснити як склалися алгоритми і відповідна їх програмна реалізація. На поставлені запитання по виконаній роботі студент повинен дати вичерпні відповіді. Мета опитування полягає у встановленні глибини засвоєння студентом відповідного матеріалу з курсової роботи, ступеня самостійності її виконання. При оцінюванні роботи враховується якість її виконання і оформлення, своєчасність виконання етапів роботи, результати співбесіди.

## Приклад типової курсової роботи

### Основні етапи виконання курсової роботи

Виконання курсової роботи складається з наступних етапів:

1. *Змістовна постановка задачі.* Після отримання завдання для виконання курсової роботи студент повинен вивчити предметну область, що пов'язана з темою курсової роботи та її застосуванням. Для цього він самостійно підбирає і опрацьовує необхідну літературу по тематиці роботи, консультуючись при цьому з науковим керівником.
2. *Аналіз поставленої задачі.* Перш, ніж приступити до програмної реалізації розроблених алгоритмів, потрібно ретельно проаналізувати задачу. Якщо практична реалізація алгоритму розв'язання задачі досить складна, він може бути спрощений за погодженням з керівником.
3. *Розробка і опис алгоритму розв'язання задачі.*
4. *Розробка і написання програми розв'язання задачі.* При написанні програми потрібно користуватися сучасними засобами та розробками в області програмування.



Національний  
водного господарства  
та природокористування



Національний університет  
водного господарства та природокористування

Міністерство освіти і науки України  
Національний університет водного господарства та природокористування  
Кафедра прикладної математики

Курсова робота  
на тему:

“Алгоритми шифрування та дешифрування текстової інформації”

Виконав: студент групи ПМ-23

ФПМіКІС

Шепетько В.В.

Науковий керівник:  
д.т.н., професор  
Власюк А.П.

Рівне - 2007

## Зміст

### Вступ

1. Шифрування - метод захисту інформації
  2. Класифікація алгоритмів шифрування
    - 2.1. Класифікація алгоритмів
    - 2.1. Симетричні потокові алгоритми шифрування
    - 2.3. Симетричні блокові алгоритми шифрування
    - 2.4. Асиметричні алгоритми шифрування
  3. Реалізація алгоритмів шифрування
  4. Текст програми
  5. Інструкція користувача
  6. Тестовий приклад
- Висновки.  
Література



Національне агентство з питань  
водного господарства та пр

### Вступ

Минуле і теперішнє століття є століттям інформатики та інформатизації. Нові технології дають можливість передавати і зберігати все більші обсяги інформації. Це благо має і зворотний бік: інформація стає все більш уразливою.

Тому великого значення набуває проблема захисту інформації від несанкціонованого доступу (НСД) при передачі і збереженні. Сутність цієї проблеми – постійна боротьба фахівців із захисту інформації зі своїми "опонентами".

Проблема захисту інформації шляхом її шифрування, що виключає її зчитування сторонніми особами, хвилювала людський розум з давніх часів. Із широким поширенням писемності криптографія стала формуватися як самостійна наука. Перші криптосистеми зустрічаються вже на початку нашої ери. Так, Цезар у своїй переписці використовував систематичний шифр, що був названий його ім'ям. Криптографічні системи бурхливо розвивалися в роки першої та другої світових воєн. Починаючи з післявоєнного часу і по нинішній день поява обчислювальних засобів прискорила розробку та удосконалювання криптографічних методів.

Чому проблема використання криптографічних методів в інформаційних системах (ІС) стала на даний момент особливо актуальною?

З одного боку, розширилося використання комп'ютерних мереж, зокрема глобальної мережі Інтернет, по якій передаються великі обсяги інформації державного, військового, комерційного і приватного характеру, що не допускає можливість доступу до неї сторонніх осіб. З іншого боку, поява нових потужних комп'ютерів, технологій мережевих і нейронних обчислень уможливила дискредитацію криптографічних систем, що до недавнього часу вважалася не можливою.

## 1. Шифрування - метод захисту інформації



Національний університет  
водного господарства  
та природокористування

Проблемою захисту інформації шляхом її перетворення займається криптологія (kryptos – таємний, logos – наука). Криптологія розділяється на два напрямки: криптографію і криптоаналіз. Криптографія займається пошуком і дослідженням математичних методів перетворення інформації. Сфера інтересів криптоаналізу – дослідження можливості розшифрування інформації без знання ключів.

В курсовій роботі основна увага приділена криптографічним методам. Сучасна криптографія містить у собі чотири великих розділи:

- симетричні криптосистеми;
- криптосистеми з відкритим ключем;
- системи електронного підпису;
- керування ключами.

Основні напрямки використання криптографічних методів: передача конфіденційної інформації за допомогою каналів зв'язку (наприклад, електронної пошти), встановлення істинності переданих повідомлень, збереження інформації на носіях у зашифрованому вигляді. Криптографія дає можливість перетворити інформацію таким чином, що її прочитання (відновлення) можливе тільки при знанні ключа.

Як інформацію, що підлягає шифруванню і дешифруванню, будемо розглядати тексти, побудовані на деякому алфавіті. Під цими термінами розуміється наступне:

**Алфавіт** – послідовність знаків, які використовуються для кодування інформації.

**Текст** – впорядкований набір з елементів алфавіту.

Як приклади алфавітів, використовуваних у сучасних ІС можна привести наступні:

- алфавіт Z33 – 32 букви російського алфавіту та пробіл;
- алфавіт Z256 – символи, що входять у стандартні коди ASCII і KOI-8;
- бінарний алфавіт  $Z2 = \{0, 1\}$ ;

➤ вісімковий алфавіт або шістнадцятковий алфавіт.

**Шифрування** (encryption) – процес перетворення: вихідний текст, що також називають відкритим текстом (plaintext), замінюється шифрованим текстом (ciphertext).

**Дешифрування** – зворотний шифруванню процес. На основі ключа зашифрований текст перетворюється у вихідний.

**Ключ** – інформація, необхідна для безперешкодного шифрування і дешифрування текстів.

Криптографічна система являє собою сімейство  $T$  перетворень відкритого тексту. Члени цього сімейства індексуються, чи позначаються символом  $k$ ; параметр  $k \in K$  ключем. Простір ключів  $K$  -- це набір можливих значень ключа. Звичайно, ключ являє собою послідовний ряд букв алфавіту. **Криптостійкістю** називається характеристика шифру, визначальна його стійкість до дешифрування без знання ключа (тобто криптоаналізу). Мається на увазі кілька показників криптостійкості, серед яких:

- кількість усіх можливих ключів;
- середній час, необхідний для криптоаналізу.

Для сучасних криптографічних систем захисту інформації сформульовані наступні загальноприйняті вимоги:

- зашифроване повідомлення повинне піддаватися читанню тільки при наявності ключа;
- число операцій, необхідних для визначення використаного ключа шифрування по фрагменті шифрованого повідомлення і відповідного йому відкритого тексту, повинне бути не менше загального числа можливих ключів;
- високий рівень захисту даних проти дешифрування і можливої модифікації;
- захищеність інформації повинна ґрунтуватися тільки на знанні ключа і не залежати від того, відомий алгоритм чи ні (правило Кірхгоффа);
- мала зміна вихідного тексту чи ключа повинна приводити до значної зміни шифрованого тексту (ефект "обвалу");
- область значень ключа повинна виключати можливість дешифрування даних шляхом перебору значень ключа;
- економічність реалізації алгоритму при достатній швидкодії;
- мати строгую нижню оцінку і виходити за межі можливостей

- сучасних комп'ютерів (з урахуванням можливості використання мережевих обчислень);
- знання алгоритму шифрування не повинне впливати на надійність захисту;
  - структурні елементи алгоритму шифрування повинні бути незмінними;
  - додаткові біти, що вводяться в повідомлення в процесі шифрування, повинні бути надійно сховані в шифрованому тексті;
  - довжина шифрованого тексту повинна дорівнювати довжині вихідного тексту;
  - не повинно бути протиставлених і легко встановлюваних залежностей між ключами, послідовно використовуваними в процесі шифрування;
  - будь-який ключ з безлічі можливих повинний забезпечувати надійний захист інформації;
  - алгоритм повинний допускати як програмну, так і апаратну реалізацію. При цьому зміна довжини ключа не повинна вести до якісного погіршення алгоритму шифрування.



## 2. Класифікація алгоритмів шифрування

### 2.1 Класифікація алгоритмів

1. Симетричні (із секретним, єдиним ключем, одноключові, single-key).

1.1. Поточкові (шифрування потоку даних):

- з одноразовим чи нескінченним ключем (infinite-key cipher);
- зі скінченним ключем (система Вернама - Vernam);
- на основі генератора псевдовипадкових чисел (ПВЧ).

1.2. Блокові (шифрування даних поблочно):

1.2.1. Шифри перестановки (permutation, P-блоки);

1.2.2. Шифри заміни (підстановки, substitution, S-блоки):

- моноалфавітні (код Цезаря);
- поліалфавітні (шифр Видженера, циліндр Джефферсона, диск Уетстоуна);

1.2.3. Складені шифри:

- Lucifer (фірма IBM, США);
- DES (Data Encryption Standard, США);

- FEAL-1 (Fast Enciphering Algorithm, Японія);
- IDEA/IPES (International Data Encryption Algorithm, Improved Proposed Encryption Standard, фірма Ascom-Tech AG, Швейцарія);
- В-Crypt (фірма British Telecom, Великобританія).

## 2. Асиметричні (з відкритим ключем, public-key):



- Діффі-Хеллман DH (Diffie, Hellman);
- Райвест-Шамір-Адлеман RSA (Rivest, Shamir, Adleman);

- Эль-Гамаль (ElGamal).

Крім того, є поділ алгоритмів шифрування на власне шифри (ciphers) і коди (codes). Шифри працюють з окремими бітами, буквами, символами. Коди оперують лінгвістичними елементами (склади, слова, фрази).

## 2.2. Симетричні поточкові алгоритми шифрування

Симетричні алгоритми шифрування базуються на тому, що відправник і одержувач інформації використовують той самий ключ. Цей ключ повинен зберігатися в таємниці і передаватися способом, що виключає його перехоплення.

У поточкових шифрах, тобто при шифруванні потоку даних, кожен біт вихідної інформації шифрується незалежно від інших за допомогою гамування.

Гамування - накладення на відкриті дані гами шифру (випадкової чи псевдовипадкової послідовності одиниць і нулів) за визначеним правилом. Звичайно використовується "виключаюче ЧИ", назване також додаванням по модулю 2 і реалізоване в програмах асемблера командою XOR. Для розшифрування та ж гама накладається на зашифровані дані.

При однократному використанні випадкової гами однакового розміру з зашифровуваними даними злам коду неможливий (так звані криптосистеми з одноразовим чи нескінченним ключем). У даному випадку "нескінченний" означає, що гама не повторюється.

У деяких поточкових шифрах ключ коротший повідомлення. Так, у системі Вернама для телеграфу використовується паперове кільце, що містить гаму. Звичайно, стійкість такого шифру не ідеальна.

Найчастіше використовують гаму, одержувану за допомогою генератора псевдовипадкових чисел (ПВЧ). У цьому випадку ключ – це число (початкове значення, вектор ініціалізації) для запуску генератора

ПВЧ. Кожен генератор ПВЧ має період, після якого послідовність повторюється. Очевидно, що період псевдовипадкової гами повинний перевищувати довжину інформації, що зашифровується. Генератор ПВЧ вважається коректним, якщо спостереження фрагментів його виходу не дозволяє відновити всю послідовність при відомому алгоритмі, але невідомому початковому значенні.

При використанні генератора ПВЧ можливі кілька варіантів:

1. Побітове шифрування потоку даних. Цифровий ключ використовується як початкове значення генератора ПВЧ, а вихідний потік бітів сумується по модулю 2 з вихідною інформацією. У таких системах відсутня властивість поширення помилок.
2. Побітове шифрування потоку даних зі зворотним зв'язком (ЗЗ) по шифртексту. Така система аналогічна попередній, за винятком того, що шифртекст повертається як параметр у генератор ПВЧ. Характерна властивість поширення помилок. Область поширення помилки залежить від структури генератора ПВЧ.
3. Побітове шифрування потоку даних із ЗЗ по вихідному тексті. Базою генератора ПВЧ є вихідна інформація. Характерна властивість необмеженого поширення помилки.
4. Побітове шифрування потоку даних із ЗЗ по шифртексті і по вихідному тексті.

### 2.3. Симетричні, блокові алгоритми шифрування

При блоковому шифруванні інформація розбивається на блоки фіксованих розмірів і шифрується поблочно. Блокові шифри бувають двох основних видів:

- шифри перестановки (transposition, permutation, P-блоки);
- шифри заміни (підстановки, substitution, S-блоки).

Шифри перестановок переставляють елементи відкритих даних (біти, букви, символи) у деякому новому порядку. Розрізняють шифри горизонтальної, вертикальної, подвійної перестановки, ґрати, лабіринти, лозунгові та ін.

Шифри заміни заміняють елементи відкритих даних на інші елементи за визначеним правилом. Розрізняють шифри простої, складної, парної заміни, буквено-складове шифрування і шифри колонної заміни. Шифри заміни поділяються на дві групи:

- моноалфавітні (код Цезаря) ;
- поліалфавітні (шифр Відженера, циліндр Джефферсона, диск Уетстоуна).



У моноалфавітних шифрах заміни буква вихідного тексту замінюється на іншу, заздалегідь визначену букву. Наприклад, у кодї Цезаря буква замінюється на букву, що віддалена від неї в латинському алфавіті на деяке число позицій.

У поліалфавітних підстановках для заміни деякого символу вихідного повідомлення в кожному випадку його появи, послїдовно використовуються різні символи з деякого набору.

У сучасних криптографічних системах, як правило, використовують обидва способи шифрування (заміни і перестановки). Такий шифратор називають складеним.

8

## 2.4. Асиметричні алгоритми шифрування

В асиметричних алгоритмах шифрування (чи криптографії з відкритим ключем) для зашифрування інформації використовують один ключ (відкритий), а для розшифрування - інший (секретний). Ці ключі різні і не можуть бути отримані один з одного.

Схема обміну інформацією така:

- одержувач вибирає відкритий та секретний ключі, секретний ключ зберігає в таємниці, відкритий же робить доступним (повідомляє відправнику, групі користувачів мережі, публікує);
- відправник, використовуючи відкритий ключ одержувача, зашифровує повідомлення, що пересилається одержувачу;
- одержувач одержує повідомлення і розшифровує його, використовуючи свій секретний ключ.

## 3. Реалізація алгоритмів шифрування

Алгоритми шифрування реалізуються програмними чи апаратними засобами. Є безліч програмних реалізацій різних алгоритмів. Через свою дешевість (деякі і зовсім безкоштовні), а також через зростання швидкодії процесорів ПЕОМ, простоти роботи і безвідмовності, вони дуже конкурентноздатні. Широко відома програма Diskreet з пакета Norton Utilities.

Не можна не згадати пакет PGP (Pretty Good Privacy, версія 2.1, автор Philip Zimmermann), у якому комплексно вирішені практично всі проблеми захисту переданої інформації. Застосовано стиснення даних перед шифруванням, могутнє керування ключами, симетричний (IDEA) і асиметричний (RSA) алгоритми шифрування, обчислення контрольної функції для цифрового підпису, надійна генерація ключів.

Публікації журналу "Монітор" з докладними описами різних алгоритмів і відповідними лістингами дають можливість кожному бажаючому написати свою програму (чи скористатися готовим лістингом).

Апаратна реалізація алгоритмів можлива за допомогою спеціалізованих мікросхем (виробляються кристали для алгоритмів ДН, RSA, DES, Skipjack, ДСТ 28147-89), або з використанням компонентів широкого призначення (через дешевину і високу швидкодію перспективні цифрові сигнальні процесори

Серед російських розробок слід зазначити плати "Криптон" (фірма "Анкад") [2] і "Грим" (методологія й алгоритми фірми "ЛАН-Крипто", технічна розробка НПЦ "ЭЛиПС").

"Криптон" - одноплатні пристрої, що використовують криптопроцесори (спеціалізовані 32-розрядні мікроЕОМ, що також називаються "блюмінг"). Блюмінги апаратно реалізують алгоритми ДСТ 28147-89. Вони складаються з обчислювача й ОЗП для збереження ключів. Причому в криптопроцесорі є три області для збереження ключів, що дозволяє будувати багаторівневі ключові системи.

Для більшої надійності шифрування одночасно працюють два криптопроцесори. Блок даних у 64 біт вважається правильно зашифрованим якщо збігається інформація на виході обох блюмінгів. Швидкість шифрування - 250 КБ/с.

Крім двох блюмінгів на платі розташований:

- контролер сполучення із шиною комп'ютера (за винятком "Криптон-ес" плати розраховані на роботу із шиною ISA);
- BIOS плати, призначений для здійснення інтерфейсу з комп'ютером і виконуючий самотестування пристрою і введення ключів у криптопроцесори;
- датчик випадкових чисел (ДВЧ) для вироблення ключів шифрування, виконаний на шумових діодах.

Випускаються наступні різновиди плат "Криптон":

- "Криптон-ес" призначена для ПЕОМ серії ЕС 1841-1845;
- "Криптон-3";
- "Криптон-4" (скорочені габаритні розміри за рахунок переміщення ряду дискретних елементів у базові кристали, підвищена швидкість обміну завдяки внутрішньому буферу на 8 байт);
- "Криптон-ик" додатково оснащена контролером ПК (інтелектуальна картка).

У пристроях "Криптон-ес", "Криптон-3", "Криптон-4" ключі зберігаються у виді файлу на дискеті. У "Криптон-ик" ключі знаходяться на ІК, що затруднює підробку і копіювання.

У платі "Грим" використовуються цифрові сигнальні процесори фірми Analog Devices ADSP-2105 і ADSP-2101, що дає швидкість шифрування відповідно 125 і 210 КБ/с. На платі є фізичний ДВЧ і ОЗП з програмами початкового тесту, перевірки прав доступу, завантаження і генерації ключів. Ключі зберігаються на нестандартно форматованій дискеті. Плата реалізує алгоритм ДСТ 28147-89.



Національний університет  
застосування  
та природокористування

#### 4. Текст програми

Програма написана на мові програмування *Object Pascal* в середовищі програмування *Delphi*.

*unit Unit2;*

*interface*

*uses*

*Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, ComCtrls, ToolWin, StdCtrls, Menus, StdActns, ActnList, ImgList, Buttons, Mask, Spin;*

*Type* {розділ описання використаних компонентів, що були розміщені на формі}

*TMain = class(TForm)*  
*StatusBar1: TStatusBar;*  
*Panel1: TPanel;*  
*ToolBar1: TToolBar;*  
*ToolButton1, ToolButton2, ToolButton3, ToolButton4, ToolButton5,*  
*ToolButton6, ToolButton7, ToolButton8: TToolButton;*  
*MainMenu1: TMainMenu;*  
*N1, N2, N3, N4, N5, N6, N7, N8, N9, N10, N11, N12, N13, N14, N15, N16,*  
*N21, N31, N41, N51: TMenuItem;*  
*GroupBox1: TGroupBox;*  
*Mem1: TMem;*  
*RadioGroup1: TRadioGroup;*  
*OpenDialog1: TOpenDialog;*  
*SaveDialog1: TSaveDialog;*  
*ActionList1: TActionList;*  
*ImageList1: TImageList;*

```

Open: TAction;
Save: TAction;
Close: TAction;
Saveas: TAction;
Exit: TAction;
EditCopy1: TEditCopy;
EditCut1: TEditCut;
EditDelete1: TEditDelete;
EditPaste1: TEditPaste;
EditSelectAll1: TEditSelectAll;
EditUndo1: TEditUndo;
Label1: TLabel;
SpinEdit1: TSpinEdit;
Button1: TButton;
Button2: TButton;
BitBtn1: TBitBtn;
Shifr: TAction;
Deshifr: TAction;
ToolButton9: TToolButton;

```

```

procedure FormClose(Sender: TObject; var Action: TCloseAction);

```

```

procedure OpenExecute(Sender: TObject);      {розділ описання процедур
обробки подій}

```

```

procedure CloseExecute(Sender: TObject);

```

```

procedure ExitExecute(Sender: TObject);

```

```

procedure ShifrExecute(Sender: TObject);

```

```

procedure DeshifrExecute(Sender: TObject);

```

```

procedure SaveExecute(Sender: TObject);

```

```

procedure SaveasExecute(Sender: TObject);

```

```

procedure N13Click(Sender: TObject);

```

```

procedure N12Click(Sender: TObject);

```

```

procedure N16Click(Sender: TObject);

```

```

procedure N14Click(Sender: TObject);

```

```

procedure N15Click(Sender: TObject);

```

```

private

```

```

{ Private declarations }

```

```

public

```

```

{ Public declarations }

```

```

end;

```

```

var

```

```

Main: TMain;

```

```

str:string;

```

```

fname:string;

```

## *implementation*

*uses Unit1;*

*{SR \*.DFM}*

*procedure TMain.FormClose(Sender: TObject; var Action: TCloseAction);*

*{описання процедури обробки події закриття форми}*

*begin*

*start.close;*

*end;*

*procedure TMain.OpenExecute(Sender: TObject);* *{описання процедури обробки події, яка виникає при натисненні*

*командної*

*кнопки “відкрити файл”}*

*var f:textfile;*

*s:string;*

*begin*

*if OpenFileDialog1.Execute then*

*begin*

*fname:=OpenDialog1.FileName;*

*AssignFile(F, OpenFileDialog1.FileName);*

*Reset(F);*

*memo1.lines.clear;*

*while not eof(f) do*

*begin*

*readln(f,s);*

*memo1.lines.add(s);*

*end;*

*CloseFile(F);*

*end;*

*end;*

*procedure TMain.CloseExecute(Sender: TObject);* *{описання процедури обробки події, яка виникає при натисненні*

*командної*

*кнопки “закрити файл”}*

*begin*

*memo1.lines.clear;*

*end;*

```

procedure TMain.ExitExecute(Sender: TObject); {описання
процедури обробки події закриття головної форми}
begin
    messagebeep(0);
    if MessageDlg('Бу справді хочете вийти ?',mtConfirmation,
[mbYes, mbNo], 0) = mrYes then
        start.close;
    end;

```



Національний університет  
водного господарства  
та природокористування

```

function xora(key:string;text:string):string; {описання
процедури,

```

що виконує шифрування\дешифрування текстової інформації  
text, використовуючи ключ key, XOR- методом }  
**var** longkey,res:string;

```

    i : integer;
    toto: char;
begin
    res:='';longkey:='';
    for i:=0 to (length(text) div length(key)) do
        longkey:=longkey+key;
    for i:=1 to length(text) do
        begin
            toto := chr((ord(text[i]) XOR ord(longkey[i])));
            res:=res + toto;
        end;
    xora:=res;
end;

```



```

procedure TMain.ShifrExecute(Sender: TObject); {описання
процедури обробки події, яка виникає при натисненні

```

командної

```

кнопки “виконати шифрування”}
//-----спеціальна функція, яка виконує алгоритм
перестановки
//(шифрування)

```

```

function shifr(code:string;ci:integer):string;
var nove,nov:string;
    i,n,ni:integer;
begin
    nove:='';
    for i:=1 to length(code) do nove:=nove+'!';
    for i:=1 to length(code) do if code[i]='!' then code[i]='@';

```

```

n:=1;ni:=1;nov:=nove;
for i:=1 to length(code) do
begin
while n<ci do
begin
if ni>=length(nove) then ni:=ni-length(nove)+1 else
inc(ni);
if code[ni]<>'.' then inc(n);
end;
n:=1;
nove[i]:=code[ni];code[ni]:= '.';
while (code[ni]='.') and (nov<>code) do
if ni>=length(nove) then ni:=ni-length(nove)+1 else
inc(ni);
end;
for i:=1 to length(nove) do if nove[i]='@' then nove[i]:= '.';
shifr:=nove;
end;

```



Національний університет  
водного господарства  
та природокористування

```

//----- власне сама процедура
var i,j:integer;
choise,key:integer; // в залежності від вибору алгоритму,
//виконується той, чи інший
keys:string;// алгоритм, результат виконання виводиться на
//компоненту
begin // memo1 з використанням властивості Lines.strings
key:=spinedit1.value;
choise:=main.RadioGroup1.ItemIndex;
case choise of
0: begin
// виконання алгоритму – шифрування методом Цезаря
for j:=0 to memo1.lines.Count-1 do
begin
str:=memo1.lines.strings[j];
for i:=1 to length(str) do
begin
if (ord(str[i])>=32) then str[i]:=chr(ord(str[i])+key mod 255);
end;
memo1.lines.strings[j]:=str;
end;
end;
1: begin // виконання алгоритму – шифрування методом
//перестановки
key:=spinedit1.value;

```



begin

```

key:=spinedit1.value;
choise:=main.RadioGroup1.ItemIndex;
case choise of
0: begin

```

// виконання алгоритму – шифрування методом Цезаря

```

for j:=0 to memo1.lines.Count-1 do

```

```

begin

```

```

str:=memo1.lines.strings[j];

```

```

for i:=1 to length(str) do

```

```

begin

```

```

if (ord(str[i])>=32) then str[i]:=chr(ord(str[i])+key mod 255);

```

```

end;

```

```

memo1.lines.strings[j]:=str;

```

```

end;

```

```

end;

```

```

1: begin // виконання алгоритму – шифрування методом

```

```

//перестановки

```

```

key:=spinedit1.value;

```

```

for j:=0 to memo1.lines.Count-1 do
begin

memo1.lines.strings[jj]=(shifr(memo1.lines.strings[jj],key));
end;
end;
2:begin //виконання алгоритму – шифрування методом
//гамування
key:=spinedit1.value;
for j:=0 to memo1.lines.Count-1 do
begin

```



Національний університет  
водного господарства  
та природокористування

```

memo1.lines.strings[jj]=(encrypt(key,memo1.lines.strings[jj]));
end;
end;
3:begin// виконання алгоритму – шифрування методом

```

XOR

```

keys:=inttostr(spinedit1.value);
for j:=0 to memo1.lines.Count-1 do
begin

```

```

memo1.lines.strings[jj]=(xora(keys,memo1.lines.strings[jj]));
end;
end;
else end;
end;

```



Національний університет  
водного господарства  
та природокористування

```

procedure TMain.DeshifrExecute(Sender: TObject); {описання
процедури обробки події, яка виникає при натисненні

```

командної

```

кнопки “виконати дешифрування”}

```

//-----спеціальна функція, яка виконує алгоритм  
перестановки

```

//(дешифрування)
function deshifr(code:string;ci:integer):string;
var nove:string;
i,n,ni:integer;
begin
nove:='';
for i:=1 to length(code) do nove:=nove+'!';
for i:=1 to length(code) do if code[i]='!' then code[i]='@';
n:=1;ni:=1;

```



*inc(ni);*

```
for i:=1 to length(code) do
  begin
    while n<ci do
      begin
        if ni>=length(nove) then ni:=ni-length(nove)+1 else
          if nove[ni]='.' then inc(n);
        end;
        n:=1;
        nove[ni]:=code[i];
        if i<>length(nove) then
          begin
            while nove[ni]<>'.' do
              if ni>=length(nove) then ni:=ni-length(nove)+1 else
                inc(ni);
            end;
          end;
        for i:=1 to length(nove) do if nove[i]='@' then nove[i]='.';
        deshifr:=nove;
      end;
```

//-----власне і є сама процедура

```
var chose,key,j,i:integer;
```

```
keys:string;
```

```
begin
```

```
key:=spinedit1.value;
```

```
chose:=main.RadioGroup1.ItemIndex;
```

```
case chose of
```

```
0:begin
```

// виконання алгоритму – дешифрування методом Цезаря

```
for j:=0 to memo1.lines.Count-1 do
```

```
begin
```

```
str:=memo1.lines.strings[j];
```

```
for i:=1 to length(str)+1 do
```

```
begin
```

```
if (ord(str[i])>=32)then str[i]:=chr(ord(str[i])-key mod 255);
```

```
end;
```

```
memo1.lines.strings[j]:=str;
```

```
end;
```

```
end;
```

```
1:begin // виконання алгоритму – дешифрування методом
```

```
//перестановки
```

```
key:=spinedit1.value;
```

```
for j:=0 to memo1.lines.Count-1 do
```

**begin**

```
memo1.lines.strings[j]:=(deshifr(memo1.lines.strings[j],key));
```

```
end;
```

```
end;
```

```
2:begin
```

```
//виконання алгоритму – дешифрування методом гамування
```

```
key:=spinedit1.value;
```

```
for j:=0 to memo1.lines.Count-1 do
```

```
begin
```

```
memo1.lines.strings[j]:=(encrypt(key,memo1.lines.strings[j]));
```

```
end;
```

```
end;
```

```
3:begin
```

```
// виконання алгоритму – дешифрування методом XOR
```

```
keys:=inttostr(spinedit1.value);
```

```
for j:=0 to memo1.lines.Count-1 do
```

```
begin
```

```
memo1.lines.strings[j]:=(xora(keys,memo1.lines.strings[j]));
```

```
end;
```

```
end;
```

```
else
```

```
end;
```

```
end;
```

```
procedure TMain.SaveExecute(Sender: TObject); {описання  
процедури обробки події, яка виникає при натисненні
```

командної

```
кнопки “ зберегти файл ”}
```

```
var f:textfile;
```

```
i:integer;
```

```
begin
```

```
if fname<>" then
```

```
begin
```

```
AssignFile(F,FName);
```

```
Rewrite(F);
```

```
for i:=0 to memo1.lines.count-1 do
```

```
begin
```

```
writeln(f,memo1.lines.strings[i]);
```

```
end;
```



Національний університет  
водного господарства  
та природокористування



Національний університет  
водного господарства  
та природокористування

```

    CloseFile(F);
end;
end;

```

*procedure TMain.SaveasExecute(Sender: TObject);* {описання процедури обробки події, яка виникає при натисненні

командної

кнопки “ зберегти файл ”}

```

var f:textfile;
    i:integer;
begin
    if saveDialog1.Execute then
    begin
        AssignFile(F, SaveDialog1.FileName);
        Rewrite(F);
        for i:=0 to memo1.lines.count-1 do
            begin
                writeln(f,memo1.lines.strings[i]);
            end;
        CloseFile(F);
    end;
end;

```

*procedure TMain.N15Click(Sender: TObject);* {процедура виведення інформації про програму }

```

begin
    Application.MessageBox(
        ' Курсова робота на тему: "Алгоритми шифрування та дешифрування текстової інформації". ',
        ' *** Про програму *** ',
        MB_OK);
end;

end.

```

## 5. Інструкція користувача



Рис.1.

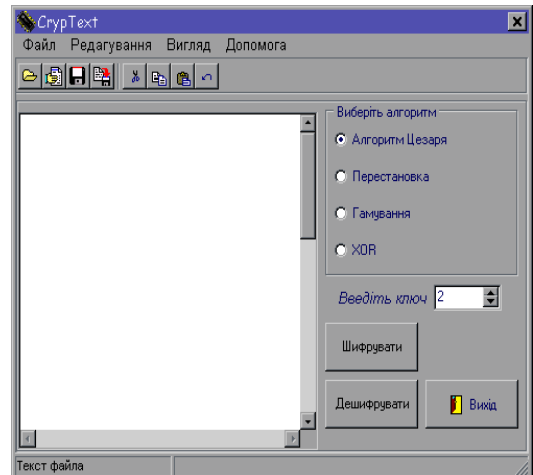


Рис.2.

При запуску програми СтурText.exe на екран відразу ж виводиться заставка із запрошенням натиснути клавішу для продовження (Рис. 1.).

При натисненні на відповідну кнопку з'являється головне вікно додатку (Рис.2.). Далі для продовження роботи з програмою шифрування\дешифрування текстової інформації користувач може: сам набрати текст, вставити текст із буфера обміну, відкрити файл із уже набраним текстом. Наступним кроком буде вибір алгоритму за яким буде здійснюватися шифрування інформації. Як видно із рис. 2., таких алгоритмів є 4:

- Алгоритм Цезаря
- Перестановка
- Гамування
- XOR

Після вибору алгоритму користувач натискає на клавішу “Шифрувати”: програма виконує переведення вихідного тексту у зашифрований. Аналогічно виконуються дії і для розшифрування закодованого тексту. Також є можливість зберегти зашифрований (розшифрований) текст у тому ж самому файлі або створити новий. Для полегшення всіх операцій всі вони винесені на панель швидкого доступу.

## 6. Тестовий приклад

В якості прикладу розглянемо результати застосування різних алгоритмів для шифрування наступного тексту програми:

***Program SAT;***



```

\iŷM| 1→
_Ÿµ-LP)VOγ ЦШi‘\—kκ
  аз-fF ↑ @J,,
yIĚ]@↑ 6▣ D↑ юШIn—R>zЯУК.
  аз|@← *¶
  аŸQW\*@ ▣ цBrФ
  аз-| \c| ← | ЫОобRЫ"ђa\|↑ VBx↓ ньЛь*Й– П.ЊД$6p↓ z&1□
  аз-| \c!! G↑ Ц<,Д

```



```

ТмШТ@5-Vрситет
uСКЯ6сподарства
користування
  аз-| \cZ ↑ ц==Ф← T9ђ—5L!! ☼ L”
  аз[K↑ x
  ,ŸYL↑
  а,,RW☼ ▣ →
  а°LL▣ &Rγ ☼ –Ѐ
  а†PV*!! BWЫУ&
  а°LL▣ &Rγ ☼ –Ѐ
  аμ[D↑ (XL
  ...©Z

```

Порівнявши зашифровані тексти можна сказати, що алгоритм гамування більш криптостійкий (у даній програмі XOR-шифрування виконане для кожного символу кодової таблиці окремо, а гамування – зашифрує цілий рядок). Ця стійкість пояснюється досить просто: перший символ зашифровується за вказаним ключем, а інші зашифровуються вже зміненим ключем.

Розглянемо також простий алгоритм перестановки. Отримаємо

```

g ;rAoTmaSrP
ste;c rsu
,neajt,eriirgv:n
h ;;c Cra
crnieoecdeAodirc usP;
Vne ig
o:t5og io ef06i 2n b r = d
ch)r(-: c;t3'; (i'c): :3 i-'r- i=' ,eW
i= nay f0ed 0r; heie kt1
  e ; d n
e dn;
gB eni
lrC;src
r(;i7w)et#
nieAc i do;cs

```

r(;i7w)et#  
ee kr;y da  
dn.E

Після застосування алгоритму Цезаря дістанемо

Rtqitco"UCV=  
wugu"etv=  
xct"k.l.p<kpvgigt=  
"E<ejct=  
Rtqegfwtg"Сpuekkeqfg=  
"Dgikp  
"hqt"k<?2"vq"478"fq"dgikp  
"e<?ejt\*k+="Ytkvg\*k<5.)///).e<5.)")+=  
"kh"k?322"vjgp"tgcfmg{=  
"gpf=  
"gpf=  
"Dgikp  
"Entuet=  
"ytkvg\*%9+=  
"Сpuekkeqfg=  
"ytkvg\*%9+=  
"tgcfmg{=  
"Gpf0

Усі алгоритми використані в даній курсовій роботі мають дуже низьку криптостійкість порівняно з сучасними алгоритмами такими як: RSA, DES, ДСТ 28147-89 і багато інших. Для збільшення їхньої криптостійкості рекомендовано користуватися алгоритмами стиснення інформації перед власне зашифруванням.

### **Висновки**

Варто відмітити, що шифрування інформації не є панацеєю. Його варто розглядати тільки як один з методів захисту інформації і застосовувати обов'язково в сполученні з законодавчими, організаційними й іншими мірами. Отже, розроблена мною програма має відчутні переваги в порівнянні з аналогами, вона досить компактна, що досить важливо на даний час, цю програму можна переносити на гнучких носіях інформації (HD). Маю наміри надалі вдосконалювати дану програму.

### **Література**

1. С. Мафтик. Механизмы защиты в сетях ЭВМ – М.: Мир, 1993. – 456с.
2. В. Ковалевский. Криптографические методы. – С.-Пб.: Компьютер Пресс, 1993. – 247 с.
3. В. Водолазкий. Стандарт шифрования ДЕС //Монитор. – 1992, №3-4.

–С.25-47.

4. С. Воробьев. Защита информации в персональных ЭВМ. – М.: Мир, 1993. – 341 с.
5. Спесивцев А.В., Вегнер В.А., Крутяков А.Ю. и др. Защита информации в персональных ЭВМ. - М.: Радио и связь, 1992. – 452 с.
6. Сяо Д., Керр Д., Мэдник С. Защита ЭВМ. - М.: Мир, 1982. – 380 с.
7. ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования.



Національний університет  
водного господарства  
та природокористування



Національний університет  
водного господарства  
та природокористування



## РОЗДІЛ VI. Елементи програмування інтелектуальних систем

### Загальні вимоги до виконання лабораторних робіт

Вся тематика лабораторних робіт спрямована на розробку найпростіших інтелектуальних систем в середовищі програмування *Delphi*. Відповідно до вимог програми з даної дисципліни розроблено 4 лабораторні роботи, виконання яких кожним студентом є обов'язковим для допуску до здачі відповідного іспиту (заліку). Лабораторні роботи містять приклади програм з необхідними теоретичними відомостями.



Економічного господарства  
та природокористування

### Порядок виконання лабораторних робіт

1. Ознайомитись з темою, метою роботи.
2. Користуючись прикладами, які наведені в кожній лабораторній роботі, вивчити методи роботи з компонентами.
3. Вибрати відповідний варіант (за вказівкою викладача) та ознайомитись із завданням.
4. Розробити проект, що означає виконати наступні етапи:
  - проектування інтерфейсу проекту.
  - програмування: здійснити програмування процедур опрацювання подій.
5. Зберегти результати виконаної роботи в індивідуальній робочій папці.
6. Продемонструвати результати виконаної роботи викладачу.
7. Оформити звіт про виконання лабораторної роботи, який включає: назву теми, завдання, хід роботи, тексти програм, коментарі до них.
8. Захистити звіт про виконання лабораторної роботи. При захисті вимагається пояснити всі етапи виконання лабораторної роботи та відповісти на контрольні питання, які зазначені в кінці лабораторної роботи.

### Лабораторна робота №1

**Тема:** Розробка найпростішої інтелектуальної системи для здійснення діалогу.

#### **Мета роботи:**

- Ознайомитись з прийомами створення обличчя уявного співрозмовника в середовищі візуального об'єктно-орієнтованого програмування *Delphi*.
- Ознайомитись з найпростішими формами діалогу.

- Ознайомитись з прийомами організації багатокрокового діалогу в середовищі візуального об'єктно-орієнтованого програмування *Delphi*.

### *Теретичні відомості з елементами прикладу:*

Для побудови уявного співрозмовника використати компоненту *TShape*, яка міститься на вкладці *Additional* палітри компонентів середовища *Delphi*. Дану компоненту тільки умовно можна віднести до засобів відображення графічної інформації, оскільки дана компонента – це просто різні геометричні фігури, які відповідним чином заштриховані.



Серед властивостей даної компоненти відмітимо наступні Властивість *Shape* визначає форму фігури і може приймати наступні значення :

- 7) *stCircle* – круг;
- 8) *stEllipse* – еліпс;
- 9) *stRectangle* – прямокутник;
- 10) *stSquare* – квадрат;
- 11) *stRoundRect* – прямокутник із заокругленими кутами;
- 12) *stRoundSquare* – квадрат із заокругленими кутами.

Властивість *Brush* (пензель) є об'єктом типу *TBrush*, що має ряд підвластивостей, зокрема: колір (*Brush.Color*) та стиль (*Brush.Style*) заливки фігури. Властивість *Pen* (перо) визначає стиль ліній.

Для проведення анімації використати невізуальну компоненту *TTimer*, яка міститься на вкладці *System* палітри компонентів, та компоненту *TButton*, яка розташована на сторінці *Standard*.

Компонента *TButton* дозволяє створювати командну кнопку. Властивість *Caption* визначає надпис або ще називають заголовком. Основна подія будь-якої кнопки – *OnClick*, що виникає при одноразовому натисканні мишею. Для обробки даної події автоматично створюється процедура, в якій записуються оператори, що повинні виконуватися при одноразовому натисканні мишею.

Для проведення діалогу використати компоненти *TEdit* та *TLabel*, які розміщені на вкладці *Standard*

Компонента *TEdit* призначена для введення/відображення одного текстового рядка. Для даної компоненти текст, що вводиться та виводиться є значенням властивості *Text* типу *String*. Дану властивість можна задавати під час проектування або програмно.

Властивість *MaxLength* визначає максимальну довжину тексту, що вводиться (якщо *MaxLength* = 0, то довжина тексту необмежена). Властивість *Modified* доступна тільки під час виконання програми та показує, чи проводилось редагування тексту у вікні редактора.

Серед найбільш використовуваних методів можна назвати метод **SetFocus**, за допомогою якого здійснюється передача фокуса елементу керування.

Призначенням компоненти **TLabel** є текстовий підпис. Текст, що відображається у компоненті **TLabel**, визначається значенням її властивості **Caption**.

Колір фону компоненти **TLabel** визначається властивістю **Color**, а колір тексту – підвластивістю **Color** властивості **Font**. Розміщення компоненти на формі визначається, зокрема властивостями **Top**, **Left**, **Height**, **Width**, **Aline**, **Anchor**, **Constraints**. Ці властивості визначають координати компоненти, її розміри та їх зміни при зміні користувачем розмірів батьківського вікна.



*Створити уявного співрозмовника та придумати трьохкроковий діалог на тему згідно Вашого варіанту та запрограмуйте його. Передбачте всі фрази та ефекти, які згадані в теоретичних відомостях.*

### Створення співрозмовника

Для побудови уявного співрозмовника використаємо вісім компонент TShape. На рис. 6.1 показана орієнтовна компоновка «обличчя» нашого майбутнього співрозмовника з використанням восьми компонент Shape1, Shape2, ... Shape8.

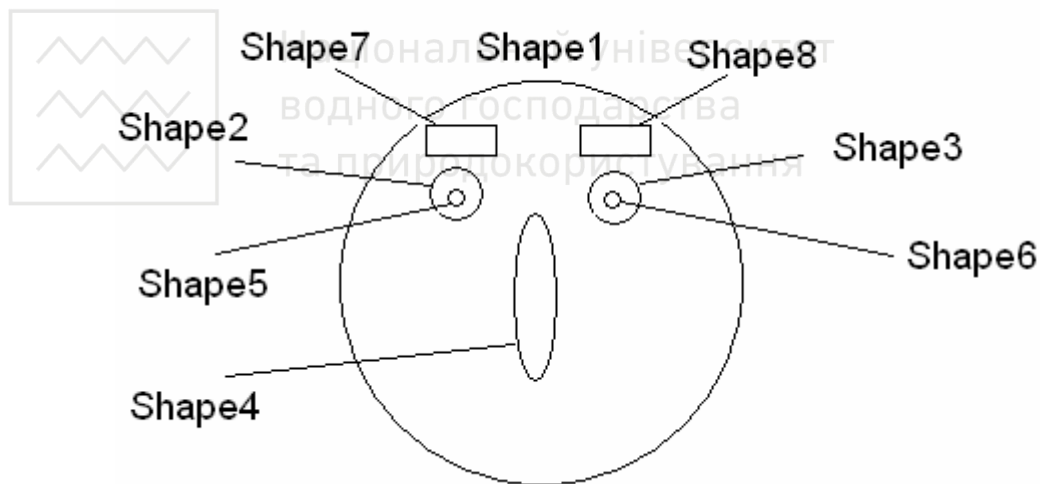


Рис. 6.1 Компоновка <<обличчя>> з фігурок

Геометричні розміри і компонент на формі можна змінювати за допомогою миші, пересуваючи один з обмежуючих компоненту маркерів або визначаючи відповідні параметри властивостей компонент в

**Інспектори об'єктів.** За допомогою **Інспектора об'єктів** можна встановити і необхідний колір фігур, звертаючись до її властивостей **Brush** (пензлик), а в ньому (два рази клацнути мишкою в полі + Brush) – до пункту Color (Колір).

Розміри фігур можуть бути довільними. Єдине обмеження – ширина фігурок Shape7 і Shape8 повинна бути достатньою для того, щоб ці фігурки могли повністю накривати собою стилізовані „очі”, що відповідають фігурці Shape2 і Shape3. Фігурки Shape7 і Shape8 будуть відігравати роль „вій”, тому колір цих фігурок краще підібрати так, щоб він співпадав з кольором фігурки Shape1. Контур фігурок Shape7 і Shape8 також можна зробити невидимим. Для цього у вікні **Інспектора об'єктів** вказаних компонент для властивості Pen (Перо) потрібно вибрати під властивість **Color** і вибрати те значення, котре співпадає з назвою кольору фігури Shape1.

### Найпростіша анімація.

Запрограмуємо найпростішу анімацію – моргання віями. Для цього розмістимо в формі компоненту **TTimer1**. Компонента **TTimer1** під час виконання програми невидима, тому її можна помістити в будь-яке місце форми. Призначення компоненти в межах даного завдання – сповіщати систему кожен раз після закінченню певного інтервалу часу. В **Інспекторі об'єктів** компоненти **TTimer1** значення властивості **Interval** (Інтервал) по замовчуванню встановлено в 1000 мілісекунд, таким чином, наш таймер буде зрацьовувати рівно через секунду. Для спритних вій така реакція дуже сповільнена, тому слід встановити нове значення для властивості **Interval** 400 мілісекунд. Властивість **Enabled** компоненти **TTimer1** потрібно встановити в **false** для того, щоб на початку годинник був вимкнений.

Процес моргання віями ми розіб'ємо на два етапа :

- 1) опускання вій ;
- 2) підняття вій кожні 400 мілісекунд.

Щоб виконати перший етап, потрібно розмістити на формі кнопку **TButton1**. При подвійному натисканні лівою клавішею миші на кнопці, отримаємо шаблон процедури обробки події **OnClick** для цієї кнопки: **TForm1.Button1Click**. Для опускання вій можна збільшувати висоту (властивість **Height**) фігурок **Shape7** і **Shape8**. Припустимо, значення властивості **Height** для фігурок **Shape2** і **Shape3** ( очей ) становить 33 пікселя, а висота фігурок **Shape7** і **Shape8** становить 25 пікселів. Тоді в тілі процедури обробки подій **OnClick** для кнопки **TButton1** слід зазначити наступне:

```
Shape7.Height : = 58 // оскільки 58 = 25+33
Shape8.Height : = 58
Timer1.Enabled : = true; // цей оператор запускає
//таймер
```

Однак, це не єдиний спосіб закрити очі створеному уявному співрозмовнику. Можна, наприклад, повністю перемістити прямокутник вий вниз, програмним шляхом змінюючи його властивість *Top*.

Реакцію виконану роботу таймера по закінченню 400 мілісекунд напишемо в тілі процедури ***TForm1.Timer1Timer***:

```
Shape7.Height := 58 // Поновлюємо  
Shape8.Height := 58 // Попередні значення  
Timer1.Enabled := false; // І відключаємо таймер
```

Зверніть увагу, що в цій процедурі таймер відключає сам себе. Наступний його запуск буде виконано при натиску кнопки *Button1*.

Тепер все готово до запуску програми (F9). Наше творіння почало подавати перші ознаки життя. Домовимось називати його «Томагочі» - деяка подібність з відомою іграшкою в нього є.

### Сценарій діалога «А я ...»

Перед тим, як навчить нашого «Томагочі» розмовляти, давайте познайомимось з найпростішими формами діалога. Наведемо приклад найпримітивнішого типу діалога:

- А у нас сьогодні гість.  
А у вас?
- А у нас в квартирі газ.  
А у вас?
- А у нас водопровід.  
Ось...

Щоб вести такий „діалог”, зовсім не потрібно звертати увагу на відповіді співрозмовника. Це зовсім не діалог, а змішування кількох монологів або розмова глухих, котрим попередньо повідомили тему розмови. З точки зору програмування, в організації „А я...” – діалога роль співрозмовника можна звести до натискання клавіші ***Enter***, реакцією на котру буде чергова фраза комп’ютера: « А я...», «А ми...», «А у мене...». У цьому випадку комп’ютер мало чим буде відрізнятись від інших джерел інформації: родію, телевізора і т. п. Наш „ Томагочі” повинен бути більш цікавим.

### Однокроковий діалог

Наступним за рівнем складності є *однокроковий діалог*, що організовується за сценарієм: „одне питання – одна відповідь”.

Для введення питань до „Томагочі” використаємо спеціальне вікно ***TEdit1*** (вікно редагування). Слід визначити достатню ширину даної компоненти, щоб можна було розмістити невелике речення.

Для відповідей „Томагочі” використаємо компоненту ***TLabel1***, яку розмістимо на рівні рота „Томагочі”. Очистимо поле ***Caption*** у вікні Інспектора об’єктів компоненти ***TLabel1*** та визначимо потрібні характеристики шрифту .

Для ознайомлення «Томагочі» з текстом вікна *TEdit1* використаємо вже існуючу кнопку *TButton1*. В тілі процедури *TForm1.Button1Click* потрібно вставити наступні рядки:

```
If Edit1.Text = "Привіт!"  
then Label1.Caption := "Добрий день!"
```

Під час експерименту можуть з'явитися цілком природні питання. Чому „Томагочі” не реагує на фрази „привіт!”, „Привіт”, „Привіт ! та інші?

Для початку слід забезпечити те, щоб „Томагочі” не звертав увагу на те, великими чи маленькими буквами набрана звернена до нього фраза. Для цього в Інспекторі об'єктів значення властивості *CharCase* компоненти *TEdit1* задати як *ecLowerCase*. В цьому випадку при читанні символів із вікна *TEdit1* вони будуть перевтілюватися до нижнього регістру.

Можна розширити кругозір „Томагочі”, помістивши, наприклад, замість попереднього умовного речення більш універсальне:

```
If (Edit1.Text = "Привіт!") or  
(Edit1.Text = "привіт") or  
(Edit1.Text = "привіт !") or  
(Edit1.Text = "здраствуй!") or  
(Edit1.Text = "Салют !") or  
(Edit1.Text = "Hi !") // цей ряд можна  
// продовжити  
then Label1.Caption := "Добрий день!" :
```

Для того, щоб фраза „Добрий день!” не застрягала в роті „Томагочі”, проведемо наступні дії. Візьмемо ще один таймер *TTimer2* та розмістимо його на формі зі стандартним інтервалом 1000 мілісекунд. Попередньо потрібно його вимкнути, задаючи значення властивості *Enabled* рівною *false*. У процедурі *Form1.Button1Click* поряд з оператором *Timer1.Enabled:=true*; розмістимо аналогічно *Timer2.Enabled := true*;

Після цього в новій процедурі *TForm1.Timer2Timer* зазначимо:

```
Label1.Caption := " ; // присвоєння пустої стрічки  
// приводить до очистки застряглої  
// у роті „Томагочі” фрази  
Timer2.Enabled := false: // відключення таймера 2
```

### Вдосконалення програми

Вміння вимовляти єдину фразу „Добрий день!” зовсім не схвально характеризує розумові здібності „Томагочі”. Перед тим, як навчити його іншим хитрощам, нам зручно буде ввести деяку допоміжну змінну для зберігання зверненої до „Томагочі” фрази.

В процедурі *TForm1.Button1Click* потрібно описати змінну  
*s: string;*

Присвоюємо даній змінній значення, яке буде введено у вікно *TEdit1*. В тілі процедури *TForm1.ButtonClick* запишемо:

```

s := Edit1.Text;
if s = 'як тебе звали?'
then Label1.Caption := 'Гоша' ;
if s = 'скільки тобі років?'
then Label1.Caption := 'двадцять' ;
if s = 'як життя?'
then Label1.Caption := 'Нічого' ;
if s = 'Тобі подобається програмування?'
then Label1.Caption := 'Дуже !' ;

```



Національний університет  
водного господарства  
та природокористування

Чим більше запрограмовано фраз, тим більше кмітливим буде здаватися "Томагочі".

Дещо поліпшимо також процедуру роботи з вікном *TEdit1*, помістивши наприкінці після всіх умовних пропозицій два рядки:

```

Edit1.Text := ''; // очищення вікна TEdit1
Edit1.SetFocus; // підготовка вікна
// до введення нового рядка
// символів

```

Після виконання останньої команди у вікні *TEdit1* з'явиться курсор, що буде підморгувати, немов запрошуючи ввести в це вікно новий текст.

Але не можна передбачити весь набір фраз, з якими комусь надуматися звертатися до нещасливого "Томагочі". Зараз ми передбачимо невелику "хитринку" - вона допоможе "Томагочі" викручуватися з деяких скрутних ситуацій. Ідея така: якщо наприкінці зверненої до "Томагочі" незрозумілої фрази стоїть знак оклику, то малоймовірно, що на цю фразу буде потрібна якась зрозуміла відповідь (все-таки між знаками питання і оклику є деяке розходження). У даному випадку можна обробити якою-небудь нічого не означаючою реплікою.

Але як визначити, що наприкінці фрази стоїть знак оклику? Уявимо собі фразу яка знаходиться в рядку *s*, як потяг, у кожному вагоні якого сидить окремий символ. Усі вагони цього потяга пронумеровані: 1, 2, 3, ... (насправді, у цьому потязі мається вагон і з нульовим номером, але він не призначений для перевезення букв). Номер останнього вагона ми довідаємося, звернувшись до функції *length(s)* (англійське слово *length* - довжина), що мається в бібліотеці функції *Object Pascal*. Наприклад, якщо в рядку *s* знаходиться фраза "який гарний день!", то ми маємо справу з потягом, що показаний на рис. 6.2.

я	к	и	й		г	а	р	н	и	й		д	е	н	ь	!
---	---	---	---	--	---	---	---	---	---	---	--	---	---	---	---	---

„потяг” *s*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17																

номер вагону

Рис. 6.2. Фраза в стрічці s.

У даному випадку **length(s)** становить 17.

Як прочитати символ, що знаходиться в четвертому вагоні потягу? — дуже просто: s[4] (там сидить буква й). В десятому вагоні? — s[10] (там сидить буква и). В останньому вагоні? — s[length(s)] (там сидить окличний знак). Чому ми замість конкретного номера останнього вагону 17 пишемо громіздку конструкцію length (s)? Тому що ця конструкція годиться для потягу довільної довжини, — якраз те, що нам потрібно, адже ми не можемо наперед *передбачити*, скільки букв буде у вхідній фразі (скільки вагонів буде потрібно для їх розміщення).

Отже, до умовних пропозицій програми «Томагочі» додаємо ще одну пропозицію:

```
If (length(s) > 0) and (s[length(s)] = '!')
then Label 1.Caption := 'Ну-ну';
```

```
// або яка-небудь інша відповідь
// такого типу
```

### Перевірка, чи містить фраза задане слово?

Якщо хтось звертається до «Томагочі» з пропозицією «А я...», то і наш вихованець може відповісти в тому ж дусі: «А я вмю кліпати очками», або «А у мене 64 мегабайти оперативної пам'яті». Розпізнавати деяку групу символів в пропозиції краще всього за допомогою спеціальної функції мови pos(s1,s2), яка видає номер першого співпадаючого символу, якщо рядок s1 міститься серед символів рядка s2, і 0 інакше. Наприклад, словосполучення «А я» в зверненій до «Томагочі» пропозиції можна відстежити за допомогою наступного коду:

```
If pos ('А я', s) > 0 then // відповідна репліка «Томагочі»
```

При отриманні довгої низки незрозумілих йому фраз, «Томагочі» буде тільки старанно кліпати очками. У його співбесідника може скластися мінливе враження, що «Томагочі» взагалі не здатний вести діалог цивілізованим чином. Щоб цього не трапилося, передбачимо більш природну реакцію, що не викликає пересудів.

Якщо фраза не знайома «Томагочі», то вона відсутня у всіх умовних пропозиціях нашої програми. Як це перевірити? Тут може стати в нагоді прийом, що пов'язаний з використанням прапорця — деякої логічної ознаки. На початку виконання процедури **TForm1.Button1Click** опустимо прапорець. Якщо в якійсь умовній пропозиції аналізована фраза виявиться, піднімемо прапорець. В кінці перевіримо, підіймався прапорець чи ні. Якщо не підіймався, значить, фраза виявилася невідомою.



Опишемо змінну, яка гратиме роль логічної ознаки або прапорця типу **boolean** (спеціальний тип, його назва пов'язана з ім'ям англійського математика Джорджа Буля (1815-1864), що розробив теоретичні основи алгебри логіки):

**f: boolean;**

В нашому випадку тип **boolean** визначає допустиму множину значень змінної **f: true** і **false**.

Першим оператором вже відомої процедури **TForm1.Button1Click** тепер у нас буде



**f := false; // опустимо прапорець**

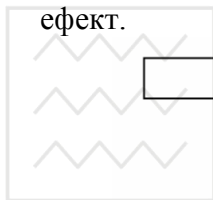
Його ми розмістимо після слова **begin** в будь-якому місці перед умовними пропозиціями. Кожну умовну пропозицію переробимо за зразком:

```
if s = 'npuvim!' then
begin
  Labell.Caption := 'Добрий день!';
  f := true; // підніmemo прапорець
end;
```

Після титанічної роботи по переробці всіх умовних пропозицій в кінці додамо нову умовну пропозицію:

```
if not f then Labell.Caption := 'Що-що?';
```

Було б непогано, якби у цей момент біля «Томагочі» з'явилося імпровізоване вухо (рис. 6.3). Спробуйте самостійно запрограмувати цей ефект.



**Діалог**

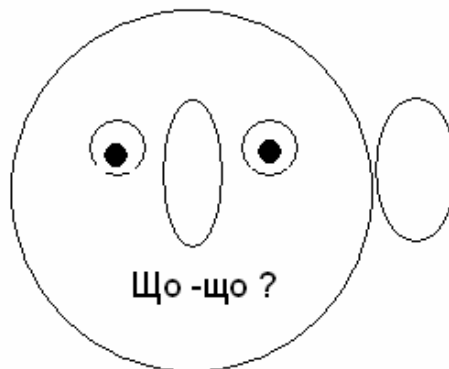


Рис. 6.3. Вид проекту після змін

Не дивлячись на всю простоту однокрокового діалогу, сценарії такого типу знаходять вживання в серйозних комерційних додатках, наприклад в автоматизованих довідкових системах, що передбачають голосове спілкування і мобільний зв'язок. Безумовно, в таких системах застосовуються також спеціальні програми розпізнавання мови.

### Двокроковий діалог

Що повинен робити комп'ютер, одержавши питання: «Скільки вона коштує?»



Під займенником *вона* може ховатися все, що завгодно: жаба, гармата або фабрика по виробництву брязкалець. Але якщо це питання прозвучало в контексті попереднього питання: «Де можна купити іграшку Томагочі?», то ясно, що він торкається саме іграшки Томагочі, і на нього можна відповісти цілком конкретно. Отже, іноді доводиться аналізувати відразу два питання: дане і попереднє. Перше, що приходить на думку при спробі організації двокрокового діалогу, — скористатися вже знайомими нам логічними прапорцями. Якщо, скажімо, відбулася розмова про місце купівлі іграшки, то можна підняти спеціальний прапорець, а потім, коли мова піде про вартість іграшки, — перевірити, чи підіймався цей прапорець чи ні. Проте тут ми стикаємося з однією принциповою трудністю.

Як тільки відбувається натискання лівою клавішею миші на кнопці **TButton1**, операційна система відводить пам'ять для описаних в розділі `var` процедури **TForm1.Button1Click** змінних. Після цього починають послідовно виконуватися оператори процедури, описані в її тексті. Після завершення виконання останнього оператора процедури, комп'ютер звільняє виділену для неї пам'ять — він «забуває» про всі внутрішні змінні процедури, з якими тільки що мав справу. Змінні, доступні тільки під час виконання процедури, називаються **локальними** (*місцевими*) змінними даної процедури. В них не можна зберігати дані, якими можна б було скористатися при черговому запуску процедури. Якщо в цьому все ж таки виникає необхідність, дані потрібно «здати на зберігання» **глобальним** (*загальним*) змінним, описаним *зовні* інших процедур і доступним всім іншим процедурам, коли б вони не виконувалися.

Існує декілька способів опису глобальних змінних. Давайте умовимося, якщо це не буде обумовлено особливо, описувати всі потрібні нам глобальні змінні в розділі **var**, що розміщується у верхній частині великого програмного розділу, в якому створюються заготовки всіх наших процедур. Як ви вже, напевно, встигли помітити, всі ці заготовки розміщені в розділі **implementation** (виконання).

Прапорець, який може бути піднятий під час одного проходу по операторах процедури **TForm1.Button1Click** з тим, щоб під час іншого проходу на нього звернули увагу, звичайно ж, повинен бути глобальною ознакою.

У верхній частині розділу *implementation* після службового коментаря `{SR *.DFM}` (що згенерував, *Delphi* ) опишемо глобальну змінну:

```
var
  flagTomagochi: boolean = false;
```

Для глобальних змінних дозволяється безпосередньо вказувати початкове значення (в даному випадку *false*). Зверніть увагу, що це значення задається через знак рівності, а не за допомогою оператора присвоювання. Місце операторів – на сцені, де виконуються якісь дії, а зараз поки оголошуються лише «дійові особи і виконавці».

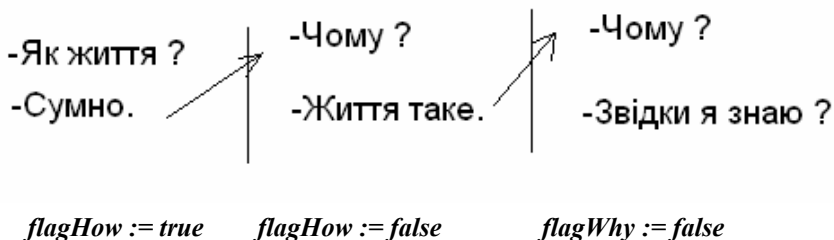
До компанії умовних пропозицій процедури *TForm1.Button1Click* додамо ще два:

```
if s = 'де можна купити Томагочи ?' then
  begin
    flagTomagochi := true;
    Label 1.Caption :=
      'В магазині «Дитячий світ»';
  end;
if (s = 'скільки вона коштує ?') and flagTomagochi
  then
  begin
    Label 1.Caption := 'Двадцять п'ять монет';
    flagTomagochi := false; // забуваємо
                          // попередня розмова
  end;
```

Запустіть програму і поекспериментуйте з порядком задання питань.

Природно чекати, що наступним по рівню складності буде трикроковий, далі — чотирікроковий, багатокроковий діалог. Проте (така вже особливість нашої мови) на практиці у більшості випадків основну роль грає саме двокроковий діалог.

Звернемо увагу на один «підводний камінь», що підкараулює розробників сценаріїв двокрокового діалогу в системах подієвого управління типу *Delphi*. Розглянемо наступну модель діалогу, в якій використовуються логічні прапори *flagHow*, *flagWhy* (рис. 6.4).



*flagWhy := true*

Рис. 6.4. Модель діалогу з логічними прапорами.

Для другої відповіді необхідна наявність питання «Чому?» і встановлення в *true* прапора *flagHow*; для третьої відповіді необхідні наявність питання «Чому?» і установка в *true* прапора *flagWhy*. Якщо в програмі обробку фраз розташувати в природному (приведеному на рис. 11.4) порядку, то у відповідь на перше питання «Чому?» на екрані відразу виникне відповідь «Звідки я знаю?», і складається враження, що відповідь «Життя таке» ігнорується. Насправді тут спостерігається ефект так званого «квазі-перестрибування».

Відповідь «Життя таке» видається, але не помічається — не встигне наш «Томагочі» моргнути очками, як фраза тут же, в повній відповідності з логікою роботи програми, замінюється на «Звідки я знаю». Щоб цього і не відбулося, потрібно в даному випадку поміняти місцями в тексті програми обробку третьої і другої фрази сценарію.

### **Завдання для виконання лабораторної роботи:**

*Створіть „Томагочі”, придумайте трикроковий діалог на тему згідно Вашого варіанту та запрограмуйте його. Передбачте всі фрази та ефекти, які згадані в теоретичних відомостях.*



Національний університет  
водоохоронного факультету  
та гідрологічного факультету

1. літній відпочинок
2. апаратна частина комп'ютера
3. кар'єра
4. подорож в інтернеті
5. полювання
6. світ спорту
7. відпочинок на природі
8. приготування їжі
9. музика
10. історія нашого краю
11. рибалка
12. автомобілі

## Лабораторна робота №2

**Тема: Розробка інтелектуальної системи для можливостями діалогу з використанням асоціативних мереж.**

### Мета роботи:



- Ознайомитись з методами організації діалогу з використанням асоціативних мереж в середовищі візуального об'єктно-орієнтованого програмування *Delphi*.
- Ознайомитись з роботою динамічної пам'яті в середовищі візуального об'єктно-орієнтованого програмування *Delphi*.

### Теретичні відомості з елементами прикладу:

Придумати керований асоціаціями сценарій розмови на тему „Прогулянка по місту”. Передбачити фрази відповідної розмови та організувати в асоціативну сітку.

Розмова про міський парк може асоціюватись з висловлюваннями про гойдалки, про водний стадіон або зоопарк. Розмова про водний стадіон може асоціюватись з висловлюваннями про морський пляж, виндсерфінг, спортивні змагання. На рис.11.5 представлений один із можливих фрагментів такої асоціативної сітки.

Приклади фраз для організації в асоціативну сітку:

PHR1: „А ми відпочивали в парку”;

PHR2: „А я гойдався на гойдалках”;

PHR3: „Мені дуже сподобались „американські гірки””;

PHR4: „А ми ходили в зоопарк”;

PHR5: „В жирафа така велика шия”;

PHR6: „А я катався на поні”;

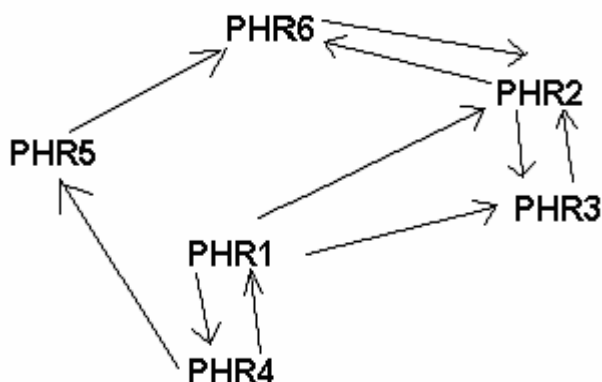


Рис. 6.5. Один із можливих фрагментів асоціативної сітки

Стрілки на малюнку визначають можливі асоціативні зв'язки. Для опису асоціативної сітки в *Delphi* зручно використовувати динамічні структури наступного типу:

```

type
  TPtr = ^TNode;
  TNode = record
    Phrase : string;
    AP : array of TPtr;
  end;

```



Національний університет  
водного господарства  
та природокористування

*TPtr* – вказівник на типовий елемент ланцюга, який може включати довільну кількість вказівників на інші елементи. Для цього використовується масив вказівників *AP* з невизначеною кількістю елементів. При опису конкретного елемента мережі кількість його вказівників відомо і довжину його масиву вказівників можна визначити за допомогою стандартної процедури мови *SetLength*. Наприклад, при описі елемента *PHR1* на рис. 8 можна застосувати наступну конструкцію:

```
var PHR1: TPtr; // Описуємо вказівник на елемент PHR1.
```

В наведеному нижче лістингу „елемент *PHR1*” означає те саме, що і фраза „елемент, на який вказує вказівник *PHR1*”.

В основному тексті програми ініціалізуємо елемент *PHR1*:

```

New(PHR1); // виділяємо пам'ять для елемента PHR1.
SetLength(PHR1^.AP,3); // Встановлюємо довжину масиву
//AP елемента PHR1, рівну // 3.
PHR1^.AP[0]:= PHR2; // Формуємо вказівники елемента
//PHR1 (див. Рис. 6.5).
PHR1^.AP[1]:=PH3; // Елементи PHR2, PHR3, PHR4
//перед цим повинні бути
PHR1^.AP[2]:=PHR4; // ініціалізовані процедурою New.
PHR1^.Phrase:='А ми відпочивали в парку'. // Формуємо
//інформаційне поле елемента PHR1.

```

```
Dispose(PHR1);
```

При формуванні інформаційних полів елементів можна використовувати датчики випадкових чисел для вибору еквівалентних по змісту фраз з деякого наперед створеного набору.

Для навігації по асоціативній мережі можна випадковим чином вибирати будь-якого з сусідів даного вузла. Якщо на нього налаштований деякий допоміжний вказівник *г*, то налаштувати його на сусідній елемент можна за допомогою наступної конструкції:

```

case random(Length(r.AP)) of
  0: r:=r.AP[0];
  1: r:=r.AP[1];
  // i m.д.
  N-1: r:=r.AP[N-1];
end;

```

Тут  $N$  – максимальна кількість вказівників в одного елементу мережі;  $Length(r.AP)$  – довжина масива  $AP$  в елементі, на який вказує  $r$ .



### Завдання для виконання лабораторної роботи:

В середовищі програмування **Delphi** розробити інтелектуальну систему для організації керованого асоціаціями діалогу:

- придумати керований асоціаціями сценарій розмови;
- передбачити фрази відповідної розмови та організувати в асоціативну сітку.

Теми діалогу наступні:

1. літній відпочинок
2. апаратна частина комп'ютера
3. кар'єра
4. подорож в інтернеті
5. полювання
6. світ спорту
7. відпочинок на природі
8. приготування їжі
9. музика
10. історія нашого краю
11. рибалка
12. автомобілі

## Лабораторна робота №3

**Тема:** Створення автоматизованої системи для розв'язку задач.

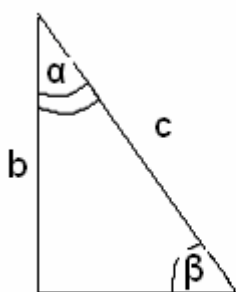
**Мета роботи:**

- Ознайомитися з принципом за яким комп'ютер шукає потрібну схему розв'язку.
- Навчитись створювати базу знань.

**Теретичні відомості з елементами прикладу:**

Як ми міркуємо, коли розв'язуємо шкільні задачі? Розглянемо одну з них. Нехай по двом катетам прямокутного трикутника необхідно потрібно визначити радіус вписаного в цей трикутник кола.

Готової формули для отримання потрібної нам відповіді в підручнику немає. Прийдеться вивести її з якихось інших формул. Що нам відомо про прямокутний трикутник? Випишемо ряд відомих співвідношень, позначаючи їх спеціальними знаками – кружками (ми навмисне відмовляємось від нумерацій формул числами, щоб підкреслити їх неупорядкованість, ми ті чи інші формули згадуємо в довільному порядку). Ряд позначень в цих формулах пояснюється на рисунку 11.6:  $R$  – радіус описаного навколо трикутника кола;  $r$  – радіус кола, вписаного в нього;  $p$  – півпериметр, а  $S$  – площа трикутника.



- |            |                             |            |                 |
|------------|-----------------------------|------------|-----------------|
| $\bigcirc$ | $\cos\alpha = b/c$          | $\bigcirc$ | $R = c/2$       |
| $\bigcirc$ | $a = b \tan\alpha$          | $\bigcirc$ | $p = (a+b+c)/2$ |
| $\bigcirc$ | $a^2 + b^2 = c^2$           | $\bigcirc$ | $S = ab/2$      |
| $\bigcirc$ | $\alpha + \beta = 90^\circ$ | $\bullet$  | $S = pr$        |



Рис. 6.6. На рисунку пояснено ряд позначень у формулах для прямокутного трикутника

Ви будете споглядати виписані формули, і раптом ви здогадаєтесь, що спочатку, використовуючи теорему Піфагора  $a^2 + b^2 = c^2$ , знайдемо гіпотенузу  $c = \sqrt{a^2 + b^2}$ , потім півпериметр  $p$ , а потім з формули  $S = ab/2$  - площу, а потім з допомогою формули  $S = pr$  визначимо потрібне значення  $r$ . Чи міг би комп'ютер здогадатися зробити це сам? – Міг би. Складемо схему зв'язку змінних та формул у вигляді мережі, яку міг проаналізувати комп'ютер (рис. 6.7).



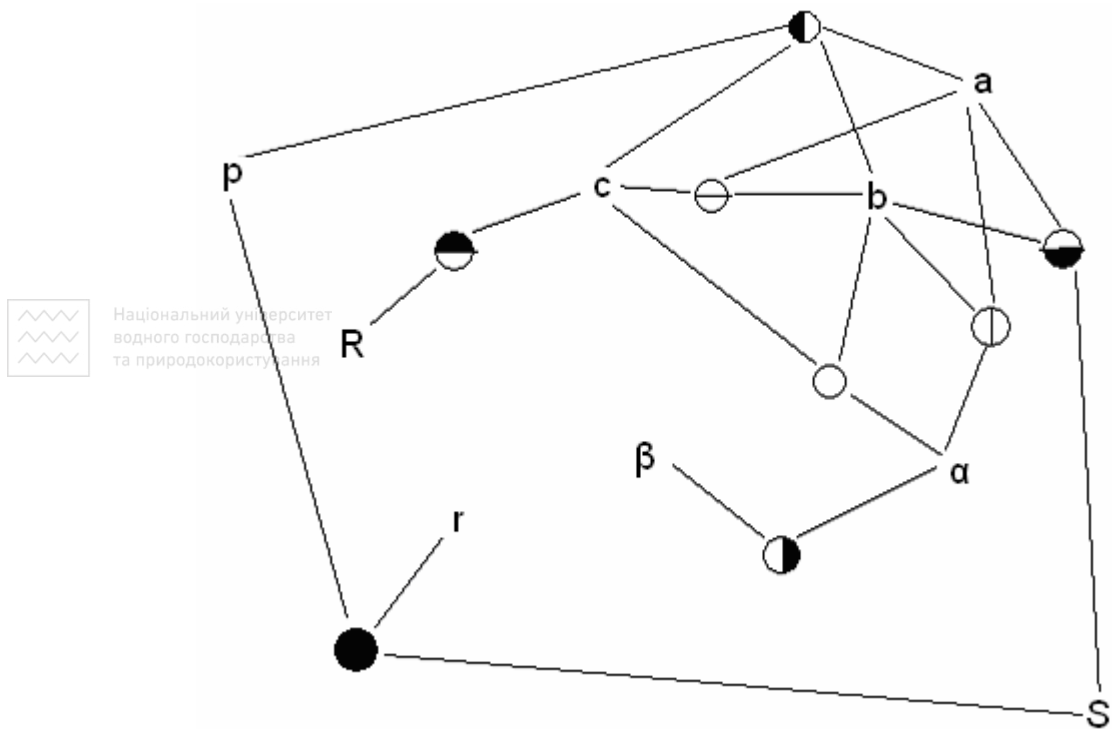


Рис. 6.7. Схема зв'язку змінних

Розглянемо принцип, по якому комп'ютер міг би відшукати потрібну схему розв'язку. Рис. 6.7 нагадує марсіанські канали. Припустимо, що це дійсно система пустих (незаповнених водою) каналів. Змінні на схемі уявимо пустими колодзями, а кружечки – розподільчими станціями, які відкривають шлюзи для пуску води в пустий канал тільки тоді, якщо в інших каналах, підведених до даної станції вода вже є. Вода в нашому випадку буде означати наповнення змінних арифметичним змістом – значенням. Що дано в задачі? –  $a$  та  $b$ . Нехай в пунктах  $a$  та  $b$  “забили чисті джерела”. По окремим каналам вода поступить до п'яти розподільчих станцій, але тільки три з них зможуть відкрити шлюзи для заповнення колодязів  $c$ ,  $\alpha$ ,  $S$ . Далі вода з цих колодязів поступить в інші канали, і шлюзи на другому кроці вже зможуть відкрити розподільні станції, які наповнюють колодязі  $p$ ,  $R$ ,  $\beta$ . На третьому кроці шлюзи відкриє розподільча станція, яка наповнює колодязь  $r$ . Таким чином, наведена сукупність формул *достатня*, щоб вирішити поставлену проблему. Виходячи з відомих значень  $a$  та  $b$ , ми зможемо знайти значення  $r$ .

Ця мережа це фрагмент своєрідної бази знань. Для опису колодязів в *Delphi* зручно використовувати записи слідуючого типу:

```

type
  tmytype = record
    value : double;
    init  : boolean;
  end;

```

Поле *init* вказує на те, чи б'є в даному колодязі джерело чистої води. Для того щоб побудувати схему розв'язку будемо використовувати масив стрічок:



Национальний університет водного господарства та природокористування

```

tmyarray = array [1..8] of string;
var trace : tmyarray; i, j : integer;

```

Розподільчі станції ми будемо описувати процедурами. Наприклад для теореми Піфагора маємо

```

procedure Pifag (var a,b,c : tmytype);
begin
  if (a.init = true) and (b.init=true) then
  begin
    if c.init = false then
    begin
      j:=j+1; trace[j]:= 'Pifagor';
    end;
    c.init:=true; c.value:=sqrt(sqr(a.value)+sqr(b.value));
  end;
  if (a.init=true) and (c.init=true) then
  begin
    if b.init = false then
    begin
      j:=j+1; trace[j]:= 'Pifagor';
    end;
    b.init:=true; b.value:=sqrt(sqr(c.value)-sqr(a.value));
  end;
  if (b.init=true) and (c.init=true) then
  begin
    if a.init = false then
    begin
      j:=j+1; trace[j]:= 'Pifagor';
    end;
    a.init:=true; a.value:=sqrt(sqr(c.value)-sqr(b.value));
  end;
end;

```



Створивши такі процедури для всіх формул і помістивши їх в цикл ми отримаємо автоматизовану програму для розв'язку всіх задач на прямокутний трикутник.

## Завдання для виконання лабораторної роботи:

В середовищі програмування *Delphi* створити втоматизовану систему для розв'язку всіх задач відповідно до тематики:

- 1) ромб; 2) трикутник у якого один з кутів дорівнює  $30^{\circ}$ ; 3) прямокутник;
- 4) рівнобедрений трикутник; 5) трикутник у якого одна із сторін рівна 10;
- 6) трикутник у якому рівні сторона і висота, яка на неї опущена; 7) трикутник у якому рівні три медіани; 8) трикутник у якому рівні три висоти; 9) трикутник у якому рівні три бісектриси; 10) трикутник у якому дві медіани рівні 10; 11) трикутник у якому дві висоти рівні 10; 12) трикутник у якому дві бісектриси рівні 10; 13) трикутник у якому рівні сторона і медіана, яка на неї опущена; 14) трикутник у якому рівні сторона і бісектриса, яка на неї опущена; 15) паралелограм у якого один з кутів рівний  $30^{\circ}$ .

## Лабораторна робота №4

**Тема: Розробка інтелектуальних систем для розв'язку задач з елементарної геометрії на основі загально відомих правил та відомих фактів.**

### **Мета роботи:**

- Ознайомитися з типовими задачами елементарної геометрії.
- Ознайомитися з принципом за яким комп'ютер із загальних правил та початкових фактів отримує нові факти.

### **Теретичні відомості з елементами прикладу:**

Припустимо, в нас є набір деяких фактів, пов'язаних з відношенням більше. Наприклад, про чотири змінні  $a$ ,  $b$ ,  $c$ ,  $d$  відомо, що їх значення задовольняють нерівностям  $a > b$ ,  $c > d$ ,  $b > c$ . Чи можна опираючись на ці факти, сказати, що більше:  $a$  чи  $d$  ?

Розв'язуючи цю задачу, ми скоріше за все, випишемо дані в умові задачі нерівності в ланцюг:  $a > b > c > d$ , звідки слідує, що  $a > d$ . Як до такого висновку міг би прийти комп'ютер ?

Хоча ми цього не підкреслювали, але розв'язувати задачу нам допомогло загальне правило:

Якщо  $X > Y$  і  $Y > Z$ , то  $X > Z$ .

Тут  $X, Y, Z$  – довільні змінні, значення яких допускається оцінювати за допомогою відношення  $>$ . „Попросимо” комп’ютер підставляти замість  $X, Y, Z$  конкретні об’єкти,  $a, b, c, d$ . Ось що ми отримаємо:

$a > c$  ( підстановка  $X = a, Y = b, Z = c$ );

$b > d$  ( підстановка  $X = b, Y = c, Z = d$ );

$a > d$  ( підстановка  $X = a, Y = c, Z = d$ ).

Вміючи тільки підставляти, комп’ютер із загального правила та трьох початкових фактів (нерівностей) може отримати три нові факти, серед яких виявиться і той, який нас цікавить.

С таким же успіхом комп’ютер може справитись з іншими загальними правилами, які описують властивості різних відношень: рівності, паралельності, подібності, ділимості та багатьох інших.

Розглянемо, наприклад, наступну задачу з елементарної геометрії. Прямі 3 та 2 дотикаються кола з центром в точці  $O$  відповідно в точках  $A$  та  $B$ . (рис. 6.8)

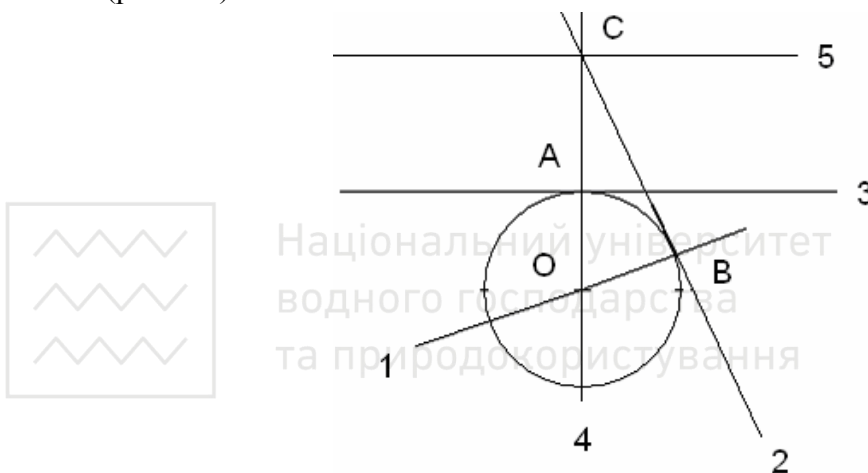


Рис. 6.8 Ілюстрація до задачі

Прямі 4 та 1 це продовження радіусів  $OA$  та  $OB$ , які проведені у відповідні точки дотику. Через точку  $C$ , яка є точкою перетину прямих 4 та 2, проводиться пряма 5 так, що  $5 \parallel 3$ . Відомо кут  $\angle(1,4)$  між прямими 1 та 4 (додатній напрямком - проти годинкової стрілки). Найдіть кут між прямими 2 і 5, тобто  $\angle(2,5)$ .

Випишемо основні факти, які випливають з умови задачі і властивості дотичних до кола:

- $5 \parallel 3$ ;
- $1 \perp 2$ ;

- $4 \perp 3$ ;

В програмі ці факти можна описати наступним чином. Лінію зручно описувати змінною типу записи:

```
TMyLine = record
  name : char;
  prl : string;
  prp : string;
end;
```

в стрічці *prl* буде міститися назви тих прямих, які паралельні до даної, а в *prp* ті, які перпендикулярні. Для кута зручно вести наступний:

```
tmystring = string[2];
TMyAngle = record
  value : char;
  init : boolean;
  name : tmystring
end;
```

Всі прямі на рисунку утворюють 20 кутів для їх нумерації введемо відповідний тип:

```
TNzykt = (an12,an13,an14,an15,an21,an23,
an24,an25,an31,an32,an34,an35,an41,an42,
an43,an45,an51,an52,an53,an54);
```

Всі кути та прямі зручно помістити в масиви:

```
TMyArray1 = array [1..5] of TMyLine;
TMyarray2 = array [an12..an54] of TMyAngle;
```

Тоді наші основні факти можна описати слідуєчим

чином:

```
Var a : TMyArray1; b : TMyArray2;
begin
  a[5].prl:='3';
  a[1].prp:='2';
  a[4].prp:='3';
  for i:=1 to 5 do
    a[i].name:=IntToStr(i)[1];
    b[an14].init:=true; b[an14].value:='a';
    b[an14].name:='14';
    for i:=1 to 5 do
      begin
        if (i<>5) then a[i].prl:='';
        if (i<>1) and (i<>4) then a[i].prp:='';
      end;
    for ngl:=an12 to an54 do
      if ngl<>an14 then
        begin
```



```
case ngl of
an12 : b[ngl].name:='12';
an13 : b[ngl].name:='13';
an15 : b[ngl].name:='15';
an21 : b[ngl].name:='21';
an23 : b[ngl].name:='23';
an24 : b[ngl].name:='24';
an25 : b[ngl].name:='25';
an31 : b[ngl].name:='31';
an32 : b[ngl].name:='32';
an34 : b[ngl].name:='34';
an35 : b[ngl].name:='35';
an41 : b[ngl].name:='41';
an42 : b[ngl].name:='42';
an43 : b[ngl].name:='43';
an45 : b[ngl].name:='45';
an51 : b[ngl].name:='51';
an52 : b[ngl].name:='52';
an53 : b[ngl].name:='53';
an54 : b[ngl].name:='54';
end;
b[ngl].init:=false;
end;
```

Для виводу нових фактів комп'ютер може скористатися наступними правилами:

- якщо  $X \parallel Y$ , то  $Y \parallel X$ ;
- якщо  $X \perp Y$ , то  $Y \perp X$ ;
- якщо  $X \parallel Y$  та  $Z \perp Y$ , то  $Z \perp X$ ;
- якщо  $X \perp Y$  та  $Z \perp U$ , то  $\angle(X, Z) = \angle(Y, U)$  і  $\angle(U, X) = \angle(Z, Y)$ .

Ці правила можна описати за допомогою наступних процедур:

```
procedure rule1(var X, Y : TMyLine);
begin
if Pos(Y.name, X.prl) > 0 then
if Pos(X.name, Y.prl) = 0 then Y.prl := Y.prl + X.name;
if Pos(X.name, Y.prl) > 0 then
if Pos(Y.name, X.prl) = 0 then X.prl := X.prl + Y.name;
end;
procedure rule2(var X, Y : TMyLine);
begin
if Pos(Y.name, X.prp) > 0 then
if Pos(X.name, Y.prp) = 0 then Y.prp := Y.prp + X.name;
if Pos(X.name, Y.prp) > 0 then
```



```
if Pos(Y.name,X.prp)=0 then X.prp:=X.prp+Y.name;
end;
procedure rule3(var X,Y,Z : TMyLine);
begin
if (Pos(Y.name,X.prl)>0) and (Pos(Y.name,Z.prp)>0) then
if Pos(X.name,Z.prp)=0 then Z.prp:=Z.prp+X.name;
end;
procedure rule4(var X,Y,Z,U : TMyLine);
var s,s1 : tmystring; ngl,ngls,ngls1 : TNzvkt;
begin
If (Pos(Y.name,X.prp)>0) and (Pos(U.name,Z.prp)>0) then
begin
s:=X.name; s:=s+Z.name;
s1:=Y.name; s1:=s1+U.name;
for ngl:=an12 to an54 do
begin
if b[ngl].name=s then ngls:=ngl; // тут важливо,
//щоб X,Y,Z,U –
if b[ngl].name=s1 then ngls1:=ngl; // були різними прямими
end;
if b[ngls].init and not b[ngls1].init then
begin
b[ngls].value:=b[ngls1].value;
b[ngls].init:=true;
end;
if b[ngls].init and not b[ngls1].init then
begin
b[ngls1].value:=b[ngls].value;
b[ngls1].init:=true;
end;
s:=U.name; s:=s+X.name;
s1:=Z.name; s1:=s1+Y.name;
for ngl:=an12 to an54 do
begin
if b[ngl].name=s then ngls:=ngl; // тут важливо,
//щоб X,Y,Z,U –
if b[ngl].name=s1 then ngls1:=ngl; // були різними прямими
end;
if b[ngls1].init and not b[ngls].init then
begin
b[ngls].value:=b[ngls1].value;
b[ngls].init:=true;
end;
end;
```



```

if b[ngls].init and not b[ngls1].init then
begin
b[ngls1].value:=b[ngls].value;
b[ngls1].init:=true;
end;
end;
end;

```

По аналогії з автоматизованою системою для розв'язку задач потрібно помістити ці всі процедури в цикл та задати достатню кількість ітерацій, та забезпечити те, щоб процедури застосовувались до різних прямих.

### Завдання для виконання лабораторної роботи:

Розробити систему логічного виводу і за її допомогою розв'язати наступні задачі:

- Для рис.6.9 вивести список кутів, які рівні

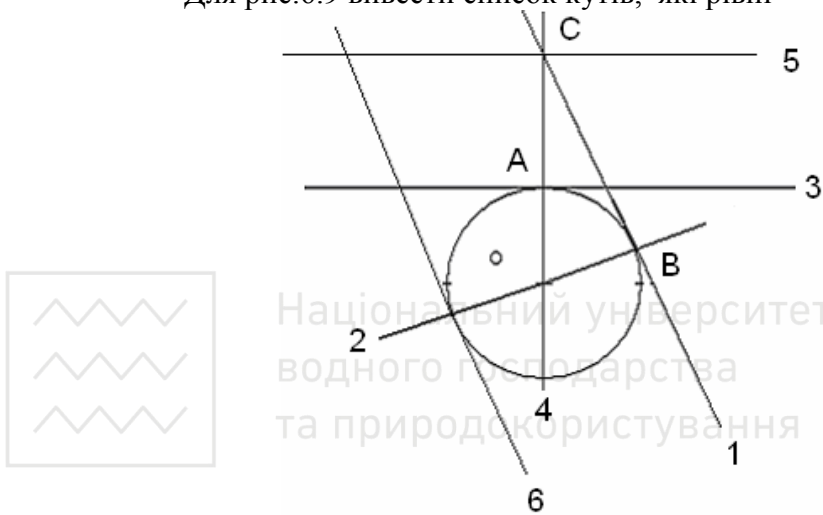


Рис. 6. 9

1)  $\angle(2,4)$ ; 2)  $\angle(1,5)$ ; 3)  $\angle(1,3)$ ; 4)  $\angle(3,2)$ ; 5)  $\angle(4,1)$ ; 6)  $\angle(4,6)$ ; 7)  $\angle(6,5)$ ;

Для рис. 6.10 вивести список кутів, які рівні



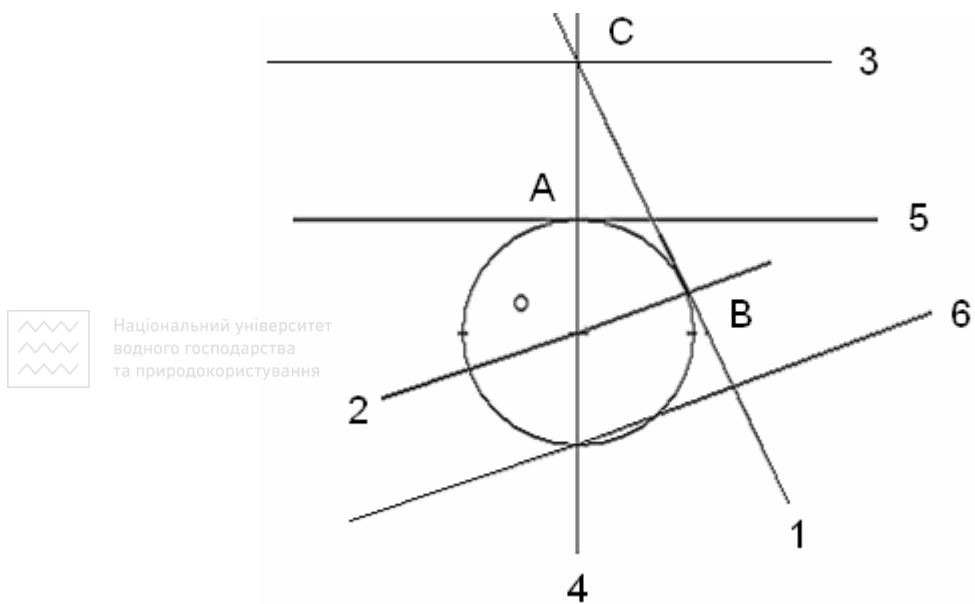


Рис. 6.10.

8)  $\angle(6,4)$ ; 9)  $\angle(2,4)$ ; 10)  $\angle(3,1)$ ; 11)  $\angle(5,2)$  12)  $\angle(1,5)$  13)  $\angle(2,4)$  14)  $\angle(1,4)$ ;



Національний університет водного господарства та природокористування

## Навчальне видання

*Власюк Анатолій Павлович  
Мартинюк Петро Миколайович  
Прищеп Оксана Володимирівна  
Філатова Інна Анатоліївна  
Філатов Михайло Сергійович  
Роценюк Алла Михайлівна  
Демчук Олена Станіславівна  
Демчук Микола Борисович  
Мічута Ольга Романівна  
Цветкова Тетяна Павлівна  
Федорчук Наталія Анатоліївна*



Національний університет  
водного господарства  
та природокористування

## Лабораторний практикум з програмування Навчальний посібник



*Художнє оформлення обкладинки: Бодов Олександр Степанович*

Підписано до друку 30.10.2009р. Формат 70x100  $\frac{1}{16}$ .

Папір друкарський №1. Гарнітура Times. Друк різнографічний.

Ум.-друк. аркуш. 28,8. Обл.-вид. арк. 27,0.

Тираж 120 прим. Зам.№ 1817.

*Редакційно-видавничий центр  
Національний університет водного  
господарства та природокористування  
33028, Рівне, вул. Соборна, 11*

*Свідоцтво про внесення суб'єкта видавничої справи до державного  
реєстру видавців, виготовників і розповсюджувачів видавничої продукції  
РВ №31 від 26.04.2005р.*