

Основи програмування

Шпортко О. В.

Лабораторний практикум
для студентів денної та заочної форм навчання

Освітньої програми	Інженерія програмного забезпечення Інтернету речей
Рівня вищої освіти	першого (бакалаврського)
За спеціальністю	121 Інженерія програмного забезпечення
Галузі знань	12 Інформаційні технології

Рівне 2022

Міністерство освіти і науки України
ПВНЗ «Міжнародний економіко-гуманітарний університет
імені академіка Степана Дем'янчука»
Факультет кібернетики
Кафедра інформаційних систем та обчислювальних методів

Основи програмування

Лабораторний практикум
для студентів денної та заочної форм навчання

Освітньої програми	Інженерія програмного забезпечення Інтернету речей
Рівня вищої освіти	першого (бакалаврського)
За спеціальністю	121 Інженерія програмного забезпечення
Галузі знань	12 Інформаційні технології

З питань тиражування та використання цього видання звертайтеся на email books_shpoutko@ukr.net

Рівне 2022

УДК 004.422/004.424

ББК 32.973-01

Ш 84

Друкується за рішенням Навчально-методичної комісії ПВНЗ "Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука" (протокол № 8 від 18.05.2022 р.).

Основи програмування. Лабораторний практикум для студентів денної та заочної форм навчання освітньої програми «Інженерія програмного забезпечення Інтернету речей» першого (бакалаврського) рівня вищої освіти за спеціальністю 121 Інженерія програмного забезпечення галузі знань 12 Інформаційні технології / уклад. О. В. Шпортько. Рівне: ПВНЗ "МЄГУ ім. акад. С. Дем'янчука", 2022. 104 с.

Укладач: О. В. Шпортько, кандидат технічних наук, доцент, доцент кафедри інформаційних систем та обчислювальних методів ПВНЗ "Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука".

Рецензенти: А. Я. Бомба, доктор технічних наук, професор, професор кафедри комп'ютерних наук та прикладної математики Національного університету водного господарства та природокористування;

П. С. Янчук, кандидат фізико-математичних наук, доцент, професор кафедри інформаційних систем та обчислювальних методів ПВНЗ "Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука".

Відповідальний за випуск: Ю. Г. Лотюк, кандидат педагогічних наук, доцент, завідувач кафедри інформаційних систем та обчислювальних методів ПВНЗ "Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука".

Лабораторний практикум розроблений у відповідності з практичною частиною робочої програми дисципліни «Основи програмування» для студентів денної та заочної форм навчання освітньої програми «Інженерія програмного забезпечення Інтернету речей» першого (бакалаврського) рівня вищої освіти за спеціальністю 121 Інженерія програмного забезпечення. Містять впорядкований набір завдань і вказівок до виконання лабораторних робіт та самостійного опрацювання матеріалу дисципліни. Основна увага приділяється вивченню процесу алгоритмізації завдань та формуванню вмій і навиків програмування розроблених алгоритмів засобами мови програмування C#. Призначений для студентів, викладачів та всіх, хто прагне оволодіти основами програмування для ефективної обробки даних.

Затверджені кафедрою інформаційних систем та обчислювальних методів ПВНЗ "МЄГУ ім. акад. С. Дем'янчука" (протокол № 9 від 10.05.2022 р.).

Схвалені навчально-методичною комісією факультету кібернетики ПВНЗ "МЄГУ ім. акад. С. Дем'янчука" (протокол № 9 від 10.05.2022 р.).

ББК 32.973-01

Зміст

Передмова	4
1. Створення програм лінійної структури з використання унарних та бінарних операцій.	6
2. Використання арифметичних функцій.....	8
3. Створення програм лінійної структури з вводом/виводом у вікні консольного додатку.....	12
4. Організація вводу/виводу з використанням діалогових вікон.	17
5. Використання повного та неповного операторів розгалуження.....	21
6. Застосування операторів розгалуження для розв'язування прикладних задач.	25
7. Обробка виключних ситуацій. Засоби відлагодження розроблених програм.....	30
8. Оператор вибору. Створення найпростіших тестових програм.....	33
9. Оператори циклу з перед- та післяумовою.....	36
10. Програмування алгоритмів з використанням оператора циклу з параметром.	42
11. Застосування перелічувального циклу. Вкладені цикли та оператори розгалуження.	46
12. Програмування алгоритмів обробки одновимірних масивів.	51
13. Динамічні масиви.....	55
14. Сортування одновимірних масивів.....	59
15. Двовимірні масиви. Обробка багатовимірних масивів.....	64
16. Програмування алгоритмів обробки текстових рядків та слів.....	70
17. Структури. Алгоритми обробки структурованих даних.....	74
18. Створення методів користувача. Рекурсивні виклики методів.....	78
19. Розробка та використання власних бібліотек методів класів.....	85
20. Обробка файлів як цілісних об'єктів з використанням масивів рядків.....	88
21. Обробка текстових файлів та потоків.	90
22. Опрацювання двійкових файлів та потоків послідовного доступу.	94
23. Обробка двійкових файлів та потоків з структурованими даними.....	98
Рекомендована література	101

Передмова

На сьогоднішній день важко уявити собі діяльність сучасного суспільства без комп'ютерної техніки, локальних та глобальних мереж і відповідного програмного забезпечення. З комп'ютерами ми стикаємося всюди: у банку та відділі бухгалтерії, в пенсійному фонді та податковій інспекції, у паспортному столі та на залізничному вокзалі, в ЖЕКу та Укртелекомі, в редакціях газет та друкарнях, в рекламних агенціях та конструкторських бюро. Високі темпи накопичення інформації, вимоги мобільності, оперативності поширення та надійності зберігання даних, швидкості та достовірності їх обробки спонукають розвиток апаратних засобів, системних та прикладних оболонок і пакетів. На сьогоднішній день 97 % населення світу користується мобільними телефонами, а кількість комп'ютерів у світі перевищує 1361 мільйонів і продовжує зростати. При цьому приблизно один раз в півтора року подвоюються показники основних технічних характеристик апаратних засобів, один раз в два-три роки змінюється покоління програмного забезпечення і один раз на п'ять-сім років змінюється база стандартів, інтерфейсів та протоколів. Разом з тим, темп кількісного зростання інформаційних систем значно перевищує темп підготовки спеціалістів, здатних ефективно працювати з ними.

В цих умовах кожен випускник ВНЗ з галузі інформаційних технологій, що має використовувати комп'ютерну техніку в своїй професійній діяльності для розробки, експлуатації та супроводження інформаційних систем, повинен не лише володіти елементарними навиками використання програмного забезпечення (на сьогодні це повинна вміти кожна освічена людина), а й розуміти внутрішню організацію системного та прикладного забезпечення, розробляти алгоритми розв'язку задач, вірно обирати мову програмування залежно від характеристик окремого завдання, швидко розв'язувати алгоритмічні задачі засобами обраної мови програмування, кваліфіковано та продуктивно працювати з сучасними оболонками мов програмування, вміти відлагоджувати створені програми та переконуватися в їх дієздатності, розробляти сучасний інтерфейс для створеного програмного коду та здійснювати його адаптацію до потреб користувача, оцінювати складність алгоритмізації та програмування поставленого завдання. Формуванню саме таких фахівців має сприяти вся дисципліна та даний лабораторний практикум зокрема. Після опанування матеріалу практикуму студент повинен вміти швидко та якісно розробляти елементарне програмне забезпечення для прикладних завдань з використанням засобів сучасних операційних систем та оболонок мов програмування.

Даний лабораторний практикум містить впорядкований набір завдань і вказівок до виконання лабораторних робіт та самостійного опрацювання матеріалу дисципліни. Виконання завдань кожної лабораторної роботи вимагає використання знань, умінь та навиків, здобутих під час виконання попередніх робіт, що, з одного боку, дозволяє автоматично повторювати вивчений матеріал, а з іншого – сприяє системності його викладу.

Перед завданнями кожної лабораторної роботи з метою актуалізації знань, отриманих під час лекційних занять, наведено відповідні контрольні запитання. Всі лабораторні роботи виконуються студентами поваріантно (номер варіанту студента співпадає з його порядковим номером у списку підгрупи). Відладження програм та перевірка різних тестових випадків виносяться на самостійне опрацювання. За результатами виконання кожної лабораторної роботи студент оформляє письмовий звіт, що складається з наступних розділів:

- теми та мети лабораторної роботи;
- відповідей на контрольні запитання лабораторної роботи;
- формулювання, математичного розв'язку (при необхідності), тексту програми та результатів тестування (запропонованих у завданні та розроблених самостійно вхідних даних і відповідних результатів виконання програми) кожного завдання лабораторної роботи;
- висновків стосовно знань, умінь та навичок, здобутих при виконанні лабораторної роботи.

З питань тиражування та використання цього видання звертайтеся на e-mail books_shp@ukr.net

Лабораторна робота № 1

Тема. Створення програм лінійної структури з використання унарних та бінарних операцій.

Мета. Формування вмінь і навиків роботи в середовищі мови програмування C#. Систематизація та застосування знань структури лінійних програм. Формування вмінь і навиків програмування алгоритмів лінійної структури.

Контрольні запитання

1. Яка структура консольного додатку в мові C#? На які основні частини поділяється програма консольного додатку?
2. Що записується у кожному розділі декларативної частини програми?
3. На які групи поділяються слова оболонки мови програмування?
4. Чим константи відрізняються від змінних? Де і як вони описуються?
5. Які знаки арифметичних операцій використовуються у виразах?
6. У якій послідовності виконуються рядки програми?
7. Як записуються і виконуються коментарі мовою C#?
8. Як вивести результати обчислень у консольному додатку?

Завдання

1. Скласти три лінійних програми для обчислення значення виразу варіантів $n, n+1, n+2$, де n – ваш номер у списку журналу групи.

Вимоги до програми:

- використати константи скрізь, де це можливо;
- числове значення виразу обчислити за допомогою оператора присвоєння;
- значення вхідних даних та остаточні результати виконання програми надрукувати у вікні консольного додатку.

Варіанти

1. $\frac{z+y}{2z-x}$ при $x=0.137$, $y=1.27$, $z=4.7561$;
2. $\frac{x(y+z)}{y-yz}$ при $x=0.0399$, $y=4.83$, $z=0.072$;
3. $\frac{x+y}{4x-xy}$ при $x=1.576$, $y=1.786$, $z=1.1236$;
4. $\frac{y-4z}{x+4y}$ при $x=12.743$, $y=0.654$, $z=0.0208$;
5. $\frac{z-y}{14x+yz}$ при $x=3.49$, $y=0.456$, $z=0.0059$;
6. $\frac{xz+3y}{(y-z)x}$ при $x=0.0976$, $y=2.371$, $z=1.1587$;
7. $\frac{(x-y)}{y+zx}$ при $x=82.356$, $y=34.42$, $z=7.0046$;

8. $\frac{y+z}{4y+2x}$ при $x=0.11578$, $y=4.675$, $z=4.654$;
9. $\frac{x-y}{xy+z}$ при $x=3.7156$, $y=3.034$, $z=0.756$;
10. $\frac{x+4y}{xy-z(x-y)}$ при $x=7.654$, $y=0.876$, $z=0.0987$;
11. $\frac{y+z}{(z-x)y}$ при $x=0.036$, $y=3.987$, $z=4.654$;
12. $\frac{2(x+y)}{z+y(x+y)}$ при $x=0.327$, $y=0.0098$, $z=4.675$;
13. $\frac{x^2-1-(z-2)}{y+1+(x+1.4)}$ при $x=3.4$, $y=25$, $z=0.35$;
14. $\frac{(2y)}{z/(x-2)}$ при $x=0.83$, $y=4.2$, $z=-6.7$;
15. $\frac{xy}{zx+0.4}$ при $x=6.8$, $y=0.75$, $z=12$;
16. $\frac{zy+x}{2z-x}$ при $x=0.137$, $y=1.27$, $z=4.7561$;
17. $\frac{x(2y+3z)}{y-xz}$ при $x=0.0399$, $y=4.83$, $z=0.072$;
18. $\frac{7x+8y}{x-4zy}$ при $x=1.576$, $y=1.786$, $z=1.1236$.

Наприклад, програма для розв'язання завдання варіанту № 10 може бути такою:

```
static void Main()
{
    const double x = 7.654, y = 0.876, z = 0.0987;
    double res=(x+4*y)/(x*y-z*(x-y));
    Console.WriteLine("При x={0}, y={1}, z={2} значення виразу рівне {3}", x, y, z, res);
    ConsoleKeyInfo s=Console.ReadKey();
}
```

3 питань

Лабораторна робота № 2

- Тема.** Використання арифметичних функцій.
Мета. Формування вмінь і навиків програмування обчислення значень виразів з використання арифметичних функцій. Систематизація та застосування знань структури лінійних програм. Закріплення вмінь і навиків програмування алгоритмів лінійної структури.

Контрольні запитання

1. Що таке змінна? Як позначаються змінні в мові С#?
2. Як записуються числа та текстові константи в С#?
3. Чим операції відрізняються від операторів? Оператори від функцій?
4. Як визначається порядок дій в арифметичних виразах, записаних мовою С#?
5. У якій послідовності виконуються рівнозначні операції?
6. Як змінити порядок виконання операцій?
7. Який рядок програми називається логічним, а який – фізичним? Яка між ними різниця?

Завдання

1. Скласти лінійну програму для обчислення значення виразів.

Вимоги до програми:

- сталі значення задати за допомогою констант;
- числові значення змінних обчислити за допомогою команди присвоєння;
- значення вхідних даних та остаточний результат виконання програми надрукувати в стовпець у вікні консольного додатку.

Варіанти

1. $\frac{z + \sqrt{\cos y}}{2z - x}$ при $x=0.437$, $y=1.27$, $z=4.7561$;
2. $\frac{x \ln(y + z)}{y - \arcsin yz}$ при $x=0.0399$, $y=4.83$, $z=0.072$;
3. $\frac{x + y}{4x - z \cos y}$ при $x=1.576$, $y=1.786$, $z=1.1236$;
4. $\frac{4x - 4z}{\ln x + 4y + \sqrt{z}}$ при $x=12.743$, $y=0.654$, $z=0.0208$;
5. $\frac{z - tgy}{14x + \sqrt{y \sin z}}$ при $x=3.49$, $y=0.456$, $z=0.0059$;
6. $\frac{xz + 3y}{(y - z) \cos x}$ при $x=0.0976$, $y=2.371$, $z=1.1587$;
7. $\frac{\ln(x - y)}{y + z \sin x}$ при $x=82.356$, $y=34.42$, $z=7.0046$;

8. $\frac{y + \cos z}{4y + 2x\sqrt{x+2}}$ при $x=0.11578$, $y=4.675$, $z=4.654$;
9. $\frac{x-y}{xy+z}$ при $x=3.7156$, $y=3.034$, $z=0.756$;
10. $\frac{\ln x + 4y}{xy - z \sin(x-y)}$ при $x=7.654$, $y=0.876$, $z=0.0987$;
11. $\frac{y + \ln z}{(z-x) \cos y}$ при $x=0.036$, $y=3.987$, $z=4.654$;
12. $\frac{2tg(x+y)}{\sqrt{z+y}\sqrt{x+y}}$ при $x=0.327$, $y=0.0098$, $z=4.675$;
13. $\frac{x^2 - 1 - |z-2|}{y+1 + \cos(x+1.4)}$ при $x=3.4$, $y=25$, $z=0.35$;
14. $\frac{(2y)^2}{\sqrt{z/(x-2)}}$ при $x=0.83$, $y=4.2$, $z=-6.7$;
15. $\frac{\ln xy}{z \sin x + 0.4}$ при $x=6.8$, $y=0.75$, $z=12$.

Наприклад, програма для розв'язання завдання варіанту № 8 може бути такою:

```
static void Main(string[] args)
{
    const double x = 0.11578, y = 4.675, z = 4.654;
    double res=(y+Math.Cos(z))/(4*y+2*x*Math.Sqrt(x+2));
    Console.WriteLine("При x={0,18:F3}, y={1,5:N2}, z={2,7:N3}"+
        "значення виразу рівне {3,8:F4}", x, y, z, res);
    Console.ReadKey();
}
```

2. Скласти лінійну програму для обчислення проміжних та кінцевого значень виразів.

Вимоги до програми:

- сталі значення задати за допомогою констант;
- числові значення змінних обчислити за допомогою команди присвоєння;
- значення вхідних даних, проміжні та остаточний результат виконання програми надрукувати в стовпець у вікні консольного додатку.

Варіанти

1. $Y = \frac{x^2 - z^2}{\ln|x-z|}$, де $x = \frac{\sin^2 a^3}{\ln|a+c|}$,
- $z = \sqrt{\frac{a+c}{ac}} + \pi$, $a=3.5$, $c=-2.16$;

$$2. Z = \ln \left| \frac{x\sqrt{x} + \cos^3 y^2}{1.604} \right|^2, \text{ де } x = \ln c + \frac{(a-c)^2}{6} e^{-a},$$

$$y = \sqrt[3]{\cos a^2 + ac + 0.06}, a=0.2, c=7;$$

$$3. Y = -\sqrt{|\ln x - \ln z| + 1.31}, \text{ де } x = \frac{e^{-2.5a} + \sin^2 a^3}{2 \ln |ca|},$$

$$z = \frac{\cos(c-a) + \sqrt[3]{ac}}{1 + \ln |ca|}, a=0.6, c=3.12;$$

$$4. Z = ce^{-2.5x+y^2} - \sqrt[3]{cx}, \text{ де } x = \frac{\ln |c+d|}{\cos \frac{\pi}{d}} + 0.17,$$

$$Y = \frac{\sin \frac{3d^3}{2} - \cos \frac{c}{4}}{\ln |d| + \ln \sqrt{c^2}}, c=4.5, d=2.21;$$

$$5. Z = \frac{|x-1| + e^{-y}}{12.35 - \ln \sqrt{|x|}}, \text{ де } y = 2a\sqrt[3]{a+b},$$

$$x = \frac{e^a + e^{\frac{1}{b}}}{\sin \sqrt{a+6.3}}, a=1.75, b=0.4;$$

$$6. P = \frac{2^x e^{-xy} + 17.4}{\sqrt{\sin^3 xy^2}}, \text{ де } x = (a^2 + c^2)^{-4.1},$$

$$y = \operatorname{arctg}^3 \frac{1}{c}, a=-2.005, c=0.87;$$

$$7. R = \operatorname{ctg} \frac{x+y}{(x-y)^2} + 1.3, \text{ де } x = \sin^4 e^{-b} + |ab|,$$

$$y = \ln |a-b| + \ln \frac{\pi}{a}, a=1.77, b=0.62$$

$$8. Z = \cos \frac{x^2}{0.14} + \ln \frac{a}{5}, \text{ де } x = \sqrt{(k+6.1)^3},$$

$$y = \ln^4 k + \frac{m^{-6}}{m+5.3}, k=14, m=0.42;$$

$$9. A = \frac{e^{-3.5|x|+\sqrt{\pi}}}{\arctg^3(y-1)}, \text{ де } x = a + \cos \frac{\pi}{b},$$

$$y = \ln \left| \frac{\pi}{16} - b \right|, a = 0.5, b = 1400;$$

$$10. T = \ln |m - y| + \cos^3 my, \text{ де } m = \sqrt{|x + a|} + 17.14 * \ln \frac{\pi}{3},$$

$$y = a^3 \sqrt{\sin^4 x^3 + 12.47}, x = 3.4, a = -1, 17;$$

$$11. L = e^2 \ln x^4 - \sqrt{|y + 1|}, \text{ де } x = 21.4(a - 0.5)^2 + \cos \frac{\pi}{b},$$

$$y = \ln \left| \frac{\pi}{a} - b \right| + tg^2 b^3, a = 0.7, b = -4;$$

$$12. F = \frac{x^e - e^x + 0.12}{\sqrt{|\sin(y-1)|}}, \text{ де } x = e^{-\pi} + \pi^{-e},$$

$$y = \ln a^3 - \arctga, a = 6.45;$$

$$13. R = \arctg \frac{x+1}{y-2} + \ln |k + x|, \text{ де } x = \sqrt{|m + n|} + 17.14mn,$$

$$y = \sqrt[3]{|km - 3|} + \frac{\pi}{6}, m = 3, n = -2.2, k = 0.801;$$

$$14. N = \arctg(\sin^2 x + tg^2 y), \text{ де } x = \ln |a + 2.3| - \ln |b - 3.2|,$$

$$y = \sin^2(a - b)^3, a = 15.3, b = -0.012;$$

$$15. Z = 1.4^{\ln m} * \frac{\sqrt{m^3 + 2.5x}}{x}, \text{ де } x = \cos^2 \frac{\pi}{y} + 29.45,$$

$$y = (3m)^e, m = 13.44.$$

Наприклад, програма для розв'язання завдання варіанту № 12 може мати такий вигляд:

```
static void Main(string[] args)
{const double a=6.45;
double x, y, F;
x=Math.Log(Math.Pow(a,3))-Math.Atan(a);
y=Math.Exp(-Math.PI)+Math.Pow(Math.PI,-Math.E);
F=(Math.Pow(x,Math.E)-Math.Exp(x)+0.12)/(Math.Sqrt(Math.Abs(Math.Sin(y-1))));
Console.WriteLine("При a={0} обчислено x={1,8:N3}, y={2,7:F2}, F={3,8:F4}", a,
Console.ReadKey();
}
```

Лабораторна робота № 3

Тема. Створення програм лінійної структури з вводом/виводом у вікні консольного додатку.

Мета. Формування вмінь і навиків використання підпрограм вводу/виводу з використанням командного рядка. Застосування знань структури лінійних програм. Закріплення вмінь і навиків роботи в середовищі мови програмування C# та програмування алгоритмів лінійної структури.

Контрольні запитання

1. Як записуються мовою C# ключові слова початку і кінця програми?
2. Як вказується в мові C# тип змінної?
3. Чи можна записувати в одному логічному рядку декілька операторів мовою C#?
4. Як записується і виконується команда присвоєння мовою C#?
5. Як записуються і виконуються функції вводу з командного рядка?
6. Як записуються і виконуються процедури виводу у вікно консольного додатку?
7. Що таке формат виводу? Для ідентифікаторів яких типів і як він застосовується?

Завдання

1. Скласти лінійну програму для обчислення значення виразів.

Вимоги до програми:

- значення змінних, позначених ?, ввести з командного рядка;
- числові значення змінних, заданих формулами, обчислити за допомогою команди присвоєння;
- значення вхідних даних та остаточні результати виконання програми вивести в стовпець у вікні консольного додатку.

Варіанти

- | | | | | |
|----|---|--------|--------|--------|
| 1. | $\frac{z + \sqrt{\cos y}}{2z - x}$ | x - ?, | y - ?, | z - ?; |
| 2. | $\frac{x \ln(y + z)}{y - \arcsin yz}$ | x - ?, | y - ?, | z - ?; |
| 3. | $\frac{x + y}{4x - z \cos y}$ | x - ?, | y - ?, | z - ?; |
| 4. | $\frac{xy - 4z}{\ln x + 4y + \sqrt{z}}$ | x - ?, | y - ?, | z - ?; |
| 5. | $\frac{z - tgy}{14x + \sqrt{y \sin z}}$ | x - ?, | y - ?, | z - ?; |

- | | | | | |
|-----|---|-----------|-----------|-----------|
| 6. | $\frac{xz + 3y}{(y - z) \cos x}$ | $x - ?$, | $y - ?$, | $z - ?$; |
| 7. | $\frac{\ln(x - y)}{y + z \sin x}$ | $x - ?$, | $y - ?$, | $z - ?$; |
| 8. | $\frac{y + \cos z}{4y + 2x\sqrt{x + 2}}$ | $x - ?$, | $y - ?$, | $z - ?$; |
| 9. | $\frac{x - y}{xy + z}$ | $x - ?$, | $y - ?$, | $z - ?$; |
| 10. | $\frac{\ln x + 4y}{xy - z \sin(x - y)}$ | $x - ?$, | $y - ?$, | $z - ?$; |
| 11. | $\frac{y + \ln z}{(z - x) \cos y}$ | $x - ?$, | $y - ?$, | $z - ?$; |
| 12. | $\frac{2tg(x + y)}{\sqrt{z} + y\sqrt{x + y}}$ | $x - ?$, | $y - ?$, | $z - ?$; |
| 13. | $\frac{x^2 - 1 - z - 2 }{y + 1 + \cos(x + 1.4)}$ | $x - ?$, | $y - ?$, | $z - ?$; |
| 14. | $\frac{(2y)^2}{\sqrt{z}/(x - 2)}$ | $x - ?$, | $y - ?$, | $z - ?$; |
| 15. | $\frac{\ln xy}{z \sin x + 0.4}$ | $x - ?$, | $y - ?$, | $z - ?$. |

Наприклад, програма для розв'язання завдання варіанту № 8 може бути такою:

```
static void Main(string[] args)
{
    double x, y, z, res;
    Console.Write("Введіть x: ");
    x = Convert.ToDouble(Console.ReadLine());
    Console.Write("Введіть y: ");
    y = Convert.ToDouble(Console.ReadLine());
    Console.Write("Введіть z: ");
    z = Convert.ToDouble(Console.ReadLine());
    res = (y + Math.Cos(z)) / (4 * y + 2 * x * Math.Sqrt(x + 2));
    Console.WriteLine("При x={0;I0:F3}, y={1;I5:N2}, z={2;I7:N3} " +
        "значення виразу рівне {3;I8:F4}", x, y, z, res);
    Console.ReadKey();
}
```

2. Скласти лінійну програму для розв'язання поставленого завдання.

Вимоги до програми:

- числові значення незалежних змінних ввести з командного рядка;

- числові значення змінних, заданих формулами, обчислити за допомогою команди присвоєння;
- значення вхідних даних та остаточні результати виконання програми вивести у вікні консольного додатку.

Варіанти

1. Відома сума початкового вкладу на рахунку в ощадбанку. Знаючи щорічний процент прибутку, обчислити суму грошей на рахунку, яка назбирається за 10 років (зв'язок між вкладеним капіталом P і майбутнім значенням A капіталу, отриманого за N років при проценті прибутку K , задається

$$\text{формулою } P = \frac{A}{\left(1 + \frac{K}{100}\right)^N};$$

2. Позначимо населення деякої країни в 1904р. через N_1 . В 1968р. воно стає рівним N_2 ($N_2 > N_1$). Обчислити середньорічний приріст населення за цей період;
3. Визначити площу кільця, якщо відомі його внутрішній і зовнішній радіуси;
4. У коло ввімкнено паралельно два провідники. Обчислити їх загальний опір;
5. Обчислити периметр і площу прямокутного трикутника, якщо відомі гіпотенуза і один з катетів;
6. Обчислити середнє арифметичне і середнє квадратичне двох натуральних чисел;
7. Обчислити площу поверхні і об'єм парника, що має форму половини циліндра;
8. Знайти час руху пішоходів, які йдуть назустріч один одному, якщо відомі швидкості обох пішоходів і початкова відстань;
9. Підрахувати зарплату працівника за тиждень, якщо відомі кількість відпрацьованих за тиждень годин (H), погодинна ставка (K) в прізвище працівника (P). Результати вивести у вигляді фрази, де вказані прізвище працівника та його зарплата;
10. Потрібно випустити картонні коробки із заданими розмірами (в сантиметрах). Визначити об'єм коробки, площу поверхні, витраченого на неї картону і вартість виробництва. За розкрій картону і фарбування коробки із заготовки береться постійна плата 2 грн., яка не залежить від розмірів коробки. Ціна самого картону – 0,02 грн. за квадратний сантиметр. Припуск картону на склеювання стінок ігнорувати;
11. Вказати вік студента якщо відомі його прізвище і рік народження.

$$\frac{A + B + |A - B|}{2}$$

12. Користуючись формулою $\text{MAX}(A, B) = \frac{A + B + |A - B|}{2}$, визначити більше з чисел K , L , M і N ;
13. Дано периметр квадрата. Обчислити площу відповідного вписаного кола і описаного кола;
14. В гастрономічному відділі продовольчого магазину такий асортимент продуктів: масло "Вершкове" по 5 грн. за 1 кг, сир "Костромський" по 7 грн.

за 1 кг, сир "Ковбасний" по 4,7 грн. за 1кг, ковбаса "Черкаська" по 7,5 грн. за 1 кг, сосиски "Шкільні" 4,3 грн. за 1 кг. Підрахувати вартість покупки, яка містить повний набір вказаних продуктів;

15. Обчислити арифметичні вирази $y = \sqrt{x}$, $R = \sin\left(x + \frac{\pi}{4}\right)$ і залишок від відділення цілого K на N.

Наприклад, програма для розв'язання завдання варіанту № 3 може мати такий вигляд:

```
static void Main(string[] args)
{
    Double R1, R2, S;
    Console.WriteLine("Введіть внутрішній радіус кільця: ");
    R1 = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("Введіть зовнішній радіус кільця: ");
    R2 = Convert.ToDouble(Console.ReadLine());
    S = Math.PI * (Math.Pow(R2, 2) - Math.Pow(R1, 2));
    Console.WriteLine("При зовнішньому радіусі {0} і внутрішньому радіусі {1} "+"
        "площа кільця становить {2,0:N3}", R2, R1, S);
    Console.ReadKey();
}
```

3. Скласти лінійну програму для розв'язання поставленого завдання.

Вимоги до програми:

- числові значення незалежних змінних ввести з командного рядка;
- числові значення змінних, заданих формулами, обчислити за допомогою команди присвоєння;
- значення вхідних даних та остаточні результати виконання програми вивести у вікні консольного додатку.

Варіанти

1. Визначити реальний дохід, якщо відомі номінальний дохід та індекс споживчих цін.
2. Визначити обсяг грошової маси, яка знаходиться в обігу, якщо відомі середня ціна на продукцію, швидкість обороту грошей та кількість виробленої продукції в постійних цінах.
3. Визначити індекс споживчих цін, якщо відомі вартості споживчої корзини в поточному році та базовому році.
4. Визначити купівельну спроможність грошей, якщо відомий індекс споживчих цін.
5. Визначити темп інфляції, якщо відомі індекси споживчих цін в поточному та базовому році.
6. Визначити суму доходу, якщо відомі середня ціна та кількість виробленої продукції.
7. Визначити рівень зайнятості, якщо відомі кількість зайнятих та загальна кількість працездатних осіб.
8. Визначити суму прибутку, якщо відомі суми доходів та витрат.

9. Визначити номінальний дохід, якщо відомі реальний дохід та індекс споживчих цін.
10. Визначити швидкість обороту грошей якщо відомі середня ціна на продукцію, обсяг грошової маси, яка знаходиться в обігу, та кількість виробленої продукції в постійних цінах.
11. Визначити вартість споживчої корзини в поточному році, якщо відомі індекс споживчих цін та вартість споживчої корзини в базовому році.
12. Визначити індекс споживчих цін, якщо відома купівельна спроможність грошей.
13. Визначити вартість споживчого кошика в поточному році, якщо відомі темп інфляції та вартість споживчого кошика в базовому році.
14. Визначити кількість виробленої продукції, якщо відомі середня ціна та суму доходу.
15. Визначити кількість зайнятих осіб, якщо відомі рівень зайнятості та загальна кількість працездатних осіб.
16. Визначити суму доходу, якщо відомі суми прибутку та витрат.

Наприклад, програма для розв'язання завдання варіанту № 16 може бути такою:

```
static void Main(string[] args)
{
    Double d, p, v;
    Console.Write("Введіть суму прибутку: ");
    p = Convert.ToDouble(Console.ReadLine());
    Console.Write("Введіть суму витрат: ");
    v = Convert.ToDouble(Console.ReadLine());
    d=p+v;
    Console.WriteLine("При сумі прибутку {0} грн. і сумі витрат {1} грн. "+
        "дохід становитиме {2,0:N2} грн.", p, v, d);
    Console.ReadKey();
}
```

З питань тиражування та викупу

Лабораторна робота № 4

- Тема.** Організація вводу/виводу з використанням діалогових вікон.
Мета. Формування вмінь і навиків використання підпрограм вводу/виводу з використанням діалогових вікон. Застосування знань структури лінійних програм. Закріплення вмінь і навиків роботи в середовищі мови програмування C# та програмування алгоритмів лінійної структури.

Контрольні запитання

1. Які додаткові бібліотеки і навіщо потрібно підключити для використання функцій вводу/виводу в діалоговому режимі?
2. Як називаються функції з однаковими назвами але різними кількостями чи типами аргументів?
3. Які аргументи і якого типу найчастіше використовує функція `Interaction.InputBox`?
4. Що повертає функція `Interaction.InputBox`?
5. Як перетворити числові значення в рядкові? Які функції застосовуються для перетворення рядкових значень в числові?
6. Які аргументи і якого типу використовує функція `MessageBox.Show`?
7. Як перейти на новий фізичний рядок при формуванні повідомлень за допомогою функцій вводу/виводу?
8. Що повертає функція `MessageBox.Show`?

Завдання

1. **Скласти лінійну програму для розв'язання поставленого завдання.**

Вимоги до програми:

- числові значення незалежних змінних ввести за допомогою функції `Interaction.InputBox`;
- числові значення змінних, заданих формулами, обчислити за допомогою команди присвоєння;
- значення вхідних даних та остаточні результати виконання програми вивести у діалоговому вікні.

Варіанти

1. Обчислити середнє арифметичне і середнє квадратичне двох натуральних чисел;
2. Обчислити площу поверхні і об'єм парника, що має форму половини циліндра;
3. Знайти час руху пішоходів, які йдуть назустріч один одному, якщо відомі швидкості обох пішоходів і початкова відстань;
4. Підрахувати зарплату працівника за тиждень, якщо відомі кількість відпрацьованих за тиждень годин (Н), погодинна ставка (К) в прізвище працівника (Р). Результати вивести у вигляді фрази, де вказані прізвище працівника та його зарплата;
5. Потрібно випускати картонні коробки із заданими розмірами (в сантиметрах). Визначити об'єм коробки, площу поверхні, витраченого на неї

картону і вартість виробництва. За розкрий картону і фарбування коробки із заготовки береться постійна плата 2 грн., яка не залежить від розмірів коробки. Ціна самого картону – 0,02 грн. за квадратний сантиметр. Припуск картону на склеювання стінок ігнорувати;

6. Вказати вік студента якщо відомі його прізвище і рік народження.

7. Користуючись формулою $\text{MAX}(A,B) = \frac{A+B+|A-B|}{2}$, визначити більше з чисел K, L, M і N;

8. Дано периметр квадрата. Обчислити площу відповідного вписаного кола і описаного кола;

9. В гастрономічному відділі продовольчого магазину такий асортимент продуктів: масло "Вершкове" по 5 грн. за 1 кг, сир "Костромський" по 7 грн. за 1 кг, сир "Ковбасний" по 4,7 грн. за 1кг, ковбаса "Черкаська" по 7,5 грн. за 1 кг, сосиски "Шкільні" 4,3 грн. за 1 кг. Підрахувати вартість покупки, яка містить повний набір вказаних продуктів;

10. Обчислити арифметичні вирази $y = \sqrt{x}$, $R = \sin\left(x + \frac{\pi}{4}\right)$ і залишок від відділення цілого K на N.

11. Відома сума початкового вкладу на рахунок в ощадбанку. Знаючи щорічний процент прибутку, обчислити суму грошей на рахунок, яка назбирається за 10 років (зв'язок між вкладеним капіталом P і майбутнім значенням A капіталу, отриманого за N років при проценті прибутку K, задається

$$\text{формулою } P = \frac{A}{\left(1 + \frac{K}{100}\right)^N};$$

12. Позначимо населення деякої країни в 1904р. через N1. В 1968р. воно стає рівним N2 (N2>N1). Обчислити середньорічний приріст населення за цей період;

13. Визначити площу кільця, якщо відомі його внутрішні і зовнішні радіуси;

14. У коло ввімкнено паралельно два провідники. Обчислити їх загальний опір;

15. Обчислити периметр і площу прямокутного трикутника, якщо відомі гіпотенуза і один з катетів;

Приклад програми для розв'язування завдання варіанту № 15. Після підключення бібліотек System.Windows.Forms та Microsoft.VisualBasic в розділі проекту Ссылки (References) у верхній частині файла коду програми необхідно **додатково** відкрити простори імен:

```
using System.Windows.Forms;  
using Microsoft.VisualBasic;
```

Тоді в просторі імен проекту код програми може бути таким:

```

class Program
{
    static void Main(string[] args)
    {
        Double a, b, c, P, S;
        String s;
        c = 5;
        a = 3;
        s = Interaction.InputBox("Введіть гіпотенузу", "Введення", c.ToString());
        c = Convert.ToDouble(s);
        s = Interaction.InputBox("Введіть довжину катета", "Введення", a.ToString());
        a = Convert.ToDouble(s);
        b = Math.Sqrt(c*c-a*a);
        P = a+b+c;
        S = (a+b)/2;
        MessageBox.Show("Якщо гіпотенуза рівна " + c.ToString() +
            ", а один з катетів " + a.ToString() +
            ", то другий катет становитиме " + b.ToString() +
            ", периметр трикутника " + P.ToString() +
            ", а площа " + S.ToString(), "Результати обчислень",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

```

2. Скласти лінійну програму для розв'язання поставленого завдання.

Вимоги до програми:

- числові значення незалежних змінних ввести за допомогою функції `Interaction.InputBox`;
- числові значення змінних, заданих формулами, обчислити за допомогою команди присвоєння;
- значення вхідних даних та остаточні результати виконання програми вивести у діалоговому вікні.

Варіанти

1. Визначити швидкість обороту грошей якщо відомі середня ціна на продукцію, обсяг грошової маси, яка знаходиться в обігу, та кількість виробленої продукції в постійних цінах.
2. Визначити темп інфляції, якщо відомі вартості споживчого кошика в поточному та базовому році.
3. Визначити суму прибутку, якщо відомі суми доходів та витрат.
4. Визначити суму доходу, якщо відомі суми прибутку та витрат.
5. Визначити суму доходу, якщо відомі середня ціна та кількість виробленої продукції.
6. Визначити рівень зайнятості, якщо відомі кількість зайнятих та загальна кількість працездатних осіб.
7. Визначити реальний дохід, якщо відомі номінальний дохід та індекс споживчих цін.

8. Визначити обсяг грошової маси, яка знаходиться в обігу, якщо відомі середня ціна на продукцію, швидкість обороту грошей та кількість виробленої продукції в постійних цінах.
9. Визначити номінальний дохід, якщо відомі реальний дохід та індекс споживчих цін.
10. Визначити купівельну спроможність грошей, якщо відомий індекс споживчих цін.
11. Визначити кількість зайнятих осіб, якщо відомі рівень зайнятості та загальна кількість працездатних осіб.
12. Визначити кількість виробленої продукції, якщо відомі середня ціна та сума доходу.
13. Визначити індекс споживчих цін, якщо відомі вартості споживчого кошика в поточному році та базовому році.
14. Визначити індекс споживчих цін, якщо відома купівельна спроможність грошей.
15. Визначити вартість споживчого кошика в поточному році, якщо відомі темп інфляції та вартість споживчого кошика в базовому році.
16. Визначити вартість споживчого кошика в поточному році, якщо відомі індекс споживчих цін та вартість споживчого кошика в базовому році.

Наприклад, код у файлі програми для розв'язання завдання варіанту № 4 може бути таким:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Microsoft.VisualBasic;

namespace LR4IODlg
{
    class Program
    {
        static void Main(string[] args)
        {
            Double v, d, p;
            String s;
            p = 5000;
            v = 12500;
            s = Interaction.InputBox("Введіть суму прибутку", "Введення", p.ToString());
            p = Convert.ToDouble(s);
            s = Interaction.InputBox("Введіть суму витрат", "Введення", v.ToString());
            v = Convert.ToDouble(s);
            d = p+v;
            MessageBox.Show("При сумі прибутку " + p.ToString() +
                " грн. та сумі витрат " + v.ToString() +
                " грн. \ndохід становитиме " + d.ToString() +
                " грн. ", "Результати обчислень",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
}
```

Лабораторна робота № 5

Тема. Використання повного та неповного операторів розгалуження.

Мета. Формування вмінь і навиків програмування алгоритмів розгалуженої структури. Закріплення вмінь і навиків використання функцій вводу-виводу. Застосування вмінь і навиків програмування алгоритмів лінійної структури.

Контрольні запитання

1. Які різновиди операторів розгалуження використовуються в C#?
2. Коли використовується кожен з різновидів операторів розгалуження?
3. Як виконується кожен з операторів розгалуження в C#?
4. Як вірно підібрати початкові дані для перевірки коректності функціонування програм, що містять команду розгалуження?
5. Які додаткові бібліотеки, як і навіщо слід підключити до консольного додатку, щоб виконати ввід/вивід у діалогових вікнах?

Завдання

1. **Скласти програму обчислення значення функції з використанням неповного оператора розгалуження.**

Вимоги до програм:

- числові значення параметра обчислень вводити з клавіатури за допомогою процедури `ReadLine`;
- для обчислення значень незалежної змінної використати неповний оператор розгалуження;
- початкові дані та результати обчислень вивести у командному рядку за допомогою `WriteLine`;
- забезпечити коректність функціонування програми на всіх вітках розгалуження.

Варіанти

$$1. Y = \begin{cases} \sin^2 x & 0 \leq x < 1 \\ \sqrt{\ln x} \cdot \sin x & 1 \leq x \leq 3 \\ \ln x & x > 3, x < 0 \end{cases};$$

$$2. Y = \begin{cases} -\arctg((x + \pi) / x^2) & 0 < x \leq 1.2 \\ \ln \sqrt{x^3} & 1.2 < x < 9 \\ e^{-x} & x \geq 9, x \leq 0 \end{cases};$$

$$3. Y = \begin{cases} 2x^3 \sqrt{x^2 + 5} & 1 < x \leq 12 \\ \arctg x & 0 < x \leq 1 \\ e^{x+3} & x \leq 0, x > 12 \end{cases};$$

$$4. Y = \begin{cases} \ln \left| \frac{\pi}{15} - x \right| & 0 \leq x < \frac{1}{4} \\ (x^2 + 2,04)^{-3,14} & \frac{1}{4} \leq x < 1 ; \\ \arccos \frac{x}{4} & x \geq 1, x < 0 \end{cases}$$

$$5. Y = \begin{cases} 0 & x \leq -1 \\ ctg \frac{x-1}{e} & -1 < x \leq 0 ; \\ \ln x & x > 0 \end{cases}$$

$$6. Y = \begin{cases} e^{-|x|} & x \geq 1 \\ \ln \sqrt{1-x^2} & |x| < 1 ; \\ arctg x & x \leq -1 \end{cases}$$

$$7. Y = \begin{cases} 2^{x-1} + 3,5 & \pi \leq x < 8,6 \\ \sqrt{|\pi - 3x|} & -8,6 < x < \pi ; \\ 2,7 & |x| \geq 8,6 \end{cases}$$

$$8. Y = \begin{cases} 1,3^{2+x} * x^2 & |x| \geq 5 \\ \ln^{x-1} & 1 \leq x \leq 1 ; \\ \cos^{|x-1|} & 1 < |x| < 5 \end{cases}$$

$$9. Y = \begin{cases} \sin(-x^2 + 4) & x \leq -3 \\ 2x + \ln 4,4 & x \geq 0 ; \\ e^x & -3 < x < 0 \end{cases}$$

$$10. Y = \begin{cases} \sqrt[3]{\ln x + \ln x^2} & x > 1 \\ e^{-x} + 1 & 0 < x \leq 1 ; \\ \cos^3 x & x \leq 0 \end{cases}$$

$$11. Y = \begin{cases} \ln(x + 13,2) & 4 < |x| < 10 \\ \sin e^x - 2 & |x| \leq 4 ; \\ 1,5x & x < -4, x \geq 10 \end{cases}$$

$$12. Y = \begin{cases} \sqrt{\sin^3(x-1)} & -1,2 \leq x \leq 1,2 \\ e^{-x} & x \geq 6 \\ 3,5 \ln|x| & 1,2 > |x| > 6 \\ 1,5x & x \leq -6 \end{cases};$$

$$13. Y = \begin{cases} e^{-2,5x^3} & x < 0 \\ \sqrt{|0,05 - \ln x|} & 1 < x < 5 \\ \frac{x-1}{x - \sin^2 x} & x > 5 \\ 2x & x \leq -6 \end{cases};$$

$$14. Y = \begin{cases} x^e - e^{-x} & |x| < 2 \\ \ln x^2 & x \leq -2; \\ \sin^2 x & x \geq 2 \end{cases};$$

$$15. Y = \begin{cases} \cos^3 x^2 - \sin x & x \geq 0 \\ x^2 - 0,83 & 0 < x < 3. \\ \frac{1,4 + x}{\ln x} & x \geq 3 \end{cases}.$$

Наприклад, програма для розв'язання першого варіанту завдання може мати такий вигляд:

```
class Program
{
    static void Main(string[] args)
    {
        double x, y=0;
        Console.Write("Введіть x = ");
        x = Convert.ToDouble(Console.ReadLine());
        if (x >= 0 && x < 1)
            y = Math.Pow(Math.Sin(x), 2);
        if (x >= 1 && x <= 3)
            y = Math.Sqrt(Math.Abs(Math.Log10(x) - Math.Sin(x)));
        if (x > 3 || x < 0)
            y = x * Math.Pow(Math.Log10(x), 3);
        Console.WriteLine("При x = {0} обчислено y = {1}", x, y);
        Console.ReadKey();
    }
}
```


2. Скласти програму обчислення значення функції завдання №1 з використанням повного оператора розгалуження.

Вимоги до програм:

- числові значення параметра обчислень ввести з клавіатури за допомогою функції `InputBox`;
- для обчислення значень незалежної змінної використати повний оператор розгалуження;
- початкові дані та результати обчислень вивести в одному діалоговому вікні за допомогою функції `MessageBox`;
- забезпечити коректність функціонування програми на всіх вітках розгалуження.

Наприклад, програма для розв'язання першого варіанту завдання може мати вигляд:

```
using Microsoft.VisualBasic;
using System.Windows.Forms;

namespace LR5
{
    class Program
    {
        static void Main(string[] args)
        {
            double x, y=0;
            string S;
            x = 93;
            S = Interaction.InputBox("Введіть x:", "Введення", x.ToString());
            x = Convert.ToDouble(S);
            if (x >= 0 & x < 1)
                y = Math.Pow(Math.Sin(x), 2);
            else
                if (x >= 1 & x <= 3)
                    y = Math.Sqrt(Math.Abs(Math.Log10(x) - Math.Sin(x)));
                else
                    y = x * Math.Pow(Math.Log10(x), 3);
            MessageBox.Show("При x = " + x.ToString() + " обчислено y = " + y.ToString(), "Результати",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
}
```

3. Скласти програму обчислення значення функції завдання №1 з можливістю виконання неодноразових обчислень.

Вимоги до програм:

- числові значення параметра обчислень ввести з клавіатури за допомогою функції `InputBox`;
- початкові дані та результати обчислень вивести в одному діалоговому вікні за допомогою функції `MessageBox` з можливістю повторного виконання програми;
- забезпечити коректність функціонування програми на всіх вітках розгалуження.

Лабораторна робота № 6

Тема. Застосування операторів розгалуження для розв'язування прикладних задач.

Мета. Формування вмінь і навиків використання операторів розгалуження для розв'язування прикладних задач. Закріплення вмінь і навиків використання функцій вводу-виводу. Застосування вмінь і навиків програмування алгоритмів лінійної структури.

Контрольні запитання

1. Коли застосовуються вкладені оператори розгалуження?
2. Від чого залежить швидкість виконання вкладених операторів розгалуження?
3. Як забезпечити вивід результатів декількох вкладених операторів розгалуження в одному діалоговому вікні?
4. Як записується і виконується команда безумовного переходу в C#?
5. Як забезпечити повторне виконання програми за бажанням користувача?

Завдання

1. **Скласти програму для розв'язання поставленого завдання.**

Вимоги до програм:

- числові значення параметрів обчислень ввести з клавіатури в діалоговому вікні;
- результати обрахунків програми вивести в одному діалоговому вікні;
- після відображення результатів обрахунків програми передбачити можливість при бажанні користувача повторного розв'язання завдання з іншими значеннями параметрів обчислень.

Варіанти

1. Два числа задані формулами

$$x = \frac{m-3n}{7}, y = \frac{2m+n}{3-n^2}.$$

Перерозподілити значення змінних так, щоб в x виявилось більше із цих чисел, а в y - менше. Вивести початкові і кінцеві значення x та y ;

2. Три числа задані формулами

$$x = \frac{\sqrt{m+3n}}{9.3}, \quad y = mn - 8, \quad z = \frac{n-3m}{2}.$$

Знайти і вивести число

$$a = \max(x, z) * (\min((x, y) - 1))^2;$$

3. Три числа задані формулами

$$l = \frac{n-3m}{2}, \quad k = \frac{2n+m}{n}, \quad h = nm - 9.3.$$

Вибрати ті з них, модулі яких менші 5. Вивести їх та їхні квадрати;

4. Три числа задані формулами

$$a = \frac{n+m}{4}, \quad b = \ln n - 3.5, \quad c = \frac{n-m}{n}.$$

Вказати ті з них, які лежать поза проміжком (1;5);

5. Три числа задані формулами

$$a = \frac{n-4m}{8}, \quad b = \frac{n+8m}{m+1}, \quad c = \frac{n-m}{n}.$$

Впорядкувати їх за зростанням;

6. Три числа задані формулами

$$a = \frac{2n+m}{5}, \quad b = \frac{n}{m+5}, \quad c = nm + 4.2.$$

Знайти число

$$p = \min(a, b) + \max(b, c);$$

7. Три числа задані формулами

$$a = \frac{m+n}{5}, \quad b = \frac{mn}{m-1}, \quad c = 3.5n.$$

Вибрати серед них від'ємні і вивести їх та квадрати і корені їхніх модулів;

8. Три числа задані формулами

$$a = 2m, \quad b = \frac{\sqrt{mn}}{3}, \quad c = m + 2n.$$

Знайти кількість максимальних серед цих чисел;

9. Три числа задані формулами

$$a = 2m - n, \quad b = m + 0.5n, \quad c = \frac{52}{2-m}.$$

Вибрати серед них середнє по величині;

10. Три числа задані формулами

$$a = \frac{n+m}{5}, \quad b = \frac{n-m}{m}, \quad c = nm + 4.2.$$

Вказати кількість різних серед цих чисел;

11. Три числа задані формулами

$$a = \frac{n-m}{7}, \quad b = nm - 15, \quad c = \frac{\sqrt{n-2m}}{4}.$$

Вказати серед них ті, що належать відрізьку [-4;7];

12. Три числа задані формулами

$$a = \frac{m+4n}{5m}, \quad b = \frac{4m-n}{n}, \quad c = mn - 10.5.$$

Впорядкувати їх за спаданням;

13. Три числа задані формулами

$$a = \frac{n-2m}{n}, \quad b = nm-3.7, \quad c = \frac{n+m}{n}.$$

Вивести збільшене на 7 найменше з цих чисел;

14. Три числа задані формулами

$$a = n\sqrt{n+m}, \quad b = nm-8, \quad c = \frac{-m}{n}.$$

Знайти кількість мінімальних серед цих чисел;

15. Три числа задані формулами

$$a = \frac{n-3m}{2}, \quad b = nm+2, \quad c = \frac{n+m}{n-0.5}.$$

Вказати кількість різних серед цих чисел.

Наприклад, програма для розв'язання завдання варіанту № 1 може бути такою:

```
static void Main(string[] args)
{
    Double m, n, x, y, PROM;
    String S;
    m = 12;
    n = 37;
Povtor:
    S = m.ToString();
Povtor1:
    S = Interaction.InputBox("Введіть m ", "Введення", S);
    try
    {
        m = Convert.ToDouble(S);
    }
    catch (System.FormatException)
    {
        if (MessageBox.Show("Ви ввели не число" + Strings.Chr(13) + Strings.Chr(13) + "Бажаєте повторити?",
            "Увага", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
            goto Povtor1;
        else
            return;
    }
    S = n.ToString();
Povtor2:
    S = Interaction.InputBox("Введіть n ", "Введення", S);
    try
    {
        n = Convert.ToDouble(S);
    }
}
```

15

```

catch (System.FormatException)
{
    if (MessageBox.Show("Ви ввели не число" + Strings.Chr(13) + Strings.Chr(13) + "Бажаєте повторити?",
        "Увага", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
        goto Povtor2;
    else
        return;
}
x = (m - 3 * n) / 7;
y = (2 * m + n) / (3 - n * n);
if (x > y)
{
    PROM = x;
    x = y;
    y = PROM;
}
if (MessageBox.Show("При m = " + m.ToString() + " та n = " + n.ToString() + " обчислено x = " +
    x.ToString() + " y = " + y.ToString() + Strings.Chr(13) + Strings.Chr(13) +
    "Бажаєте повторити?", "Результати обчислень", MessageBoxButtons.YesNo,
    MessageBoxIcon.Information) == DialogResult.Yes)
    goto Povtor;
}
}

```

54

2. Скласти програму для розв'язання квадратного рівняння з довільними дійсними коефіцієнтами.

Вимоги до програм:

- числові значення параметрів обчислень ввести з клавіатури в діалоговому вікні;
- при введенні недопустимих значень чи типів параметрів обчислень передбачити виведення відповідного повідомлення та можливість завершення роботи програми чи повторного їх введення;
- результати обрахунків програми вивести в одному діалоговому вікні;
- після відображення результатів обрахунків програми передбачити можливість при бажанні користувача повторного розв'язання завдання з іншими значеннями параметрів обчислень.

Ця програма може виглядати, наприклад, так:

```

using System;
using Microsoft.VisualBasic;
using System.Windows.Forms;

namespace LR6aKvadrRivn
{
    class Program
    {
        static bool InputDouble(ref double x, string povidom)
        {
            string s;
            s = x.ToString();
        Povtor:
            s = Interaction.InputBox(povidom, "Введення", s);
            try
            {
                x = Convert.ToDouble(s);
            }
            catch (System.FormatException)
            {
                if (MessageBox.Show("Ви ввели не число.\n\nБажаєте повторити?", "Увага",
                    MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
                    goto Povtor;
                else
                    return false;
            }
            return true;
        }
    }
}

```

```

static void Main(string[] args)
{
    Double a, b, c, x1, x2, D;
    String REZ;
    a = 25;
    b = 7;
    c = -3;
    Povtor:
    if (!InputDouble(ref a, "Введіть a "))
        return;
    if (!InputDouble(ref b, "Введіть b "))
        return;
    if (!InputDouble(ref c, "Введіть c "))
        return;
    if (a != 0)
    {
        D = b * b - 4 * a * c;
        if (D > 0)
        {
            x1 = ((-b - Math.Sqrt(D)) / (2 * a));
            x2 = ((-b + Math.Sqrt(D)) / (2 * a));
            REZ = "рівняння має два корені x1 = " + x1.ToString() + ", x2 = " + x2.ToString();
        }
        else
            if (D < 0)
                REZ = "рівняння розв'язків не має";
            else
                {
                    x1 = -b / (2 * a);
                    REZ = "рівняння має кратний корінь x = " + x1.ToString();
                }
        }
    else
    {
        if (b == 0)
        {
            if (c == 0)
                REZ = "рівняння має безліч розв'язків";
            else
                REZ = "рівняння розв'язків не має";
        }
        else
        {
            x1 = -c / b;
            REZ = "рівняння має один корінь x = " + x1.ToString();
        }
    }
    if (MessageBox.Show("При a = " + a.ToString() + ", b = " + b.ToString() + ", c = " + c.ToString() + " " +
        REZ + "\n\nБажаєте рахувати ще?", "Результати обчислень", MessageBoxButtons.YesNo,
        MessageBoxIcon.Information) == DialogResult.Yes)
        goto Povtor;
    }
}

```

Лабораторна робота № 7

Тема. Обробка виключних ситуацій. Засоби відлагодження розроблених програм.

Мета. Формування вмінь і навиків створення та використання процедур та функцій користувача. Закріплення вмінь і навиків використання функцій вводу-виводу та обробки рядків. Застосування вмінь і навиків програмування алгоритмів лінійної та розгалуженої структури.

Контрольні запитання

1. Навіщо забезпечувати захист програми від некоректного вводу даних?
2. Що таке виключні ситуації? Як опрацьовуються виключні ситуації в C#?
3. Який механізм обробки виключних ситуацій використовується в C#?
4. В яких фрагментах коду програми доцільно відключати стандартну обробку виключних ситуацій? Чому?
5. Як в програмі обробити різні виключні ситуації? Всі виключні ситуації одночасно?
6. Коли та з якою метою доцільно використовувати підпрограми?
7. Де описуються підпрограми користувача?
8. Як повернути результат виконання підпрограми-функції в основну програму?
9. Для чого і як застосовується покрокове відлагодження програм?
10. Які варіанти кроків доступні в процесі покрокового відлагодження програм?

Завдання

1. **Скласти програму для розв'язання квадратного рівняння з довільними дійсними коефіцієнтами, використовуючи підпрограму для коректного введення даних.**

Вимоги до програми:

- значення змінних ввести з діалогових вікон (функції *InputInt* чи *InputDouble*);
- при введенні недопустимих значень чи типів параметрів обчислень передбачити виведення відповідного повідомлення та можливість завершення роботи програми чи повторного їх введення;
- результати обрахунків програми вивести в одному діалоговому вікні;
- після відображення результатів обрахунків програми передбачити можливість при бажанні користувача повторного розв'язання завдання з іншими значеннями параметрів обчислень.

Ця програма може виглядати, наприклад, так:

```

using System;
using Microsoft.VisualBasic;
using System.Windows.Forms;

namespace LR6aKvadrRivn
{
    class Program
    {
        static bool InputDouble(ref double x, string povidom)
        {
            string s;
            s = x.ToString();
        Povtor:
            s = Interaction.InputBox(povidom, "Введення", s);
            try
            {x = Convert.ToDouble(s);
            }
            catch (System.FormatException)
            {if (MessageBox.Show("Ви ввели не число.\n\пбажаєте повторити?", "Увага",
                MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
                goto Povtor;
            else
                return false;
            }
            return true;
        }

        static void Main(string[] args)
        {
            Double a, b, c, x1, x2, D;
            String REZ;
            a = 25;
            b = 7;
            c = -3;
        Povtor:
            if (!InputDouble(ref a, "Введіть а "))
                return;
            if (!InputDouble(ref b, "Введіть b "))
                return;
            if (!InputDouble(ref c, "Введіть c "))
                return;
            if (a != 0)
            {
                D = b * b - 4 * a * c;
                if (D > 0)
                {
                    x1 = ((-b - Math.Sqrt(D)) / (2 * a));
                    x2 = ((-b + Math.Sqrt(D)) / (2 * a));
                    REZ = "рівняння має два корені x1 = " + x1.ToString() + ", x2 = " + x2.ToString();
                }
                else
                if (D < 0)
                    REZ = "рівняння розв'язків не має";
                else
                {
                    x1 = -b / (2 * a);
                    REZ = "рівняння має кратний корінь x = " + x1.ToString();
                }
            }
            else
        }
    }
}

```



```

{
    if (b == 0)
    {
        if (c == 0)
            REZ = "рівняння має безліч розв'язків";
        else
            REZ = "рівняння розв'язків не має";
    }
    else
    {
        x1 = -c / b;
        REZ = "рівняння має один корінь x = " + x1.ToString();
    }
}
if (MessageBox.Show("При a = " + a.ToString() + ", b = " + b.ToString() + ", c = " + c.ToString() + " " +
    REZ + "\n\nБажаєте рахувати ще?", "Результати обчислень", MessageBoxButtons.YesNo,
    MessageBoxIcon.Information) == DialogResult.Yes)
    goto Povtor;
}
}
}

```

2. Скласти програму для розв'язання завдання № 1 попередньої лабораторної роботи з використанням функцій користувача та обробкою виключних ситуацій.

Вимоги до програми:

- значення змінних ввести з діалогових вікон (функції *InputInt* чи *InputDouble*, наведені вище);
- при введенні недопустимих значень чи типів параметрів обчислень передбачити виведення відповідного повідомлення та можливість завершення роботи програми чи повторного їх введення;
- результат виконання програм вивести в одному діалоговому вікні;
- після відображення результатів обрахунків програми передбачити можливість при бажанні користувача повторного розв'язання завдання з іншими значеннями параметрів обчислень.

З питань тиражування та використання матеріалу зверніться на сайт

Лабораторна робота № 8

Тема. Оператор вибору. Створення найпростіших тестових програм.

Мета. Формування вмінь і навиків програмування алгоритмів розгалуженої структури з використанням оператора вибору. Закріплення вмінь і навиків використання функцій вводу-виводу. Застосування вмінь і навиків програмування алгоритмів лінійної та розгалуженої структури.

Контрольні запитання

1. Коли в програмуванні використовуються оператори розгалуження, а коли – оператор вибору?
2. Як описується та виконується оператор вибору в C#?
3. Скільки параметрів може одночасно аналізувати оператор вибору?
4. Скільки варіантів діапазонів значень параметра може обробляти оператор вибору? Якій умові мають задовольняти ці діапазони? Які оператори виконуються при виявленні належності параметра вибору одному з діапазонів? Які оператори виконуються при умові належності параметра вибору декільком діапазнам?
5. Чому оператор вибору виконується швидше аналогічної сукупності послідовних операторів розгалуження?

Завдання

1. Розробити найпростішу навчально-контролюючу програму.

Вимоги до програми:

- при проведенні тестування передбачити введення відповідей не менше ніж на п'ять запитань;
- деякі контрольні запитання повинні вимагати введення правильної відповіді безпосередньо з клавіатури чи за допомогою кнопок, а інші – введення номера правильної відповіді з наведеного списку;
- після введення відповіді на кожне запитання проаналізувати її правильність та вивести відповідне повідомлення;
- після завершення проведення тестування за його результатами сформулювати за допомогою оператора вибору відповідні рекомендації, вивести повідомлення про кількість питань, на які була дана правильна відповідь, загальну кількість запропонованих питань та отриману оцінку (орієнтовно: за 5 правильних відповідей – 10, за 4 – 8, за 3 – 6, менше 3 - 4).

Варіанти тем навчально-контролюючих програм

1. Стандартні арифметичні функції мови програмування C#;
2. Синтаксис запису арифметичних виразів мовою C#;
3. Таблиця похідних;
4. Роль груп клавіш клавіатури комп'ютера;
5. Засоби відлагодження програм C#;
6. Основні оператори мови C#;
7. Формули скороченого множення;

8. Таблиця інтегралів;
9. Переклад слів іноземної мови на рідну;
10. Парність, непарність та монотонність функцій;
11. Основні властивості паралелограма, прямокутника, ромба та квадрата;
12. Послідовність обчислень значень арифметичних виразів, що містять різні арифметичні операції та дужки;
13. Формули обчислень площ поверхонь і об'ємів многогранників;
14. Формули скороченого множення;
15. Оператори розгалуження та вибору у C#.

Наприклад, тестова програма для перевірки знань інформації про групу може бути такою:

```
static void Main(string[] args)
{
    Повтор:
        int CountPut = 0, CountBal = 0, CountVirno = 0;
        String S;
        if (MessageBox.Show("Чи з кожної пари відпрошується Балаушко?", "Питання №1",
            MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            CountBal = CountBal + 2;
            CountVirno++;
        }
        else
            MessageBox.Show("Ви помилилися! Балаушко відпрошується з кожної пари", "Увага!",
                MessageBoxButtons.OK, MessageBoxIcon.Error);

        CountPut++;
        if (Interaction.InputBox("Який шифр групи, де навчається Коваленко?", "Питання №2", "КН-51") == "ПІ-51")
        {
            CountBal = CountBal + 2;
            CountVirno++;
        }
        else
            MessageBox.Show("Ви відповіли невірно! Коваленко навчається в групі ПІ-51", "Увага!",
                MessageBoxButtons.OK, MessageBoxIcon.Error);

        CountPut++;
        if (Interaction.InputBox("Хто староста групи КН-51?\n1. Гординський\n2. Пашко\n3. Балаушко" +
            "\n\nВведіть номер правильної відповіді", "Питання №3", "3") == "2")
        {
            CountBal = CountBal + 2;
            CountVirno++;
        }
        else
            MessageBox.Show("Ви відповіли не вірно! Старостою групи ПІ-51 є Пашко", "Увага!",
                MessageBoxButtons.OK, MessageBoxIcon.Error);

        CountPut++;
        //...
        switch (CountVirno)
        {
            case 6:
                S = "відмінно";
                break;
            case 5:
                S = "дуже добре";
                break;
        }
    }
}
```

```

    case 4:
        S = "добре";
        break;
    case 3:
        S = "задовільно";
        break;
    case 2:
        S = "незадовільно";
        break;
    case 1:
        S = "погано";
        break;
    default:
        S = "дуже погано! Жодна відповідь неправильна";
        break;
}
if (MessageBox.Show("За результатами тестування дано "+ CountVirno.ToString() +
    " правильних відповідей з " + CountPut.ToString() + ".\nКількість набраних балів: " +
    CountBal.ToString() + Strings.Chr(13) + "Ваш результат: " + S +
    "\n\nБажаєте тестуватися ще?", "Результати тестування",
    MessageBoxButtons.YesNo, MessageBoxIcon.Information) == DialogResult.Yes)
    goto Povtor;
}

```

З питань тиражування та використання цього видання звертайтеся .

Лабораторна робота № 9

- Тема.** Оператори циклу з перед- та післяумовою.
Мета. Формування вмінь і навиків програмування алгоритмів циклічної структури з перед- та післяумовою. Закріплення вмінь і навиків використання функцій вводу-виводу. Застосування вмінь і навиків програмування алгоритмів лінійної та розгалуженої структури.

Контрольні запитання

1. Що таке цикл? Які види циклів використовуються в програмуванні?
2. У чому полягають відмінності між різними видами циклів?
3. Як забезпечити циклічне виконання декількох операторів?
4. Як перервати виконання чергової та перейти до наступної ітерації циклу? Як перервати виконання циклу?
5. Які перевірки значень змінних необхідно виконати перед виконанням операторів циклу?
6. Коли виникає зациклення програми? Як перервати виконання програми при зацикленні? Як віднайти причину зациклення програми?

Завдання

1. **Скласти програму обчислення значень функції на вказаному проміжку із заданим кроком, а також програму обчислення значень функції, починаючи з заданої точки у вказаній кількості точок.**

Вимоги до програми:

- виконати ввід/вивід для у вікні консольного додатку;
- числові значення параметрів обчислень ввести з клавіатури;
- значення аргументів та відповідні їм значення результатів вивести на друк у вигляді таблиці.

Варіанти

1. $Y = \frac{\sin^2 x}{(x+1)^2}$ при а) $3.8 \leq x \leq 7.6$, $\Delta x = 0.4$; б) $0.5 \leq x$, $\Delta x = 0.1$, $n = 9$;
2. $Y = \frac{e^{2x} - 8}{x + 3}$ при а) $2.1 \leq x \leq 5.3$, $\Delta x = 0.7$; б) $1.5 \leq x$, $\Delta x = 0.3$, $n = 6$;
3. $Y = \frac{x + \cos 2x}{3x}$ при а) $2.3 \leq x \leq 5.4$, $\Delta x = 0.8$; б) $1.2 \leq x$, $\Delta x = 0.2$, $n = 7$;
4. $Y = \frac{x + \sin 2xe^{2x} - 8}{x + 2}$ при а) $0.2 \leq x \leq 9.8$, $\Delta x = 0.8$; б) $0.6 \leq x$, $\Delta x = 1.5$, $n = 6$;
5. $Y = \frac{\cos^3 x^2}{1.5x + 2}$ при а) $2.3 \leq x \leq 7.2$, $\Delta x = 0.6$; б) $0.8 \leq x$, $\Delta x = 0.3$, $n = 5$;
6. $Y = \frac{x + \sin 2x}{x^2 - 3}$ при а) $2.4 \leq x \leq 6.9$, $\Delta x = 0.4$; б) $3.1 \leq x$, $\Delta x = 0.8$, $n = 6$;

7. $Y = \frac{x^2 + 2}{3 \cos \sqrt{x} + 1}$ при а) $0.4 \leq x \leq 2.7$, $\Delta x = 0.4$; б) $0.3 \leq x$, $\Delta x = 0.8$, $n = 7$;
8. $Y = \frac{x^3 - 3}{3 \ln x}$ при а) $4.5 \leq x \leq 8.4$, $\Delta x = 1.2$; б) $2.5 \leq x$, $\Delta x = 1.5$, $n = 5$;
9. $Y = \frac{2 \sin^2(x+2)}{x^2 + 1}$ при а) $4.3 \leq x \leq 7.8$, $\Delta x = 0.5$; б) $0.7 \leq x$, $\Delta x = 0.1$, $n = 5$;
10. $Y = \frac{(3x+2)^2}{\sin x + 3}$ при а) $4.8 \leq x \leq 7.9$, $\Delta x = 0.4$; б) $0.2 \leq x$, $\Delta x = 0.7$, $n = 6$;
11. $Y = \frac{1.9x - \ln 3x}{3x + 1}$ при а) $2.5 \leq x \leq 9.1$, $\Delta x = 0.8$; б) $0.8 \leq x$, $\Delta x = 1.2$, $n = 6$;
12. $Y = \frac{2x + 8}{|\cos 2.5x| + 1}$ при а) $2.7 \leq x \leq 6.5$, $\Delta x = 0.8$; б) $0.1 \leq x$, $\Delta x = 0.3$, $n = 6$;
13. $Y = \frac{x - \ln 2x}{2x + 1}$ при а) $2.1 \leq x \leq 8.5$, $\Delta x = 0.7$; б) $0.6 \leq x$, $\Delta x = 2.5$, $n = 5$;
14. $Y = \frac{x + \tan 2x}{3x}$ при а) $2.7 \leq x \leq 7.1$, $\Delta x = 0.7$; б) $0.8 \leq x$, $\Delta x = 0.2$, $n = 6$;
15. $Y = \frac{(x+7)^2}{\sqrt{x^2 + 1}}$ при а) $2.3 \leq x \leq 7.5$, $\Delta x = 0.6$; б) $6.5 \leq x$, $\Delta x = 0.3$, $n = 9$.

Наприклад, програма для розв'язування першої задачі варіанту № 3 може мати вигляд:

```
static void Main(string[] args)
{
    Double x, y, xp, xk, dx;
    Console.WriteLine("Введіть x початкове: ");
    xp = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("Введіть x кінцеве: ");
    xk = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("Введіть крок: ");
    dx = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("x   y");
    x = xp;
    while (x <= xk)
    {
        y = (x + Math.Cos(2 * x)) / (3 * x);
        Console.WriteLine("{0}\t{1}", x, y);
        x += dx; //або x=x+dx;
    }
    Console.ReadKey();
}
```

2. Скласти програму обчислення значень функції на вказаному проміжку із заданим кроком, а також програму обчислення значень функції, починаючи з заданої точки у вказаній кількості точок по варіантах першого завдання.

Вимоги до програми:

- виконати ввід/вивід у діалогових вікнах;
- числові значення параметрів обчислень ввести з клавіатури, константи з умов завдання використати як значення по замовчуванню;
- при введенні недопустимих значень чи типів параметрів обчислень передбачити виведення відповідного повідомлення та можливість завершення роботи програми чи повторного їх введення;
- значення аргументів та відповідні їм значення результатів вивести на друк у вигляді таблиці.

Наприклад, програма для розв'язування другої задачі варіанту № 3 може мати вигляд:

```
class Program
{
    static bool inputInt(ref int i, string povidom)
    {
        string s;
        s = i.ToString();
    povtor:
        s = Interaction.InputBox(povidom, "Введення", s);
        try
        {i = Convert.ToInt32(s);
        }
        catch (System.FormatException)
        { if (MessageBox.Show("Ви ввели не ціле число." + Strings.Chr(13) + "Бажаєте повторити?", "Увага",
            MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
            goto povtor;
            else
                return false;
        }
        return true;
    }
}
```

З питань тиражу.

```

static bool inputDouble(ref double x, string povidom)
{
    string s;
    s = x.ToString();
    povtor:
    s = Interaction.InputBox(povidom, "Введення", s);
    try
    {
        x = Convert.ToDouble(s);
    }
    catch (System.FormatException)
    {
        if (MessageBox.Show("Ви ввели не число." + Strings.Chr(13) + "Бажаєте повторити?", "Увага",
            MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
            goto povtor;
        else
            return false;
    }
    return true;
}

static void Main(string[] args)
{
    int n, i;
    double x, y, xp, dx;
    xp = 1.2;
    dx = 0.2;
    n = 9;
    if (!inputDouble(ref xp, "Введіть x початкове:"))
        return;
    if (!inputDouble(ref dx, "Введіть крок:"))
        return;
    if (!inputInt(ref n, "Введіть кількість точок"))
        return;
    string rez="x\ty\n";
    x = xp;
    i = 0;
    while (i < n)
    {
        y = (x + Math.Cos(2 * x)) / (3 * x);
        rez += x.ToString() + "\t" + y.ToString()+"\n";
        x += dx;
        i++;
    }
    MessageBox.Show(rez, "Результати табулювання");
}
}

```


3. Скласти програму для розв'язання поставленого завдання (додатково 5 балів до рейтингу).

Вимоги до програми:

- числові значення параметрів обчислень ввести з клавіатури з відповідною перевіркою коректності типів та даних;
- при створенні програми використати оператор циклу з перед- чи післяумовою;
- масиви не використовувати;
- результати обчислень вивести в одному діалоговому вікні.

Варіанти

1. При заданому x обчислити наближено суму $S = e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$ ($0! = 1$), припиняючи обчислення, коли черговий член суми за абсолютною величиною буде меншим 0.001;
2. При заданому x обчислити суму $S = \sin x = \sum_{i=0}^{\infty} \frac{(-1)^i x^{2i+1}}{(2i+1)!}$, припиняючи обчислення, коли черговий член суми за абсолютною величиною буде меншим 0.0001;
3. Населення двох країн рівні N_1 і N_2 , а приріст за рік – відповідно P_1 і P_2 . Відомо, що $N_1 > N_2$, і $P_1 < P_2$. Підрахувати, через скільки років населення другої країни перевищить населення першої країни;
4. Розв'язати рівняння $x = 0.5(\sin x^2 - 1)$, обчислюючи $x_{n+1} = 0.5(\sin x_n^2 - 1)$, $n = 1, 2, \dots$ при $x_0 = 0.4$. Обчислення припинити при виконанні умови, коли буде виконана умова $|x_{n+1} - x_n| \leq 0.0001$;
5. Комп'ютер (за допомогою датчика випадкових чисел) генерує результати кидань двох кубиків до тих пір, поки не випаде дві шістки. Вияснити, через скільки кидань станеться ця подія. Вивести всі числа, що при цьому генеруються;
6. Почавши тренування, спортсмен за перший день пробіг M км. Кожного наступного дня він збільшував денну норму на 3% відносно норми попереднього дня. Визначити, через скільки днів спортсмен буде пробігати за день більше N км ($N > M$);
7. При умові попереднього завдання визначити, через скільки днів спортсмен пробіжить сумарний шлях не менше L км;
8. При заданому x обчислити суму $S = \cos x = \sum_{i=0}^{\infty} \frac{(-1)^i x^{2i}}{2i!}$, припиняючи обчислення, коли черговий член суми за абсолютною величиною буде меншим 0.0001;
9. Лісництво веде заготівлю ділової деревини. Початковий об'єм деревини на території лісництва був рівний 120000 м³. Щорічний природний приріст становить 5.5%. Річний план заготівлі деревини – 9500 м³. Через скільки

років об'єм деревини на території лісництва стане меншим 23000м³?
Вивести об'єм деревини на кінець кожного року до настання цієї події;

10. Визначити перший від'ємний елемент послідовності

$$a_n = \sin\left(\frac{n}{100}\right), \quad n = 1, 2, \dots$$

Вивести номер і значення цього елемента послідовності;

11. Обчислити добуток перших семи додатних елементів послідовності

$$a_n = \tan n, \quad n = 1, 2, \dots;$$

12. Обчислити суму перших десяти додатних членів послідовності

$$a_n = \cos n, \quad n = 1, 2, \dots;$$

13. Дано дійсне число $b > 0$. Послідовність a_1, a_2, \dots, a_n утворена за законом

$$a_1 = b, \quad a_i = a_{i-1} - \frac{1}{\sqrt{i}}, \quad i = 1, 2, \dots$$

Знайти перший від'ємний елемент цієї послідовності;

14. Дано дійсне число $b > 0$. Послідовність a_1, a_2, \dots, a_n утворена за законом

$$a_1 = 1, \quad a_i = a_{i-1}^2 + 1, \quad i = 1, 2, \dots$$

Отримати всі a_i , що не перевищують b ;
15. Вирахувати n -й член послідовності, утвореної дробами $\frac{1}{1}, \frac{2}{1}, \frac{3}{2}, \dots$, де чисельник (знаменник) наступного елемента послідовності, починаючи з третього, отримується додаванням чисельників (знаменників) двох попередніх елементів.

Наприклад, програма для розв'язування завдання варіанту № 6 може мати вигляд:

```
static void Main(string[] args)
{
    int a, b, count=0;
    Random rnd = new Random();
    do
    {
        a = 1 + Convert.ToInt32(rnd.NextDouble() * 5);
        b = 1 + Convert.ToInt32(rnd.NextDouble() * 5);
        Console.WriteLine("{0} {1}", a, b);
        count++;
    } while (a != 6 || b != 6);
    Console.WriteLine("Виконано "+count.ToString()+" кидань");
    Console.ReadKey();
}
```

Лабораторна робота № 10

Тема. Програмування алгоритмів з використанням оператора циклу з параметром.

Мета. Формування вмінь і навиків програмування алгоритмів циклічної структури для перебору рівномірно розподілених значень. Закріплення вмінь і навиків використання функцій вводу-виводу. Застосування вмінь і навиків програмування алгоритмів лінійної та розгалуженої структури.

Контрольні запитання

1. Який порядок виконання оператора циклу з параметром?
2. Як перервати виконання чи перейти до наступної ітерації циклу з параметром?
3. Які значення може приймати величина кроку в циклі з параметром?
4. Чи виконуватиметься цикл з параметром, якщо при додатному значенні кроку початкове значення параметра циклу перевищує кінцеве значення?

Завдання.

1. Скласти програму обчислення значень функцій, починаючи з заданої точки у вказаній кількості точок.

Вимоги до програми:

- виконати ввід/вивід для у вікні консольного додатку;
- числові значення параметрів обчислень ввести з клавіатури;
- значення аргументів та відповідні їм значення результатів вивести на друк у вигляді таблиці.

Варіанти

1. $Y = \frac{\sin^2 x}{(x+1)^2}$ при
 $0.5 \leq x, \Delta x = 0.1, n = 9;$

2. $Y = \frac{e^{2x} - 8}{x+3}$ при
 $1.5 \leq x, \Delta x = 0.3, n = 6;$

3. $Y = \frac{1 + \cos 2x}{3x}$ при
 $1.2 \leq x, \Delta x = 0.2, n = 7;$

4. $Y = \frac{x + \sin 2xe^{2x} - 8}{x+2}$ при
 $0.6 \leq x, \Delta x = 1.5, n = 6;$

5. $Y = \frac{\cos^3 x^2}{1.5x+2}$ при
 $0.8 \leq x, \Delta x = 0.3, n = 5;$

6. $Y = \frac{x + \sin 2x}{x^2 - 3}$ при
 $3.1 \leq x, \Delta x = 0.8, n = 6;$
7. $Y = \frac{x^2 + 2}{3 \cos \sqrt{x} + 1}$ при
 $0.3 \leq x, \Delta x = 0.8, n = 7;$
8. $Y = \frac{x^3 - 3}{3 \ln x}$ при
 $2.5 \leq x, \Delta x = 1.5, n = 5;$
9. $Y = \frac{2 \sin^2(x+2)}{x^2 + 1}$ при
 $0.7 \leq x, \Delta x = 0.1, n = 5;$
10. $Y = \frac{(3x+2)^2}{\sin x + 3}$ при
 $0.2 \leq x, \Delta x = 0.7, n = 6;$
11. $Y = \frac{1.9x - \ln 3x}{3x + 1}$ при
 $0.8 \leq x, \Delta x = 1.2, n = 6;$
12. $Y = \frac{2x + 8}{|\cos 2.5x| + 1}$ при
 $0.1 \leq x, \Delta x = 0.3, n = 6;$
13. $Y = \frac{x - \ln 2x}{2x + 1}$ при
 $0.6 \leq x, \Delta x = 2.5, n = 5;$
14. $Y = \frac{x + \tan 2x}{3x}$ при
 $0.8 \leq x, \Delta x = 0.2, n = 6;$
15. $Y = \frac{(x+7)^2}{\sqrt{x^2 + 1}}$ при
 $6.5 \leq x, \Delta x = 0.3, n = 9.$

Наприклад, програма для розв'язання завдання № 3 може мати вигляд:

```

static void Main(string[] args)
{
    Double x, y, xp, dx;
    int i, n;
    Console.Write("Введіть x початкове: ");
    xp = Convert.ToDouble(Console.ReadLine());
    Console.Write("Введіть крок: ");
    dx = Convert.ToDouble(Console.ReadLine());
    Console.Write("Введіть кількість точок: ");
    n = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("x\ty");
    for (i = 0; i < n; i++)
    {
        x = xp + i * dx;
        y = (x + Math.Cos(2 * x)) / (3 * x);
        Console.WriteLine("{0}\t{1}", x, y);
    }
    Console.ReadKey();
}

```

2. Скласти програму обчислення значень функції, починаючи з заданої точки у вказаній кількості точок по варіантах першого завдання.

Вимоги до програми:

- виконати ввід/вивід у діалогових вікнах;
- числові значення параметрів обчислень ввести з клавіатури, константи з умов завдання використати як значення по замовчуванню;
- при введенні недопустимих значень чи типів параметрів обчислень передбачити виведення відповідного повідомлення та можливість завершення роботи програми чи повторного їх введення;
- значення аргументів та відповідні їм значення результатів вивести на друк у вигляді таблиці в одному діалоговому вікні.

3. Скласти програму для розв'язання поставленого завдання.

Вимоги до програми:

- числові значення параметрів обчислень ввести з клавіатури з відповідною перевіркою коректності типів та даних;
- при створенні програми використати оператор циклу з параметром;
- масиви не використовувати;
- результати обчислень вивести в одному діалоговому вікні.

Варіанти

1. Є перші N натуральних чисел. Визначити середнє арифметичне парних значень і середнє арифметичне непарних значень;
2. Підрахувати суму всіх натуральних чисел, що діляться на 3 і не перевищують числа N;
3. Обчислити середнє геометричне елементів натурального ряду від 1 до N (N не перевищує 12);

4. Піднести до степеня 9 число 3 не використовуючи безпосередню операцію піднесення до степеня.
5. Обчислити суму перших N чисел ряду $\frac{2}{1!} + \frac{3}{2!} + \frac{4}{3!} + \dots + \frac{N+1}{N!} + \dots$;
6. Знайти різницю між сумами двоцифрових чисел натурального ряду, розмішених на парних і на непарних місцях;
7. Обчислити суму $S = \sin 1^\circ + \sin 2^\circ + \sin 3^\circ + \dots + \sin 45^\circ$;
8. Підрахувати кількість натуральних чисел, кратних числу 6, що не перевищують числа N ;
9. Обчислити суму кубів двозначних непарних чисел;
10. Кожен член натурального ряду від 1 до 30 зменшити в три рази та обчислити добуток отриманих чисел;
11. Обчислити суму всіх трьохзначних чисел кратних числу 8;
12. Знайти суму перших N доданків ряду $\frac{1}{\sqrt{2^3}} + \frac{1}{\sqrt{3^3}} + \frac{1}{\sqrt{4^3}} + \dots + \frac{1}{\sqrt{(N+1)^3}} + \dots$;
13. Обчислити суму квадратів чисел -8, -6, -4, ..., 4, 6, 8;
14. Обчислити середнє геометричне всіх непарних чисел від 10 до 90;
15. Знайти суму цілочисельних координат точок, що належать відрізьку $[-3.5; 12,8]$.

Наприклад, програма для розв'язання завдання № 4 може мати вигляд:

```
static void Main(string[] args)
{
    int i, b;
    double a, dob;
    string s="";
    a = 3;
    b = 9;
    povtor:
    if (!inputDouble(ref a, "Введіть основу"))
        return;
    if (!inputInt(ref b, "Введіть степінь"))
        return;
    dob = 1;
    for(i=1; i<=b; i++)
        dob*=a;
    if (MessageBox.Show(a.ToString()+"^"+ b.ToString() + "=" + dob.ToString()+"\n\nБажаєте рахувати ще?",
        "Результати", MessageBoxButtons.YesNo, MessageBoxIcon.Information) == DialogResult.Yes)
        goto povtor;
}
```

Лабораторна робота № 11

Тема. Застосування перелічувального циклу. Вкладені цикли та оператори розгалуження.

Мета. Формування вмінь і навиків програмування алгоритмів вкладеної циклічної структури. Закріплення вмінь і навиків використання функцій вводу-виводу. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання

1. Які цикли називаються вкладеними?
2. Яка умова накладається на назви параметрів вкладених циклів?
3. Скільки разів виконується тіло вкладеного циклу?
4. Поясніть дію груп операторів
 - 4.1.

```
For (i=1; i<=5; i++)  
  For (j=2; j<=4; j++)  
  {  
    ...  
  };
```
 - 4.2.

```
For (i=1; i<=5; i++)  
{For (j=2; j<=4; j++)  
  {  
    ...  
  }  
  ...  
  For (j=1; j<=7; j++)  
  {  
    ...  
  }  
}
```
5. Як перервати виконання вкладеного та основного циклу?
6. Чим відрізняється функціонування перелічувального циклу від циклу з параметрами?

Завдання

1. Скласти програму для обробки символів текстового рядка з використанням перелічувального циклу.

Вимоги до програми:

- значення текстового рядка для обробки ввести з вікна консольного додатку;
- для обробки рядків використати перелічувальний оператор циклу;
- результати виконання алгоритму вивести у вікно консольного додатку.

Варіанти

1. У текстовому рядку підрахувати кількість букв 'о';
2. Переписати текстовий рядок навпаки;
3. Подвоїти у текстовому рядку букву 'к';

4. Встановити, чи у текстовому рядку букв 'п' більше ніж букв 'р';
5. Визначити позиції входження у текстовий рядок букви 'С';
6. Визначити кількість входжень у текстовий рядок букви 'н';
7. Видалити з текстового рядка букву 'к';
8. Вставити у текстовий рядок після кожної букви знак питання;
9. У текстовому рядку підрахувати кількість букв 'м';
10. Подвоїти у текстовому рядку букву 'л';
11. Встановити, чи у текстовому рядку букв 'ч' більше ніж букв 'ш';
12. Визначити позиції входження у текстовий рядок букви 'Ф';
13. Визначити кількість входжень у текстовий рядок букви 'к';
14. Видалити з текстового рядка букву 'д';
15. Вставити у текстовий рядок після кожної букви знак оклику.

Наприклад, програма для розв'язання завдання варіанту № 2 може бути такою:

```
static void Main(string[] args)
{
    string str, rez = "";
    Console.WriteLine("Введіть рядок для опрацювання: ");
    str=Console.ReadLine();
    foreach (char c in str)
        rez = c + rez;
    Console.WriteLine ("Рядок навпаки: " + rez);
    Console.ReadKey();
}
```

2. Скласти програму для розв'язання поставленого завдання.

Вимоги до програми:

- числові значення параметрів обчислень ввести з клавіатури з відповідною перевіркою коректності типів даних;
- кількість елементів послідовності ввести з клавіатури;
- при створенні програми використати оператор циклу з параметром;
- масиви не використовувати;
- результати обчислень вивести в одному діалоговому вікні.

Варіанти

Дано послідовність значень стипендій, отриманих студентом за період. ...

1. Підрахувати різницю між мінімальною та максимальною отриманою стипендією;
2. Обчислити загальну суму отриманих коштів;
3. Визначити середню стипендію;
4. Обчислити кількість отриманих мінімальних стипендій;
5. Знайти кількість отриманих максимальних стипендій;
6. Визначити кількість місяців, в яких студент не отримував стипендію;
7. Знайти кількість місяців, в яких студент отримував стипендію;
8. Обчислити кількість місяців, в яких студент отримував стипендію понад 50 грн.;
9. Знайти кількість місяців, в яких студент отримував стипендію до 800 грн.;
10. Визначити відхилення середньої стипендії від мінімальної;

11. Знайти відхилення максимальної стипендії від середньої;
12. Обчислити кількість місяців, в яких студент отримував стипендію від 500 до 700 грн.;
13. Визначити кількість місяців, в яких студент отримував стипендію до 500 або понад 700 грн.;
14. Знайти кількість місяців, в яких студент отримував стипендію, кратну 100 грн.;
15. Визначити, у скільки разів максимальна з отриманих стипендій перевищує мінімальну.

Наприклад, програма для розв'язання завдання варіанту № 1 може бути такою:

```
static void Main(string[] args)
{
    int n = 10; //значення кількості стипендій по замовчуванню
    Double St = 700, MinSt = 0, MaxSt = 0;
    int i;
    if (!inputInt(ref n, "Введіть кількість стипендій"))
        return;
    for (i = 1; i <= n; i++)
    {
        if (!inputDouble(ref St, "Введіть " + i.ToString() + "-у стипендію"))
            return;
        if (i == 1)
        {
            MinSt = St;
            MaxSt = St;
        }
        else
        {
            if (St < MinSt)
                MinSt = St;
            if (St > MaxSt)
                MaxSt = St;
        }
    }
    MessageBox.Show("Різниця між мінімальною та максимальною стипендіями становить "
        + (MaxSt-MinSt).ToString() + " грн.", "Результати обчислень",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
}
}
```

3. Скласти програму для розв'язання поставленого завдання (додатково 2 бали до рейтингу).

Вимоги до програми:

- числові значення параметрів обчислень ввести з клавіатури з відповідною перевіркою коректності типів та даних;
- при створенні програми використати вкладені цикли;
- масиви та підпрограми не використовувати;
- результати обчислень вивести в одному діалоговому вікні.

Варіанти

1. Знайти натуральне число в межах від 1 до 10000 з максимальною сумою дільників;
2. Дано натуральне число N . Отримати всі менші за N натуральні числа, що є з ним взаємно прості;
3. Дано цілі числа P та Q . Отримати всі дільники числа Q взаємно прості з P ;
4. Дано натуральне число N . Отримати всі прості дільники цього числа;
5. Знайти найменше натуральне число N , яке можна подати двома різними способами у вигляді $N=x^3+y^3$ ($x > y$);
6. Дано натуральні числа A та B ($A < B$). Отримати всі прості числа P з проміжку $[A; B]$;
7. Знайти сто перших простих чисел;
8. Дано натуральні числа N та M . Отримати всі менші N натуральні числа, квадрат суми цифр яких рівний M ;
9. Дано натуральне число N . Отримати всі менші за N досконалі числа. (Натуральне число називається досконалим, якщо воно рівне сумі всіх своїх дільників, за винятком самого себе. Наприклад, число 6 – досконале, оскільки $6=1+2+3$, а число 8 – недосконале, оскільки $8 \neq 1+2+4$);
10. Дано натуральне число N . Вияснити, чи можна подати його у вигляді суми квадратів трьох натуральних чисел. Якщо можна, то вказати таку трійку x, y, z натуральних чисел, що $N = x^2 + y^2 + z^2$;
11. В інтервалі від 1 до 1000 знайти всі пари простих чисел. (Парою простих чисел називаються два простих числа, різниця між якими рівна 2, наприклад: 3 і 5, 11 і 13, 17 і 19);
12. Відомо, що будь-яке натуральне число P ($P > 7$) можна подати у вигляді $P=A^3+B^5$. Знайти всі пари чисел A та B для числа P ;
13. Дано натуральне число P ($P < 100000$). Отримати всі дружні числа в інтервалі від 1 до P . (Дружніми числами називається така пара натуральних чисел M і N , для яких сума всіх дільників числа M (крім самого числа M) рівна N , а сума всіх дільників числа N (крім самого числа N) рівна M . Наприклад, числа 220 і 284 дружні, оскільки сума дільників числа 220 ($1+2+4+5+10+11+20+22+44+55+110$) рівна 284, а сума дільників числа 284 ($1+2+4+7+142$) рівна 220);
14. Число Армстронга – це таке число із K цифр, для якого сума K -х степенів його цифр рівна самому числу. Наприклад: $153=1^3+5^3+3^3$. Знайти всі числа Армстронга з двох, трьох, чотирьох цифр;
15. Дано натуральне число N . Вияснити, чи можна подати його у вигляді суми кубів двох натуральних чисел. Якщо можна, то вказати два таких натуральних числа x та y , що $N = x^3 + y^3$.

Наприклад, програма для розв'язання завдання варіанту № 9 може бути такою:

```

static void Main(string[] args)
{int n=100, chuslo, sumaD, dilnuk;
  if (!inputInt(ref n, "Введіть натуральне число N"))
    return;
  string REZ = ""; //поки досконалі числа не знайдені
  for (chuslo = 2; chuslo <= n; chuslo++)
  {sumaD = 1; //початкове значення суми дільників
    for (dilnuk = 2; dilnuk <= chuslo / 2; dilnuk++)
      if ((chuslo % dilnuk) == 0)
        sumaD += dilnuk;
    if (chuslo == sumaD)
      REZ += "\n"+chuslo.ToString();
  }
  string promigok = "Досконалі числа на проміжку [1;" + n.ToString() + "]";
  if (REZ == "")
    REZ=promigok+" відсутні";
  else
    REZ=promigok+": " + REZ;
  MessageBox.Show(REZ, "Результати", MessageBoxButtons.OK, MessageBoxIcon.Information);
}

```

З питань тиражування та використання цього видання звертайтеся.

Лабораторна робота № 12

Тема. Програмування алгоритмів обробки одновимірних масивів.

Мета. Формування вмінь і навиків програмування алгоритмів обробки одновимірних масивів. Закріплення вмінь і навиків використання функцій вводу-виводу. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання

1. Як описується масив у програмі? Як вказується тип елементів при оголошенні масиву?
2. До чого призводить під час виконання програми відсутність попереднього опису масиву і в яких випадках?
3. Як функціонують алгоритми знаходження в масиві елемента та його індексу, що має задані властивості?
4. Як функціонують алгоритми пошуку та сортування елементів в масиві?
5. Як у впорядкований масив дописати новий елемент без порушення впорядкування?

Завдання

1. **Скласти програму формування та друку масиву з 10 цілочисельних елементів проміжку $[a,b]$ і визначення їх суми.**

Вимоги до програма:

- числові значення a та b ввести за допомогою оператора присвоєння;
- масив формувати за допомогою оператора присвоєння з використанням датчик випадкових чисел;
- результати формування вивести в одному діалоговому вікні, вказавши для кожного елемента масиву його індекс та значення.

Варіанти:

- | | | |
|-------------------|--------------------|---------------------|
| 1. $a=10, b=40;$ | 6. $a=15, b=300;$ | 11. $a=15, b=80;$ |
| 2. $a=4, b=18;$ | 7. $a=200, b=290;$ | 12. $a=200, b=400;$ |
| 3. $a=30, b=100;$ | 8. $a=17, b=80;$ | 13. $a=127, b=800;$ |
| 4. $a=8, b=50;$ | 9. $a=37, b=85;$ | 14. $a=3, b=1000;$ |
| 5. $a=5, b=70;$ | 10. $a=7, b=56;$ | 15. $a=16, b=160.$ |

Наприклад, програма для розв'язання завдання варіанту № 15 може мати такий вигляд:

```

static void Main(string[] args)
{
    int i;
    double a=16, b=160;
    string s="";
    povtor:
    if (!inputDouble(ref a, "Введіть мінімальне значення"))
        return;
    if (!inputDouble(ref b, "Введіть максимальне значення"))
        return;
    int N=10;
    double[] mas;
    double suma;
    mas=new double[N];
    Random rnd = new Random();
    for(i=0; i<N; i++)
        mas[i]=a+rnd.NextDouble()*(b-a);
    suma=0;
    foreach(double k in mas)
        suma+=k;
    s = "";
    for(i=0; i<N; i++)
        s+=mas[i].ToString()+"="+mas[i].ToString()+Strings.Chr(13);
    if (MessageBox.Show(s + Strings.Chr(13) + "Сума: " + suma.ToString() + "\n\nГенерувати ще?", "Результати",
        MessageBoxButtons.YesNo)== DialogResult.Yes)
        goto povtor;
}

```

2. Скласти програму для розв'язання поставленого завдання.

Вимоги до програми:

- елементи послідовностей зберігати в одновимірних масивах;
- числові значення елементів масиву та параметри обчислень ввести з клавіатури з відповідною перевіркою коректності типів та даних;
- для обробки елементів масиву використати оператори циклу;
- результати обчислень вивести в одному діалоговому вікні.

Варіанти

1. Серед 10 чисел послідовності є додатні і від'ємні. Скласти програму формування нової послідовності, що містить квадратні кореня з додатних чисел і кубічні корені з недодатних чисел вихідної послідовності;
2. Дано п'ять пар значень змінних x та y . Відшукати пару $(x; y)$, при якій вираз $3x - 4y + 5$ приймає найбільше значення;
3. Знайти середнє значення 6 чисел;
4. Дано 7 дійових чисел. Надрукувати квадрати тих чисел, які за абсолютною величиною не перевищують числа M ;
5. Дано 6 дійсних чисел. Визначити синус суми модулів цих чисел;
6. Дано 5 дійсних чисел. Обчислити квадратний корінь з модуля їх добутку;
7. Дано 8 дійсних чисел. Обчислити суму квадратів різниць коренів з модулів чисел та самих чисел;
8. Дано 7 дійсних чисел. Обчислити подвоєний квадрат їх суми;
9. Дано 5 дійсних чисел. Обчислити суму квадратних коренів модулів цих чисел;

10. Дано послідовність з 7 цілих чисел. Обчислити суми кожного елемента послідовності з попередніми елементами;
11. Дано 8 дійсних чисел. Обчислити кубічні корені з квадратів цих чисел;
12. Дано 7 дійсних чисел і натуральне число n. Знайти середнє арифметичне цих чисел за виключенням елемента з індексом n;
13. Дано 9 дійсних чисел. Обчислити середнє геометричне тих елементів послідовності, що передують першому від'ємному числу;
14. Дано 8 цілих чисел, які є величинами градусної міри кутів. Знайти величини відповідних кутів в радіанах;
15. Дано 10 дійсних чисел. Обчислити їх суму та добуток;
16. Визначити всі відстані між чотирма точками площини, що задаються декартовими координатами.

Наприклад, програма для розв'язання завдання варіанту № 15 може виглядати так:

```
static void Main(string[] args)
{
    int i, n=10;
    if (!inputInt(ref n, "Введіть кількість елементів масиву"))
        return;
    Double[] mas = new double[n];
    for (i = 0; i < n; i++)
        if (!inputDouble(ref mas[i], "Введіть " + i.ToString() + "-й елемент масиву"))
            return;
    Double Suma, Dobutok;
    Suma = 0;
    Dobutok = 1;
    String rez;
    for (i = 0; i < n; i++)
    {
        Suma += mas[i];
        Dobutok *= mas[i];
    }
    rez = "Для масиву" + Strings.Chr(13);
    for (i = 0; i < n; i++)
        rez += "mas[" + i.ToString() + "] = " + mas[i].ToString() + Strings.Chr(13);
    rez += "Добуток елементів рівний: " + Dobutok.ToString() + "\n" + "сума становить " + Suma.ToString();
    MessageBox.Show(rez, "Результати обчислень", MessageBoxButtons.OK);
}
```

3. Реалізувати алгоритм пошуку всіх позицій входження елемента в масив (додатково 2 бали до рейтингу).

При цьому фрагмент коду для пошуку позицій входження з консолі може бути, наприклад, такий:

```
Console.Write("Введіть кількість елементів масиву: ");
n=Convert.ToInt32(Console.ReadLine());
Double[] mas = new double[n];
for (i = 0; i < n; i++)
{Console.Write("Введіть " + i.ToString() + "-й елемент масиву: ");
mas[i]=Convert.ToDouble(Console.ReadLine());
}
```

```

double element=7;
res = "";
//реалізація послідовного пошуку
Console.WriteLine("Введіть елемент для пошуку: ");
element = Convert.ToDouble(Console.ReadLine());
for (i = 0; i < n; i++)
    if (mas[i] == element)
        {if (res != "")
            res += ", ";
            res += i.ToString();
        }
Console.WriteLine(res=="?"?"Елемент '"+element.ToString()+"' в масиві відсутній":
    "Позиції входження елемента '"+element.ToString()+"' в масив: "+res);
Console.ReadKey();

```

Реалізація такого ж пошуку (після введення масиву, як у задачі № 2) з діалогових вікон може бути, наприклад, такою:

```

double element=7;
res = "";
if (!inputDouble(ref element, "Введіть елемент для пошуку"))
    return;
for (i = 0; i < n; i++)
    if (mas[i] == element)
        {if (res != "")
            res += ", ";
            res += i.ToString();
        }
MessageBox.Show(res=="?"?"Елемент '"+element.ToString()+"' в масиві відсутній":
    "Позиції входження елемента '"+element.ToString()+"' в масив: "+res,
    "Результати обчислень", MessageBoxButtons.OK);

```

З питань тиражування та використання

Лабораторна робота № 13

Тема. Динамічні масиви

Мета. Формування вмінь і навиків використання динамічних одновимірних масивів. Закріплення вмінь і навиків використання функцій вводу-виводу та обробки одновимірних масивів. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання

1. Коли в програмах використовуються динамічні масиви? Чим вони відрізняються від статичних?
2. Як перевизначити розмір динамічного масиву в програмі? Який тип елементів використовується для перевизначеного масиву?
3. Скільки разів може змінюватися розмір динамічного масиву? Чи можливо змінити тип елемента масиву при перевизначенні?
4. Які значення елементів встановлюються для перевизначеного масиву?
5. Як звільнити пам'ять, що використовується динамічним масивом?

Завдання

1. Скласти програму формування та друку масиву з N дійсних елементів проміжку $[a, b]$ згідно першого завдання попередньої лабораторної роботи. Після друку масиву збільшити його розмір на чотири елементи і надрукувати ще раз.

Вимоги до програма:

- числові значення N , a , та b ввести з клавіатури з відповідною перевіркою коректності типів та даних;
- масив визначити лише після введення кількості його елементів;
- масив формувати за допомогою оператора присвоєння з використанням датчик випадкових чисел;
- результати формування вивести в одному діалоговому вікні, вказавши для кожного елемента масиву його індекс та значення.

Наприклад, програма для розв'язання завдання варіанту № 2 може мати такий вигляд:

```
int i;
double a, b;
string s="";
a = 4; b = 18;
int N=10;
double[] mas;
povtor:
if (!inputInt(ref N, "Введіть кількість елементів масиву"))
    return;
mas=new double[N];
if (!inputDouble(ref a, "Введіть мінімальне значення"))
    return;
if (!inputDouble(ref b, "Введіть максимальне значення"))
    return;
```



```

Random rnd = new Random();
for(i=0; i<N; i++)
    mas[i]=a+rnd.NextDouble()*(b-a);
s = "Масив до перевизначення:\n";
for (i = 0; i < N; i++)
    s += "mas[" + i.ToString() + "]=" + mas[i].ToString() + "\n";
Array.Resize(ref mas, mas.Length +4);
s+= "\nМасив після перевизначення:\n";
for (i = 0; i < mas.Length; i++)
    s+="mas["+i.ToString()+"]="+mas[i].ToString()+ "\n";
if (MessageBox.Show(s + "\nГенерувати ще?", "Результати",
    MessageBoxButtons.YesNo,MessageBoxIcon.Information)== DialogResult.Yes)
    goto povtor;

```

2. Скласти програму для розв'язання поставленого завдання.

Вимоги до програми:

- числові значення розмірності N , цілочисельних елементів масиву та параметрів обчислень ввести з клавіатури з відповідною перевіркою коректності типів та даних;
- масив визначити лише після введення кількості його елементів;
- для обробки елементів масиву використати оператори циклу;
- результати обчислень вивести в одному діалоговому вікні у вигляді стовпця з елементів вхідного та створених масивів.

Варіанти

1. Знайти суму елементів одновимірного масиву. Для формування вихідного масиву домножити кожен парний елемент вхідного масиву на отримане значення;
2. Обчислити суми і різниці елементів двох заданих масивів з непарними індексами. Результат надрукувати у вигляді паралельних стовпців;
3. Задано масив X . Обчислити значення функції $y=0,5\ln x$ при непарних значеннях аргумента, заданих в масиві X , і помістити їх в масив Y . Надрукувати елементи масивів X та Y у вигляді паралельних стовпців;
4. Знайти середнє значення елементів заданого масиву. Перетворити вхідний масив, віднімаючи від кожного парного елемента отримане середнє значення;
5. Визначити максимальне значення елементів масиву. Перетворити вхідний масив, віднімаючи від кожного непарного елемента отримане максимальне значення;
6. Сформувані два масиви, включаючи в перший елементи вхідного масиву з парними індексами, а в другий - з непарними;
7. Ввести непарні числа масиву X у зворотній послідовності;
8. Потроїти в масиві всі парні елементи;
9. Визначити елементи масиву, що є подвоєними непарними числами;
10. Визначити елементи масиву, що діляться на 5 і не діляться на 7;
11. Зберегти у вихідному масиві всі непарні елементи;
12. Визначити індекс елемента масиву рівного A . Розділити парні елементи вхідного масиву на цей індекс, а при його відсутності – на 5;

13. Перевірити істинність твердження про те, що кількість від'ємних елементів масиву перевищує кількість додатних. Вивести додатні і від'ємні елементи у вихідні масиви;
14. Знайти квадрати парних елементів масиву;
15. Вхідний масив містить лише значення 0 та 1. Визначити кількість нулів і одиниць в масиві. Поміняти 0 на 1 і навпаки, не застосовуючи оператор розгалуження. Записати у вихідний масив позиції значень 0.

Наприклад, програма для розв'язання завдання варіанту № 1 може бути такою:

```
int i, n=10;
if (!inputInt(ref n, "Введіть кількість елементів масиву"))
    return;
int[] mas = new int[n], output=new int[n];
for (i = 0; i < n; i++)
    mas[i] = i; //ініціалізація масиву
for (i = 0; i < n; i++)
    if (!inputInt(ref mas[i], "Введіть " + i.ToString() + "-й елемент масиву"))
        return;
int suma = 0;
for (i = 0; i < n; i++)
    suma += mas[i];
for (i = 0; i < n; i++)
    if (mas[i] % 2 == 0)
        output[i] = mas[i] * suma;
    else
        output[i] = mas[i];
String rez="Вхідний масив:\n";
for (i = 0; i < n; i++)
    rez += "mas[" + i.ToString() + "]= " + mas[i].ToString() + "\n";
rez += "\nВихідний масив:\n";
for (i = 0; i < mas.Length; i++)
    rez += "output[" + i.ToString() + "]= " + output[i].ToString() + "\n";
MessageBox.Show(rez, "Результати", MessageBoxButtons.OK, MessageBoxIcon.Information);
```

3. Реалізувати алгоритм швидкого пошуку хоча б однієї позиції входження елемента у відсортований масив (додатково 1 бал до рейтингу) або пошуку всіх позицій входження елемента у відсортований масив (додатково 2 бали до рейтингу).

Вимоги до програми:

- числові значення розмірності N , дійсних елементів масиву та елемента для пошуку ввести з клавіатури з відповідною перевіркою коректності типів та даних;
- масив визначити лише після введення кількості його елементів;
- для обробки елементів масиву використати оператори циклу;
- результати обчислень вивести одним оператором.

При цьому фрагмент коду для **бінарного** пошуку позиції входження з консолі може бути, наприклад, такий (введення масиву та перевірку коректності введених даних забезпечте самостійно):

```

//реалізація бінарного пошуку
Console.Write("Введіть елемент для пошуку");
element=Convert.ToDouble(Console.ReadLine());
Array.Sort(mas);
i = 0; j = n - 1; //початкові межі пошуку
while (i<j)
{avg = (i + j) / 2;
  if (mas[avg] == element)
  {i=avg;
   break;
  }
  if (mas[avg] > element)
  j = avg - 1;
  else
  i = avg + 1;
}
Console.WriteLine("Елемент '" + element.ToString() + "' +
                  (mas[i]!=element ? "в масиві відсутній" :
                  "міститься в масиві в позиції " + i.ToString()));

```

— shporiko@ukr.net

А фрагмент коду для **бінарного** пошуку позиції входження та визначення інтервалу входження з діалогових вікон може бути, наприклад, такий (введення масиву та перевірку коректності введених даних забезпечте самостійно):

```

Array.Sort(IS);
double element=IS[0];
element = Convert.ToDouble(Interaction.InputBox("Введіть елемент для пошуку:",
                                                "Введення", IS[0].ToString()));

int i = 0, j = IS.Length - 1, avg;
do
{
  avg = (i + j) / 2;
  if (IS[avg] == element)
  {
    j = avg;
    break;
  }
  if (element < IS[avg])
  j = avg-1;
  else
  i = avg+1;
} while (i < j);
string res = "", pos="";
if (IS[j] != element)
  pos = "в масиві відсутній";
else
{i = j;
 //пересуваємо i вліво
while (i > 0 && IS[i - 1] == element)
  i--;
 //пересуваємо j вправо
while (j < IS.Length - 1 && IS[j + 1] == element)
  j++;
if (i==j)
  pos="міститься в елементі масиву з індексом "+i.ToString();
else
  pos="міститься в елементах масиву з індексами від "+i.ToString() + " до "+j.ToString();
}
for (i = 0; i < IS.Length; i++)
  res += "IS[" + i.ToString() + "]=" + IS[i].ToString() + "\n";
res+="\nЕлемент '" +element.ToString()+"' " + pos;
MessageBox.Show("Масив:\n" + res, "Результати", MessageBoxButtons.OK, MessageBoxIcon.Information);

```

Лабораторна робота № 14

Тема. Сортування одновимірних масивів.

Мета. Формування вмінь і навиків сортування одновимірних масивів. Закріплення вмінь і навиків використання функцій вводу-виводу. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання

1. На які два типи поділяються алгоритми сортування по швидкодії? До яких масивів доцільно застосовувати алгоритми кожного типу?
2. На які три категорії поділяються алгоритми сортування без допоміжних структур?
3. Який принцип закладено в основу сортування алгоритмом бульбашок? Прямого включення? Прямого вибору? Швидкого сортування обміном на великих відстанях?
4. Вкажіть переваги та недоліки по швидкості виконання методу прямого включення відносно методу прямого вибору.
5. Від чого залежить швидкість виконання алгоритмів сортування?

Завдання

1. **Скласти програму формування та друку масиву з 20000 дійсних елементів проміжку [a,b]. Відсортувати сформований масив та визначити час сортування методом бульбашок. Вивести сформований масив та час його сортування.**

Вимоги до програма:

- числові значення a та b ввести за допомогою оператора присвоєння;
- масив формувати за допомогою оператора присвоєння з використанням датчик випадкових чисел;
- результати формування вивести в одному діалоговому вікні, вказавши для кожного елемента масиву його індекс та значення.

Варіанти:

- | | | |
|--------------------|---------------------|----------------------|
| 1. $a=10, b=40$; | 6. $a=15, b=300$; | 11. $a=15, b=80$; |
| 2. $a=4, b=18$; | 7. $a=200, b=290$; | 12. $a=200, b=400$; |
| 3. $a=30, b=100$; | 8. $a=17, b=80$; | 13. $a=127, b=800$; |
| 4. $a=8, b=50$; | 9. $a=37, b=85$; | 14. $a=3, b=1000$; |
| 5. $a=5, b=70$; | 10. $a=7, b=56$; | 15. $a=16, b=160$. |

Наприклад, програма для розв'язання завдання варіанту № 8 може мати такий вигляд:

```

static void generateMas(double[] mas, int N, double a, double b)
{
    Random rnd = new Random();
    for (int i = 0; i < N; i++)
        mas[i] = a + rnd.NextDouble() * (b - a);
}

static void printMas(double[] mas, int N, string s)
{
    Console.WriteLine(s);
    for (int i = 0; i < N; i++)
        Console.WriteLine("mas[" + i.ToString() + "]= " + mas[i].ToString());
}

static void Main(string[] args)
{
    const double a = 17, b = 80;
    int N, i, j, k, indexMin;
    Console.Write("Введіть кількість елементів масиву: ");
    N = Convert.ToInt32(Console.ReadLine());
    double[] mas = new double[N];
    double element, min;

    generateMas(mas, N, a, b);
    printMas(mas, N, "Початковий масив");
    Console.ReadKey();
    //метод бульбашок
    DateTime startTime = DateTime.Now;
    for (i = 0; i < N - 1; i++)
        for (j = 0; j < N - i - 1; j++)
            if(mas[j]>mas[j+1])
                {element = mas[j];
                 mas[j] = mas[j + 1];
                 mas[j + 1] = element;
                }
    DateTime finishTime = DateTime.Now;
    printMas(mas, N, "\nВідсортований масив методом бульбашок:");
    Console.WriteLine("Тривалість сортування методом бульбашок: " +
        (finishTime - startTime).ToString());
    Console.ReadKey();
}

```

2. Розв'язати попереднє завдання для масиву цілочисельних елементів.
3. Розв'язати завдання № 1, відсортувавши масив методом *прямого включення*.

Фрагмент програми для розв'язання цього завдання (в продовження програми попереднього завдання) може виглядати так:

```
generateMas(mas, N, a, b);
printMas(mas, N, "Початковий масив");
Console.ReadKey();
//метод прямого включення
startTime = DateTime.Now;
for (i = 1; i < N; i++)
{
    j = 0;
    element = mas[i];
    while (j < i && mas[j] <= mas[i])
        j++;
    if (j < i)
    {
        for (k = i - 1; k >= j; k--)
            mas[k + 1] = mas[k];
        mas[j] = element;
    }
}
finishTime = DateTime.Now;
printMas(mas, N, "\nВідсортований масив методом прямого включення:");
Console.WriteLine("Тривалість сортування методом прямого включення: " +
    (finishTime - startTime).ToString());
Console.ReadKey();
```

4. Розв'язати завдання № 1, відсортувавши масив методом *прямого вибору*.
- Фрагмент програми для розв'язання цього завдання (в продовження програми попереднього завдання) може виглядати так:

```
generateMas(mas, N, a, b);
printMas(mas, N, "Початковий масив");
Console.ReadKey();
//метод прямого вибору
startTime = DateTime.Now;
```

З шк.

```

for (i = 0; i < N-1; i++)
{
    min = mas[i];
    indexMin = i;
    for (j = i + 1; j < N; j++)
        if (mas[j]<min)
            {min = mas[j];
             indexMin=j;
            }
    if (indexMin>i)
        {mas[indexMin] = mas[i];
         mas[i] = min;
        }
}
finishTime = DateTime.Now;
printMas(mas, N, "\nВідсортований масив методом прямого вибору:");
Console.WriteLine("Тривалість сортування методом прямого вибору: " +
                  (finishTime - startTime).ToString());
Console.ReadKey();

```

5. Розв'язати завдання № 1, відсортувавши масив методом *швидкого сортування обміном на великих відстанях*. Порівняти час сортування цим методом з часом сортування стандартним методом *Sort* класу *Array*.

Фрагмент програми для розв'язання цього завдання (в продовження програми попереднього завдання) може виглядати так:

```

generateMas(mas, N, a, b);
printMas(mas, N, "Початковий масив");
Console.ReadKey();
//метод швидкого сортування обміном на великих відстанях
startTime = DateTime.Now;
quickSort(mas, 0, N - 1);
finishTime = DateTime.Now;
printMas(mas, N, "\nВідсортований масив обміном на великих відстанях:");
Console.WriteLine("Тривалість швидкого сортування обміном на великих відс
                  (finishTime - startTime).ToString());
Console.ReadKey();

```

```

generateMas(mas, N, a, b);
printMas(mas, N, "Початковий масив");
Console.ReadKey();
//стандартний метод сортування
startTime = DateTime.Now;
Array.Sort(mas);
finishTime = DateTime.Now;
printMas(mas, N, "\nВідсортований масив");
Console.WriteLine("Тривалість стандартного сортування: " +
    (finishTime - startTime).ToString());
Console.ReadKey();

```

А сама рекурсивна функція швидкого сортування може бути такою:

```

static void quickSort(double[] mas,int i,int j)
{
    if (i >= j)
        return;
    double element = mas[i];
    int i1 = i; int j1 = j;
    while(i1<j1)
    {while (j1 > i1 && mas[j1] >= element)
        j1--;
        if(i1<j1)
        {mas[i1] = mas[j1];
            mas[j1] = element;
            i1++;
        }
        while (i1 < j1 && mas[i1] <= element)
            i1++;
        if (i1 < j1)
        {mas[j1] = mas[i1];
            mas[i1] = element;
            j1--;
        }
    }
    quickSort(mas, i, i1 - 1);
    quickSort(mas, i1 + 1, j);
}
}

```


Лабораторна робота № 15

Тема. Двовимірні масиви. Обробка багатовимірних масивів.
Мета. Формування вмінь і навиків програмування алгоритмів обробки двовимірних масивів. Закріплення вмінь і навиків використання функцій вводу-виводу та обробки одновимірних масивів. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання

1. Як описується двовимірний масив у програмі?
2. Як розміщуються елементи двовимірного масиву в пам'яті комп'ютера?
3. Яка максимально можлива розмірність масиву? Скільки розмірностей може мати масив?
4. Як вивести двовимірний масив у вигляді матриці?
5. Які комбінації операторів найчастіше використовуються для обробки двовимірних масивів? Чому?

Завдання

1. **Скласти програму для розв'язання поставленого завдання (максимум 1 бал до рейтингу).**

Вимоги до програми:

- числові значення розмірностей масиву $M \times N$ ввести з вікна консольного додатку на початку програми;
- дійсні числові значення елементів масиву згенерувати за допомогою датчика випадкових чисел;
- для обробки елементів масиву використати вкладені оператори циклу;
- результати обчислень вивести у вікні консольного додатку у вигляді матриці та результату, розділивши їх порожнім рядком.

Варіанти

1. Знайти максимальний елемент масиву.
2. Підрахувати кількість десятків у масиві.
3. Визначити індекси та значення мінімального елемента масиву.
4. Знайти суму елементів масиву.
5. Підрахувати кількість нулів у масиві.
6. Визначити мінімальний елемент масиву.
7. Визначити індекси максимального елемента масиву.
8. Визначити добуток елементів масиву.
9. Визначити середнє арифметичне елементів масиву.
10. Обчислити добуток мінімального та максимального елементів масиву.
11. Обчислити середнє геометричне елементів масиву.
12. Визначити кількість від'ємних значень в масиві.
13. Підрахувати суму діагональних елементів масиву.
14. Підрахувати добуток додатних елементів масиву.
15. Визначити кількість максимальних елементів масиву.

Наприклад, програма для розв'язання завдання варіанту № 8 може мати такий вигляд:

```
static void generateMas(double[,] mas, int N, int M, double a, double b)
{
    Random rnd = new Random();
    for (int i = 0; i < N; i++)
        for (int j = 0; j < M; j++)
            mas[i,j] = a + rnd.NextDouble() * (b - a);
}

static void printMas(double[,] mas, int N, int M, string s)
{
    Console.WriteLine(s);
    for (int i = 0; i < N; i++)
        {for (int j = 0; j < M; j++)
            Console.Write("{0,7:N2}\t",mas[i, j]);
        Console.WriteLine();
    }
}

static void Main(string[] args)
{
    const double a = 125, b = 13430;
    int N, M, i, j;
    Console.Write("Введіть кількість рядків масиву: ");
    N = Convert.ToInt32(Console.ReadLine());
    Console.Write("Введіть кількість стовпців масиву: ");
    M = Convert.ToInt32(Console.ReadLine());
    double[,] mas = new double[N,M];
    generateMas(mas, N, M, a, b);
    printMas(mas, N, M, "Початковий масив");
    double min = mas[0, 0], indexI = 0, indexJ=0;
    for (i = 0; i < N; i++)
        for (j = 0; j < M; j++)
            if (mas[i,j]<min)
                {min = mas[i, j];
                 indexI = i;
                 indexJ = j;
                }
    Console.WriteLine("\nМінімальне значення масиву " + min.ToString() +
        " вперше зустрілося в позиції [" + indexI.ToString() + "," +
        indexJ.ToString() + "]\n");
}
```

```
Console.ReadKey();
```

```
}
```

2. Скласти програму для розв'язання поставленого завдання (максимум 2 бали до рейтингу).

Вимоги до програми:

- числові значення розмірностей масивів задати константами на початку програми;
- числові значення елементів масиву та параметрів обчислень ввести з клавіатури з відповідною перевіркою коректності типів та даних;
- для обробки елементів масиву використати вкладені оператори циклу;
- результати обчислень вивести в одному діалоговому вікні у вигляді матриць, розділивши вхідний та вихідний масив порожнім рядком.

Варіанти

1. Знайти мінімальне значення вхідного масиву розмірності $M \times N$. Отримати вихідний масив, замінивши всі мінімальні значення нулями;
2. Обчислити суми елементів кожного стовпця заданої матриці розмірності $M \times N$. Результат занести в одновимірний масив розмірності N ;
3. Визначити середнє арифметичне елементів масиву розмірності $M \times N$. Знайти індекси елемента масиву, найближчого до отриманого середнього значення;
4. Задана матриця розмірності 5×5 та число K ($K \leq 5$). Розділити елементи K -го рядка на діагональний елемент, розміщений у цьому ж рядку;
5. Задана матриця A , розмірності 4×4 , числа K та L ($K \neq L, 1 \leq K \leq 4, 1 \leq L \leq 4$). З L -го рядка відняти K -ий, помножений на a_{lk} / a_{kk} ;
6. Задана дійсна матриця розмірності $M \times N$, у якій не всі елементи рівні нулю. Отримати нову матрицю шляхом ділення всіх елементів даної матриці на її найбільший за модулем елемент;
7. Дана матриця розмірності $M \times N$. Знайти суму найбільших значень елементів її рядків;
8. Дана дійсна матриця розмірності $M \times N$. Отримати послідовність кількостей від'ємних елементів у кожному рядку. Результат занести в одновимірний масив розмірності M ;
9. Дана дійсна матриця розмірності $M \times N$. Отримати послідовність добутків квадратів тих елементів кожного рядка, модулі яких належать відріzkу $[1;3]$. Результат занести в одновимірний масив розмірності M ;
10. Дана дійсна матриця розмірності $M \times N$. Знайти середнє арифметичне кожного з рядків, що мають парні номери;
11. Всі елементи з найбільшим значенням в даній цілочисельній квадратній матриці порядку 8 замінити нулями;
12. Дана дійсна матриця розмірності 4×6 . Знайти середнє арифметичне найбільшого і найменшого значень її елементів;
13. В даній дійсній квадратній матриці порядку N знайти суму елементів першого рядка, в якому розміщений елемент з найменшим значенням;

14. Отримати дійсну матрицю A порядку 5, перший та другий рядки якої задаються, відповідно, формулами

$$a_{1j} = 2j + 3, \quad a_{2j} = j - \frac{3}{2+j}, \quad (j=1,2,\dots,5),$$

а елементи кожного наступного рядка рівні сумі відповідних елементів двох попередніх рядків.

15. Дано дві дійсні квадратні матриці порядку 4. Отримати нову матрицю множенням елементів кожного рядка першої матриці на найбільше із значень елементів відповідного рядка другої матриці.

Наприклад, додатковий фрагмент програма до наведеного вище коду для розв'язання завдання варіанту № 1 може мати такий вигляд:

```
for (i = 0; i < N; i++)
    for (j = 0; j < M; j++)
        if (mas[i, j] == min)
            mas[i, j]=0;
printMas(mas, N, M, "Перетворений массив");
Console.ReadKey();
```

3. Скласти програму для розв'язання поставленого завдання (максимум 2 бали до рейтингу).

Вимоги до програми:

- числові значення розмірностей масивів та параметрів обчислень ввести з клавіатури з відповідною перевіркою коректності типів та даних;
- елементи масиву задати за допомогою датчика випадкових чисел;
- для обробки елементів масиву використати вкладені оператори циклу;
- результати обчислень вивести в одному діалоговому вікні у вигляді матриць, розділивши вхідний та вихідний масив порожнім рядком.

Варіанти

1. Задано одновимірний масив X розмірності N . Сформувати з нього матрицю A , що містить по L елементів у рядку. Якщо в останньому рядку не вистачатиме елементів, то доповнити його нулями;
2. Задана матриця A розмірності $N \times N$. Сформувати два одновимірні масиви. В перший переслати по рядках верхній трикутник вхідної матриці, включаючи елементи головної діагоналі, а в другий – нижній трикутник.
3. Задана квадратна матриця. Переставити рядок з максимальним елементом головної діагоналі з рядком із заданим номером.
4. Задані матриця розмірності $N \times N$ і ціле число K з проміжку $[1;N]$. Рядок з максимальним за модулем елементом в K -му стовпці переставити з K -м рядком;
5. Задана квадратна матриця. Видалити з неї рядок і стовпчик, на перетині яких розміщений максимальний елемент головної діагоналі;
6. Знайти добуток двох заданих матриць;

7. Використовуючи дії множення (ділення) рядка на число та додавання рядків, привести задану квадратну матрицю до такого виду, щоб всі елементи нижче головної діагоналі були нульовими;
8. Перший рядок дійсної матриці розмірності $2 \times N$ містить координати x , а другий – координати y кожної з N точок. Точки попарно з'єднані відрізками. Знайти довжину найбільшого відрізка;
9. Дана цілочисельна квадратна матриця порядку 8. Знайти найменше із значень елементів її першого стовпця, що містить найбільшу суму модулів елементів;
10. Дана цілочисельна квадратна матриця порядку N . Знайти номери рядків, елементи яких утворюють симетричні послідовності (паліндроми);
11. Дана цілочисельна квадратна матриця порядку N . Отримати суми елементів, розміщених за першим від'ємним елементом кожного рядка. При відсутності від'ємних елементів у рядку значення відповідної суми покласти рівним нулю;
12. Латинським квадратом порядку N називається квадратна таблиця розмірності $N \times N$, кожен рядок і кожен стовпець якої містить всі з чисел $1, 2, \dots, N$. Дана цілочисельна квадратна матриця порядку 5. Визначити, чи є вона латинським квадратом;
13. Елемент матриці називається сідловою точкою, якщо він є одночасно найменшим у своєму рядку і найбільшим у своєму стовпці. Дана дійсна матриця розмірності 5×6 . Вияснити, чи є сідлові точки у цій матриці. Якщо є, то вказати індекси однієї з них;
14. Дано дійсне число x . Отримати квадратну матрицю порядку 10 вигляду

$$\begin{pmatrix} 1 & x & \dots & x^8 & x^9 \\ x & 0 & \dots & 0 & x^8 \\ \vdots & \vdots & & \vdots & \vdots \\ x^8 & 0 & & 0 & x^7 \\ x^9 & x^8 & \dots & x^7 & 1 \end{pmatrix}$$

середина якої заповнюється нулями;

15. Дана цілочисельна матриця розмірності 6×9 . Сформувати матрицю, яка отримується з даної при перестановці першого стовпця з дев'ятим, другого – з восьмим, третього – з сьомим, а четвертого – з шостим.

Наприклад, програма для розв'язання завдання варіанту № 6 може бути такою:

```
static void inputArray(double[,] mas, string povidom)
{for(int j=0; j<=mas.GetUpperBound(0); j++)
  for (int i=0; i<=mas.GetUpperBound(1); i++)
    mas[j, i] = Convert.ToDouble(Interaction.InputBox(povidom+"["+j+1).ToString()+
      ", "+(i+1).ToString()+"]:", "Введення", mas[j,i].ToString()));
}
```

```

static string formResArray(double[,] mas, string povidom)
{string res=povidom+"\n";
  for (int j = 0; j <= mas.GetUpperBound(0); j++)
    {for (int i = 0; i <= mas.GetUpperBound(1); i++)
      res += mas[j, i].ToString() + "\t";
      res += "\n";
    }
  return res;
}
static void Main(string[] args)
{
  int n = 3, m = 4, q = 5;
  n = Convert.ToInt32(Interaction.InputBox("Введіть кількість рядків першої матриці",
    "Введення", n.ToString()));
  m = Convert.ToInt32(Interaction.InputBox("Введіть кількість стовпців першої матриці",
    "Введення", m.ToString()));
  q = Convert.ToInt32(Interaction.InputBox("Введіть кількість стовпців другої матриці",
    "Введення", q.ToString()));

  double[,] a = new double[n, m];
  double[,] b = new double[m, q];
  double[,] c = new double[n, q];
  inputArray(a, "Введіть a");
  inputArray(b, "Введіть b");
  for (int i=0; i<n; i++)
    for (int j=0; j<q; j++)
      { c[i, j] = 0;
        for (int k = 0; k < m; k++)
          c[i, j] += a[i, k] * b[k, j]; }
  string res = formResArray(a, "При множенні матриці A") +
    formResArray(b, "\nна матрицю B") +
    formResArray(c, "\nотримано матрицю C");
  MessageBox.Show(res);
}

```

З шитань

Лабораторна робота № 16

- Тема.** Програмування алгоритмів обробки текстових рядків та слів.
Мета. Формування вмінь і навиків програмування алгоритмів обробки текстових рядків. Закріплення вмінь і навиків використання функцій вводу-виводу. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання

1. Які типи даних використовуються для зберігання текстових рядків і символів в C#?
2. Як записується операція поєднання текстових рядків? Чим ця операція відрізняється від операції додавання чисел?
3. Як записується мовою C# функція для визначення довжини рядка? Скільки разів доцільно використовувати цю функцію в програмі і чому?
4. Що таке порожній рядок? Як він записується в програмах? Яка його довжина?
5. Як виконати арифметичну операцію з числом, записаним у вигляді рядка символів? Як перевести число в текстовий рядок?

Завдання

1. Скласти програму для обробки текстового рядка.

Вимоги до програми:

- значення текстового рядка для обробки ввести з вікна консольного додатку;
- для обробки рядків використати оператори циклу;
- результати виконання алгоритму вивести у вікно консольного додатку.

Варіанти

1. У текстовому рядку підрахувати кількість букв 'o';
2. Переписати текстовий рядок навпаки;
3. Подвоїти у текстовому рядку букву 'k';
4. Встановити, чи у текстовому рядку букв 'п' більше ніж букв 'р';
5. Визначити позиції входження у текстовий рядок букви 'С';
6. Визначити кількість входжень у текстовий рядок буквосполучення "nn";
7. Видалити з текстового рядка букву 'к';
8. Вставити у текстовий рядок після кожної букви знак питання;
9. У текстовому рядку підрахувати кількість букв 'm';
10. Подвоїти у текстовому рядку букву 'l';
11. Встановити, чи у текстовому рядку букв 'ч' більше ніж букв 'ш';
12. Визначити позиції входження у текстовий рядок букви 'Ф';
13. Визначити кількість входжень у текстовий рядок буквосполучення "kk";
14. Видалити з текстового рядка букву 'д';
15. Вставити у текстовий рядок після кожної букви знак оклику.

Наприклад, програма для розв'язання завдання варіанту № 2 може бути такою:

```

String S1, S2;
Console.WriteLine("Ведіть рядок для обробки: ");
S1 = Console.ReadLine();
S2 = "";
int len = S1.Length;
for (int i = len - 1; i >= 0; i--)
    S2 += S1[i].ToString();
Console.WriteLine("При обробці рядка '" + S1 + "' отримано рядок '" + S2 + "'");
Console.ReadKey();

```

2. Скласти програму для обробки текстового рядка.

Вимоги до програми:

- значення текстового рядка для обробки ввести з діалогового вікна;
- для обробки рядків використати оператори циклу;
- результати виконання алгоритму вивести в одному діалоговому вікні, відобразивши в ньому як вхідний так і вихідний текстові рядки.

Варіанти

1. Замінити в текстовому рядку кожен крапку трьома крапками;
2. Перетворити заданий текстовий рядок з n символів, видаливши кожен символ $*$ і повторивши кожен символ, відмінний від $*$;
3. Видалити з текстового рядка всі буквосполучення *про*;
4. В текстовому рядку замінити пробіли двома пробілами;
5. В текстовому рядку підрахувати кількість букв з вашого імені.
6. Замінити в текстовому рядку кожному двома знаками питання;
7. Перетворити заданий текстовий рядок, повторивши кожен символ, відмінний від '?';
8. Видалити з текстового рядка всі буквосполучення *не*;
9. В текстовому рядку замінити лапки двома комами;
10. В текстовому рядку видалити зайві пробіли між словами, залишивши по одному;
11. Переписати даний текстовий рядок так, щоб порядок символів у ньому став оберненим, самі символи подвоїлися і перед кожним з'явилася крапка;
12. У заданому текстовому рядку подвоїти всі букви *л, н, с, о*;
13. Видалити з текстового рядка всі коми і замінити знаком $+$ всі цифри 3 ;
14. Перетворити задану послідовність символів, замінивши всі буквосполучення *оце* буквосполученням *це*;
15. Видалити групи пробілів, якими починається і якими закінчується текстовий рядок.

Наприклад, програма для розв'язання завдання варіанту № 4 може бути такою:


```
String S1, S2;
S1 = Interaction.InputBox("Ведіть рядок для обробки", "Введення",
    "Значення по замощевчуванню");
S2 = "";
int i, len=S1.Length;
for(i=0; i<len; i++)
    if (S1[i]==' ')
        S2+=" ";
    else
        S2+=S1[i].ToString();
MessageBox.Show("При обробці рядка '" + S1 + "' отримано рядок '" + S2 +
    "'", "Результати", MessageBoxButtons.OK, MessageBoxIcon.Information);
```

А програма для розв'язання завдання варіанту № 14 може бути такою:

```
String S1, S2;
S1 = Interaction.InputBox("Ведіть рядок для обробки", "Введення",
    "Значення по замощевчуванню");
S2 = "";
int i, len=S1.Length;
if (len<3)
    S2=S1;
else
    {for(i=0; i<len-2; i++)
        if (S1[i]!='o' || S1[i+1]!='ц' || S1[i+2]!='e')
            S2+=S1[i].ToString();
        S2+=S1[len-2].ToString()+S1[len-1].ToString();
    }
MessageBox.Show("При обробці рядка '" + S1 + "' отримано рядок '" + S2 +
    "'", "Результати", MessageBoxButtons.OK, MessageBoxIcon.Information);
```

3. Скласти програму для обробки слів введеного текстового рядка.

Вимоги до програми:

- значення текстового рядка для обробки ввести з діалогового вікна;
- для обробки рядка використати оператори циклу;
- результати виконання алгоритму вивести в одному діалоговому вікні, відобразивши в ньому як вхідний рядок, так і отримані результати.

Варіанти

1. Перетворити даний рядок переводячи першу букву кожного слова до верхнього регістру;
2. Визначити, який процент слів у тексті починається з букви *к*;
3. Визначити, який процент слів у тексті містить подвоєну приголосну;
4. Визначити кількість слів у рядку, які починаються на українські голосні літери;
5. В заданому текстовому рядку всюди замінити слово *пара* словом *програмування* (врахуйте, що довжини слів не співпадають);

6. Студенти шифрують свої записки, записуючи всі слова навпаки. Скласти алгоритм і програму шифрування і розшифровування тексту;
7. Скласти абрєвіатуру з перших букв слів заданого текстового рядка.
8. Підрахувати загальну кількість слів;
9. Знайти кількість слів, що починаються на букву С;
10. Знайти кількість слів, в яких перший і останній символи співпадають;
11. Знайти яке-небудь слово, що починається з букви *a*, або вказати на його відсутність;
12. У словах з закінченням *ий* замінити його закінченням *енький*;
13. Знайти і вивести всі слова, що починаються складом *не*;
14. Визначити слово даного рядка з максимальною часткою голосних;
15. Визначити, скільки разів у тексті зустрічається введене слово;

Наприклад, програма для розв'язання завдання варіанту № 1 може бути такою:

```
class Program
{
    static bool RozdilZnak (Char C)
    {
        if (C==' ' || C==',' || C== '.' || C=='-' || C=='?')
            return true;
        else
            return false;
    }

    static void Main(string[] args)
    {
        String S, S1="";
        S = Interaction.InputBox("Ведіть рядок для обробки", "Введення",
            "Рядок для обробки");

        int i=0, j=0, len=S.Length;
        //починаємо обробку рядка;
        while (i<len)
        {
            while (i<len && RozdilZnak(S[i]))
                i++;
            if (i>j)
                S1+=S.Substring(j, i-j);
            if (i<len)
            {
                j=i+1;
                while (j<len && !RozdilZnak(S[j]))
                    j++;
                S1+= (S[i].ToString()).ToUpper();//дописати першу букву, підняту до верхнього регістру
                if ((j-i)>1) //в черговому слові більше однієї букви
                    S1+= S.Substring(i+1, j-i-1);
                i=j;
            }
        }
        MessageBox.Show("При обробці рядка\n" + S + "\nотримано рядок\n" + S1 + "", "Результати",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
```

Лабораторна робота № 17

- Тема.** Структури. Алгоритми обробки структурованих даних.
Мета. Формування вмінь і навиків створення та використання структур. Закріплення вмінь і навиків використання класів, об'єктів, підпрограм, функцій вводу-виводу та обробки рядків. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання

1. Як і навіщо використовуються структури?
2. Як описати структуру? Де це можна зробити?
3. Чим структури відрізняються від масивів?
4. Що таке ініціалізатор? Який синтаксис його використання для структури?
5. Як ініціалізувати структуру значеннями по замовчуванню? Як називаються методи для ініціалізації полів структури? Скільки їх може бути?

Завдання

1. Створіть новий консольний додаток та опишіть в ньому структуру для зберігання елемента масиву завдання № 2 згідно варіанту. Опишіть також у цій структурі процедуру *Info()* для виведення її даних. В процедурі *main()* оголошіть змінну розробленої структури. Внесіть в поля цієї змінної дані безпосередньо та викличте її метод *Info()* (2 бали до рейтингу).

Наприклад, структура для зберігання даних книги та їх виведення може бути описана так:

```
struct Book
{
    public string name;
    public string author;
    public int year;

    public void Info()
    {Console.WriteLine("Книга '{0}' (автор(и) {1}) була "+
        "видана в {2} році", name, author, year); }
}
```

Тоді використання цієї структури в процедурі *main()* буде таким:

```
static void Main(string[] args)
{
    Book book;
    book.name = "Изучаем C#";
    book.author = "Ендрю Стиллмен, Джениффер Грин";
    book.year = 2017;
    //Виводимо інформацію про книгу
    book.Info();
    Console.ReadLine();
}
```

2. **Доповніть розроблену у попередньому завданні структуру двома конструкторами: першим – для введення даних структури від користувача і другим – для ініціалізації переданими даними. В процедурі *main()* оголосіть масив структур (не менше п'яти записів) з ініціалізатором. В ініціалізаторі використайте як безпосередню ініціалізацію, так і виклик двох розроблених конструкторів. Розв'яжіть завдання згідно варіанту (3 бали до рейтингу).**

Варіанти

1. Масив студентів містить їх прізвища, імена та по батькові. Знайти дані студентів з вказаним прізвищем;
2. Масив студентів містить їх прізвища, імена та по батькові. Знайти дані студентів з вказаним ім'ям;
3. Масив студентів містить їх прізвища, імена, по батькові та аббревіатури груп. Знайти дані студентів вказаного класу;
4. Заданий масив, що містить перелік прізвищ студентів і назви вулиць, на яких вони проживають. Знайти дані студентів з вказаною вулицею;
5. Масив студентів містить їх прізвища, імена та по батькові. Знайти дані студентів з вказаним значенням по батькові;
6. Масив містить прізвища студентів і назви улюблених предметів кожного з них. Знайти дані студентів, для яких улюбленим є вказаний предмет;
7. Масив містить прізвища студентів та суми отриманих балів кожним з них. Знайти дані студентів з вказаною сумою балів;
8. Масив містить перелік футбольних команд та кількість очок, набраних у чемпіонаті кожною з них. Знайти команди, які набрали понад вказану кількість очок;
9. Масив співробітників містить їх прізвища, імена, по батькові і номери їх домашніх телефонів. Знайти телефони співробітника з вказаним прізвищем;
10. Заданий масив, що містить перелік прізвищ студентів і назви вулиць, на яких вони проживають. Знайти дані студентів з вказаним прізвищем;
11. Скласти програму-словник термінів з інформатики. Знайти пояснення вказаного терміна;
12. Масив містить відомості про автомобілі: їх марки, номери і прізвища власників. Знайти дані автомобілів вказаної марки;
13. Масив містить відомості про автомобілі: їх марки, номери і прізвища власників. Знайти всі автомобілі за вказаним прізвищем власника;
14. Масив містяться відомості про кожну книгу: прізвище автора, рік та назву видання. Знайти назви книг виданих після вказаного року;
15. Масив містить дані про кожну партію експортованих товарів: назву, країну, що імпортує товар та об'єм партії, що поставляється в штуках. Знайти товари, які імпортує вказана країна.

Наприклад, для пошуку найсвіжшої книги необхідно спочатку доповнити опис структури (в середині *struct*) двома конструкторами:

```

struct Book
{
    public string name;
    public string author;
    public int year;

    public Book(bool readData):this()
    {if (readData)
        {Console.Write("Введіть назву: ");
          this.name = Console.ReadLine();
          Console.Write("Введіть автора: ");
          this.author = Console.ReadLine();
          Console.Write("Введіть рік виходу: ");
          this.year = Convert.ToInt32(Console.ReadLine());
        }
    }

    public Book(string name, string author, int year)
    {this.name =name;
      this.author=author;
      this.year=year;
    }

    public void Info()
    {Console.WriteLine("Книга '{0}' (автор(и) {1}) була видана в {2} році",
                       name, author, year);
    }
}

```

Після чого для реалізації такого пошуку в процедурі *main()* слід, наприклад, вказати:

```

Book[] masBook = new Book[]
{ new Book {name = "Изучаем С#", author = "Єндрю Стілмен, Дженифер Грін"},
  new Book("Кобзар", "Тарас Григорович Шевченко", 1840),
  new Book(true), new Book(false), new Book(true)};
Console.WriteLine("Перелік книг:");
foreach (Book cBook in masBook)
    cBook.Info();
int index = 0, lastYear=masBook[0].year;
for (int i = 1; i < masBook.Count(); i++ )
    if (masBook[i].year > lastYear)
        {lastYear=masBook[i].year;
          index=i; }
Console.Write("Серед них найсвіжіша: ");
masBook[index].Info();
Console.ReadLine();

```

3. Використовуючи масив структур, скласти програму для розв'язання поставленого завдання (додатково 2 бали до рейтингу).

Варіанти

1. Скласти програму-словник термінів з інформатики;
2. Скласти програму, згідно якої для введеного з клавіатури англійського слова друкуються його значення українською та російською мовами. В словник помістити не менше 6 англійських слів.
3. В масиві містяться координати 5 точок. Для введеного з клавіатури числа R вказати, чи є серед заданих точок такі, що містяться в колі радіуса R з центром в початку координат;
4. В масиві містяться прізвища учнів та їх оцінки з алгебри та геометрії. Після введення з клавіатури прізвища учня на екран вивести його оцінки з відповідними назвами предметів. При відсутності в масиві такого прізвища видати відповідне повідомлення;
5. Масив містить відомості про автомобілі: їх марки, номери і прізвища власників. Вказати кількість автомобілів даної марки;
6. Масив містить відомості про автомобілі: їх марки, номери і прізвища власників. Вказати прізвища власників і номери автомобілів даної марки;
7. Масив містяться відомості про кожну книгу: прізвище автора, рік та назву видання. Знайти назви книг даного автора, виданих після 1960 року;
8. Масив містить дані про кожну партію експортованих товарів: назву, країну, що імпортує товар та об'єм партії, що поставляється в штуках. Знайти країни, в які експортується даний товар;
9. Масив містить дані про кожну партію експортованих товарів: назву, країну, що імпортує товар та об'єм партії, що поставляється в штуках. Вказати загальний об'єм експорту даного товару;
10. Масив містить назви іграшок, їх вартість у гривнях та вікові межі дітей, для яких вони призначені (значення мінімальної та максимальної вікової межі в роках). Вказати, чи дана іграшка коштує не більше P гривень і чи підходить вона для дитини віком R років. При відсутності такої іграшки в переліку видати відповідне повідомлення;
11. Масив містить назви іграшок, їх вартість у гривнях та вікові межі дітей, для яких вони призначені (значення мінімальної та максимальної вікової межі в роках). Вказати, ціна яких іграшок відрізняються від введеної з клавіатури ціни не більш ніж на 50 копійок. При відсутності таких іграшок в переліку видати відповідне повідомлення;
12. Масив містить назви іграшок, їх вартість у гривнях та вікові межі дітей, для яких вони призначені (значення мінімальної та максимальної вікової межі в роках). Вказати, чи вистачить даної суми грошей S на придбання хоча б якоїсь іграшки. При наявності таких іграшок вказати всі їх назви;
13. Масив містить відомості про кожного учня школи: прізвище, ім'я, та назву класу. Вивести дані учнів вказаного класу;
14. Масив містить відомості про кожного учня школи: прізвище, ім'я, та назву класу. Вказати назву класу, в якому навчається найбільше учнів;
15. Масив містить відомості про кожного учня школи: прізвище, ім'я, та назву класу. Визначити назву класу, в якому найчастіше зустрічається однакове прізвище. Вивести це прізвище та частоту його використання.

Лабораторна робота № 18

- Тема.** Створення методів користувача. Рекурсивні виклики методів.
Мета. Формування вмінь і навиків створення та використання процедур та функцій користувача. Закріплення вмінь і навиків використання функцій вводу-виводу та обробки рядків. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання

1. Коли та з якою метою доцільно використовувати підпрограми?
2. Чи прискорює час виконання програми використання підпрограм? Чи зменшує об'єм коду програми та час її розробки таке використання?
3. Які типи підпрограм використовуються в C#? Чим відрізняються синтаксиси їх опису? Як викликаються підпрограми кожного типу?
4. Де описуються підпрограми користувача? Як визначити їх область видимості?
5. Від чого залежить результат виконання підпрограми? Параметри яких типів даних можуть використовуватися підпрограмою?
6. Чим формальні параметри підпрограми відрізняються від фактичних?
7. Як повернути результат виконання підпрограми-функції в основну програму?
8. Які виклики функцій називаються рекурсивними? Як їх здійснити? Які вимоги при цьому мають виконуватися для уникнення зациклень?

Завдання

1. Скласти програму обчислення значення виразу згідно варіанту при значеннях параметрів по замовчуванню $x=1.5$; $y=0.1$; $z=0.5$.

Вимоги до програми:

- значення змінних ввести з діалогових вікон (використати розроблені раніше функції *InputInt* чи *InputDouble*);
- для розв'язання завдання та для перевірки введення числа створити та використати підпрограми;
- результат виконання програм вивести в діалоговому вікні.

Варіанти

$$1. S = \frac{\sqrt{4^2 + x^2}}{\sqrt{x^2 + y^2}} + \frac{\sqrt{65^2 + y^2}}{\sqrt{6,4^2 + z^2}};$$

$$2. S = \frac{87}{\lg|1 + x \tan y|} + \frac{\lg|1 + 2 \tan y|}{34} + \frac{7,54}{\lg|1 + \tan 1.3|};$$

$$3. S = \frac{78}{\lg|1.3 - x \tan z|} + \frac{\lg|x - 1.9 \tan y|}{4,75} + \frac{5,34}{\lg|zx - 3 \tan 1.2|};$$

4. $S = \frac{65}{\cos(xz - 1.7)} + \frac{\cos(3.6 - x^2)}{65.87} + \frac{6.74}{\cos(x - 0.3z)}$;
5. $S = \frac{\sqrt[3]{2 + \cos x^2}}{254} + \frac{9.56}{\sqrt[3]{3 + \cos y^2}} + \frac{\sqrt[3]{2 + \cos(xy)^2}}{7.4}$;
6. $S = \frac{\lg|1 + \cos^2 x^2|}{83} + \frac{56.9}{\lg|1 + \cos^2 0.8|} + \frac{\lg|1 + \cos^2 4.9|}{7.94}$;
7. $S = \frac{\lg|\sin x + y|}{87} + \frac{234}{\lg|\sin xy + 3|} + \frac{\lg|\sin 3.1 + x^2 y|}{z}$;
8. $S = \frac{\sqrt[3]{2 + \cos x^2}}{84} + \frac{5.87x}{\sqrt[5]{3 + \cos y^2}} + \frac{\sqrt[3]{2 + \cos(xy)^2}}{9}$;
9. $S = \frac{|x + y^2 \tan z|}{756} + \frac{8.45}{|x - 1 + \tan y|} + \frac{|x^2 + \tan 0.5|}{xy}$;
10. $S = \frac{\ln|\cos^2 x^2 + 1|}{67} - \frac{8.45}{\ln|\cos^2 x + 1|} + \frac{\ln|\cos^2 0.84 + 1|}{34x}$;
11. $S = \frac{\ln|x - y|}{76} - \frac{3zx}{\ln|1.3 - xy|} + \frac{\ln|y - 1.3z|}{98}$;
12. $S = \frac{\sin^2(x - y)}{78} + \frac{xy}{\sin^2(1.3 - xy)} + \frac{\sin^2(1.3x - 0.6)}{5z}$;
13. $S = \frac{45}{\sqrt[3]{\cos x^2 - x}} + \frac{\sqrt[5]{\cos y^2 - x}}{65x} - \frac{76zy}{\sqrt{\cos z^2}}$;
14. $S = \frac{x}{\cos^2(0.7 - x)} - \frac{\cos^3(0.7 - xy)}{59x} + \frac{5xyz}{\cos^2(0.7 - 1.4y)}$;
15. $S = \frac{x}{\sqrt{\cos x^2 + 2}} + \frac{87xz}{\sqrt[5]{\cos y^2 + 3}} + \frac{\sqrt[5]{\cos(yz)^2 + x}}{xy - z}$.

Наприклад, програма для розв'язання завдання варіанту № 1 може виглядати так:


```

static bool InputDouble(ref double x, string povidom)
{
    string s = x.ToString();
Povtor:
    s = Interaction.InputBox(povidom, "Введення", s);
    try
    {
        x = Convert.ToDouble(s);
    }
    catch
    {
        if (MessageBox.Show("Ви ввели не число.\n\nБажаєте повторити?", "Увага!",
            MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
            goto Povtor;
        else
            return false;
    }
    return true;
}

static double Koren(double a, double b)
{
    return Math.Sqrt(a*a+b*b);
}

static void Main(string[] args)
{
    Double x, y, z, suma;
    x = 1.5;
    y = 0.1;
    z = 0.5;
Povtor:
    if (!InputDouble(ref x, "Введіть x:"))
        return;
    if (!InputDouble(ref y, "Введіть y:"))
        return;
    if (!InputDouble(ref z, "Введіть z:"))
        return;
    suma = Koren(4, x) / Koren(x, y) + Koren(65, y) / Koren(6.4, z);
    if (MessageBox.Show("При x=" + x.ToString() + ", y=" + y.ToString()+
        " та z=" + z.ToString() + " значення виразу рівне " + suma.ToString() +
        "\n\nБажаєте повторити?", "Результати обчислень", MessageBoxButtons.YesNo,
        MessageBoxIcon.Information) == DialogResult.Yes)
        goto Povtor;
}

```

2. Скласти програму для розв'язання поставленого завдання.

Вимоги до програми:

- значення початкових параметрів обчислень ввести з клавіатури з відповідною перевіркою коректності типів та даних;
- для розв'язання завдання створити та використати підпрограми;
- результати виконання програми вивести в одному діалоговому вікні.

Варіанти

- Обчислити значення функцій $f_1(x) = \sin(x^2)^3$, $f_2(x) = 1 - x^2$, $f_3(x) = 3^{x^2}$ для значень аргументів $x_i = a + ih$ ($i = 0, 1, 2, \dots, n$), де $h = \frac{b-a}{n}$, якщо $a=0$, $b=3$, $n=8$. Отримані значення аргументів та функцій вивести у вигляді таблиці з чотирма стовпцями;
- Два спортсмени одночасно починають рух з однієї точки. Перший спортсмен починає рух із швидкістю 10 км./год. і рівномірно, за кожну наступну годину збільшує швидкість на 1 км. Другий починає рух зі швидкістю 9 км/год і збільшує швидкість теж рівномірно на 1 км/год. Визначити, який спортсмен пройде більший шлях через 1 год.? Через 4 год.? (Відстань, пройдена першим спортсменом, описується залежністю $S=10t+0.5t^2$, а другим – залежністю $S=9t+0.8t^2$, де t - час в годинах);
- Дано три сторони трикутника. Визначити його кут. (Згідно теореми косинусів кут між сторонами a та b дорівнює $\arccos \frac{a^2 + b^2 - c^2}{2ab}$. Для обчислення арккосинуса використати співвідношення $\arccos(x) = \arctan \sqrt{\frac{1}{x^2} - 1}$);
- Обчислити значення функцій $f_1(x) = \sin \cos(2^x - 2^{-x})$ та $f_2(x) = \tan \cos^2(2^x - 2^{-x})$ на відрізку $[0;1]$ з кроком 0.1;
- Обчислити за введеними довжинами двох сторін трикутника і кутом між ними довжину третьої сторони. (При обчисленні довжини сторони використати теорему косинусів: $c^2 = a^2 + b^2 - 2ab \cos x$);
- Два трикутники задані координатами своїх вершин А, В та С. Обчислити, площі трикутників, не використовуючи формули Герона, і визначити, який з трикутників має більшу площу. При розв'язанні завдання використати такі дані:
вершини першого трикутника – А(1;1), В(5;2), С(3;3);
вершини другого трикутника – А(2;5), В(4;3), С(6;4).
(Площа трикутника, заданого координатами вершин А(x1;y1), В(x2;y2) та С(x3;y3) згідно формул векторної алгебри обчислюється за формулою $S = 0.5[(x2-x1)(y3-y1)(x3-x1)(y2-y1)]$);
- Футболіст ударом ноги посилає м'яч вертикально вгору з висоти 1 м з початковою швидкістю 20 м/с. На якій висоті м'яч буде через 1 сек.; 2 сек.; 3 сек.? (Рух м'яча описується залежністю $y(t) = y_0 + v_0 t + \frac{gt^2}{2}$, де $y(t)$ – висота м'яча в момент часу t , y_0 – початкова висота, v_0 – початкова швидкість, $g=9.8$ м/сек.²);

8. Два трикутники задані своїми сторонами a , b та c . Обчислити площі трикутників за формулою Герона і визначити, який з трикутників має більшу площу. При розв'язанні завдання використати такі дані:

для першого трикутника – $a=3, b=4, c=5$;

для другого трикутника – $a=2, b=\sqrt{37}, c=\sqrt{37}$.

9. Траєкторія снаряда, що вилітає з гармати під кутом α з початковою швидкістю v_0 , описується системою рівнянь

$$\begin{cases} x(t) = v_0 t \cos \alpha \\ y(t) = v_0 t \sin \alpha - \frac{gt^2}{2}, \end{cases}$$

де $(x(t); y(t))$ – координати снаряду в момент часу t , v_0 – початкова швидкість, $g=9.8$ м/сек². Визначити положення снаряду через 1 сек., 2 сек., 4 сек. при $\alpha = \frac{\pi}{4}$ та $v_0 = 350$ м/с;

10. Населені пункти задані своїми координатами $M_1(-1;1)$, $M_2(1.5;2)$, $M_3(6;4)$, $M_4(3;4)$. Визначити, який населений пункт найближче розміщений до залізничної колії, що задається рівнянням $3x-4y+5=0$. (Для обчислення відстані від заданого пункту до залізничі скористайтеся формулою

$$d = \frac{|ax + by + c|}{\sqrt{a^2 + b^2}},$$

де $a=3, b=-4, c=5$ згідно наведеного рівняння);

11. Точки трьохвимірного простору задані своїми координатами: $A(1.5;2.1;0.3)$, $B(3;0.2;0.7)$, $C(0.7;-2.5;2)$ та $D(0.4;8.5;-1.2)$. Знайти найвіддаленішу від початку координат точку. Вивести її координати та відповідну відстань;
12. Сформувати таблицю значень многочленів $2x^2 - 3x^2 + x - 7$ та $5x^2 - 18x + 7$ для $x=1, 2, \dots, 7$;
13. Обчислити максимум із значень двох функцій $f_1(x) = \sin \cos(2^x - 2^{-x})$ та $f_2(x) = \tan \cos^2(2^x - 2^{-x})$ у вказаній точці;
14. Відомі координати точок A, B, C та D . Знайти різницю периметрів трикутників AOC і BOD якщо точка O знаходиться в центрі координат;
15. Обчислити значення функції $f(0), f(3), f(4.5)$, якщо

$$f(x) = \tan^2 e^x + |\sin x| \ln(x+1).$$

Наприклад, програма для розв'язання завдання варіанту № 8 може мати такий вигляд:

```

static double SGeron(double a, double b, double c)
{
    double p = (a + b + c) / 2;
    return Math.Sqrt(p * (p - a) * (p - b) * (p - c));
}

static void Main(string[] args)
{
    Double a1=3, b1=4, c1=5, a2=2, b2=37, c2=37;
    Повтор:
    if (!InputDouble(ref a1, "Введіть довжину сторони а першого трикутника"))
        return;
    if (!InputDouble(ref b1, "Введіть довжину сторони б першого трикутника"))
        return;
    if (!InputDouble(ref c1, "Введіть довжину сторони с першого трикутника"))
        return;
    if (!InputDouble(ref a2, "Введіть довжину сторони а другого трикутника"))
        return;
    if (!InputDouble(ref b2, "Введіть довжину сторони б другого трикутника"))
        return;
    if (!InputDouble(ref c2, "Введіть довжину сторони с другого трикутника"))
        return;
    double S1 = SGeron(a1, b1, c1);
    double S2 = SGeron(a2, b2, c2);
    string res = "Площі трикутників рівні";
    if (S1>S2)
        res = "Площа першого трикутника перевищує площу другого трикутника";
    if (S2 > S1)
        res = "Площа другого трикутника перевищує площу першого трикутника";
    if (MessageBox.Show("Площа першого трикутника: " + S1.ToString() +
        ".\nПлоща другого трикутника: " + S2.ToString()+".\n"+res+
        "\n\nБажаєте повторити?", "Результати обчислень", MessageBoxButtons.YesNo,
        MessageBoxIcon.Information) == DialogResult.Yes)
        goto Повтор;
}

```

3. Скласти програму для розв'язання поставленого завдання з рекурсіями згідно вимог до програми попередньої задачі.

Варіанти

- Дано парне число $n > 2$. Перевірити для цього числа гіпотезу Гольбахта. Ця гіпотеза твердить, що кожне парне n більше двох можна подати у вигляді суми двох простих чисел. При розв'язанні використати підпрограму, яка дозволяє розпізнавати прості числа;
- Серед чисел введеної послідовності знайти та вивести досконалі, використавши при цьому підпрограму для розпізнавання досконалих чисел. (Натуральне число називається досконалим, якщо воно рівне сумі всіх своїх дільників, за винятком самого себе. Наприклад, число 6 – досконале, оскільки $6=1+2+3$, число 8 – не досконале, оскільки $8 \neq 1+2+4$);
- Написати програму побудови латинського квадрату. Латинський квадрат – це матриця розмірності $N \times N$, елементи якої вибрані від 1 до N так, що кожне число зустрічається один і лише один раз у кожному рядку і в кожному стовпці. (При формуванні перший рядок матриці заповнити цифрами від 1 до N . Кожен наступний рядок отримувати з попереднього, циклічно

зсуваючи його елементи на один елемент вправо (вліво). Черговий рядок матриці формувати в одновимірному масиві у підпрограмі, використовуючи масив, сформований при попередньому звертанні, як вхідний);

4. Написати підпрограму визначення найменшого за абсолютною величиною елемента одновимірного масиву довільного розміру і використати її у програмі обчислення мінімального елемента серед мінімальних елементів різних одновимірних масивів;
5. Дано натуральне число n . Вияснити, чи є серед чисел $n, n+1, \dots, 2n$ близнята, тобто прості числа, різниця між якими рівна двом. Використати підпрограму, яка визначає, чи є задане число простим;
6. Назвемо натуральне число паліндромом, якщо його запис читається однаково з початку до кінця і з кінця до початку (як, наприклад, 4884, 393, 1). Знайти всі менші 100 натуральні числа які при піднесенні до квадрату дають паліндроми. Використати підпрограму, яка визначає, чи є задана послідовність символів паліндромом;
7. Назвемо натуральне число паліндромом, якщо його запис читається однаково з початку до кінця і з кінця до початку (як, наприклад, 4884, 393, 1). Знайти всі менші 100 числа-паліндроми, які при піднесенні до квадрату теж дають паліндроми. Використати підпрограму, яка визначає, чи є задана послідовність символів паліндромом;
8. Дано натуральне число n . Знайти всі менші за n числа Мерсена. (Просте число називається числом Мерсена, якщо воно може бути подано у вигляді 2^p-1 , де p – теж просте число);
9. Дано натуральне число n в десятковій системі числення. Перевести його в двійкову систему числення;
10. Дано натуральне число n в двійковій системі числення. Перевести його у десяткову систему числення;
11. Дано натуральне число m . Знайти таке натуральне число n , двійковий запис якого отримується з двійкового запису числа m зміною порядку цифр на протилежний (m задано в десятковій системі і n треба також отримати в десятковій системі, наприклад, для $m=6$ результат $n=3$). Оформити у вигляді підпрограм переведення числа з десяткової системи числення у двійкову і навпаки;
12. Дано список, що містить прізвища осіб та їх вік в роках. Отримати список осіб впорядкований за спаданням віку;
13. Дано натуральне число N . Отримати f_1, f_2, \dots, f_n , де
$$f_i = \frac{1}{i^2 + 1} + \frac{1}{i^2 + 2} + \dots + \frac{1}{i^2 + i + 1};$$
14. Опуклий n -кутник заданий координатами своїх вершин $(x_1; y_1), (x_2; y_2), \dots, (x_n; y_n)$. Знайти його площу, сумуючи послідовно для всіх точок, починаючи з третьої, площі трикутників з вершинами у першій, другій та активній точці;
15. У послідовності з n натуральних чисел знайти підпослідовності членів, що йдуть підряд та складаються із степенів п'ятірки $(1, 5, 25, \dots)$. Вивести найдовшу з отриманих підпослідовностей.

Лабораторна робота № 19

- Тема.** Розробка та використання власних бібліотек методів класів.
Мета. Формування вмінь і навиків створення та використання власних бібліотек. Закріплення вмінь і навиків використання функцій стандартних бібліотек, власних підпрограм, функцій вводу-виводу та обробки рядків. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання

1. Що таке бібліотека класів? Для чого застосовуються бібліотеки?
2. Для чого в бібліотеках використовуються простори імен?
3. Які класи описуються в бібліотеках, щоб використовувати їх функції без створення об'єктів? Чому?
4. У скількох програмах можна підключити створені бібліотеки і як це зробити?
5. Як підключена бібліотека впливає на розмір створеного тексту програми і на розмір самої програми?

Завдання

1. **Створіть модуль з власними підпрограмами. Для цього:**
 - 1.1. Створіть у вашій папці з текстами програм вкладену папку *Units*;
 - 1.2. Створіть у цій папці новий проєкт бібліотеки класів *UnitNickName*, де *NickName* – **ваше прізвище англійською мовою**, послідовно обравши в середовищі розробки *Файл – Создать – Проект – Visual C# – Библиотека классов* (для платформи *.Net Framework*) та вказавши у цій папці зазначену назву проєкту;
 - 1.3. В панелі оглядача рішень виділіть файл проєкту (з розширенням *.cs*) та перейдіть до його коду (яка клавіша дозволяє це швидко зробити?). Опишіть у створеному модулі **публічний** статичний клас *analizText*, а в ньому створіть чи скопіюйте **публічну** статичну функцію *rozdilZnak* з ЛР обробки рядків. Після класу *analizText* опишіть **публічний** статичний клас *corectInput*, а в ньому створіть чи скопіюйте **публічні** статичні функції *inputDouble* та *inputInt* з ЛР розв'язування квадратного рівняння чи циклів з перед- та післяумовою. При цьому код бібліотеки має бути подібним до такого:

```

namespace UnitNickName
{
    public static class analizText
    {public static bool rozdilZnak(char c)
      {if (c == ' ' || c == '.' || c == ',' || c == ':' || c == ';')
        return true;
        else
        return false;
      }
      //...
    }

    public static class corectInput
    {public static bool inputDouble(ref double x, string povidom)
      {string s;
        s = x.ToString();
        Povtor:
        s = Interaction.InputBox(povidom, "Введення", s);
        try
        {x = Convert.ToDouble(s);
        }
        catch (System.FormatException)
        {if (MessageBox.Show("Ви ввели не дійсне число.\n\nБажаєте повторити?", "Увага",
            MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
            goto Povtor;
            else
            return false;
        }
        return true;
      }
    }

    public static bool inputInt(ref int x, string povidom)
    {//...
      return true;
    }
    //...
} //завершення опису простору імен

```

- 1.4. Відкомпілюйте проект та усуньте всі помилки, не звертаючи уваги на попередження про неможливість його завантаження.
2. Підключіть розроблену бібліотеку до форм ЛР обробки рядків та використайте її підпрограму замість описаної у формі. Для цього:
 - 2.1. Відкрийте проект ЛР обробки рядків;
 - 2.2. Додайте посилання на файл розробленої бібліотеки у цьому проекті. Для цього перейдіть в панель оглядача рішень і в контекстному меню групи проекту *References* оберіть пункт *Добавить ссылку*, у вікні додавання посилання натисніть кнопку *Обзор*, виділіть файл розробленої бібліотеки *UnitNickName.dll* за шляхом *...Units\<Рішення бібліотеки>\<Прокт бібліотеки>Bin\Debug\<Рішення>.dll* та послідовно натисніть кнопки *Добавить* і *ОК*. Де з'явилася назва підключеної бібліотеки?

- 2.3. Відкрийте простір імен з доданої бібліотеки у формі проекту ЛР обробки рядків. Для цього зверху у вікні коду форми допишіть, наприклад *using UnitNickName*;
 - 2.4. Видаліть **реалізацію функції *rozdilZnak*** (а не звертання до неї) у кодї форми ЛР обробки рядків;
 - 2.5. Замість звертання до локальної функції *rozdilZnak* використовуйте аналогічну функцію власної бібліотеки. Для цього у всіх місцях використання цієї функції зверніться до статичної функції статичного класу відкритого простору імен бібліотеки, використовуючи замість *rozdilZnak* звертання *analizText.rozdilZnak*.
- 3. Самостійно підключіть розроблену бібліотеку до форми ЛР розв'язування квадратного рівняння та використовуйте її підпрограми замість описаних у формі *inputInt* та *inputDouble*.**

З питань тиражування та використання цього видання звертайтеся на e-mail books_s@ukr.net

Лабораторна робота № 20

Тема. Обробка файлів як цілісних об'єктів з використанням масивів рядків.

Мета. Формування вмінь і навиків створення, запису, читання, копіювання, перейменування, видалення файлів. Закріплення вмінь і навиків використання функцій вводу-виводу та обробки рядків. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання

1. Які основні операції виконуються над файлами засобами C#?
2. Який простір імен і навіщо потрібно відкрити в програмі для роботи з файлами?
3. Які варіанти створення файлів як цілісних об'єктів передбачені в C#?
4. Які варіанти зчитування вмісту всього файла передбачені в C#?
5. Як скопіювати, перейменувати чи видалити файл засобами C#?
6. Як перевірити існування файла перед зчитуванням?
7. Як створити, скопіювати, перейменувати чи видалити каталог засобами C#?

Завдання

1. **Скласти програму, в якій: створити масив рядків для зберігання п'яти прізвищ ваших улюблених письменників; ввести дані цього масиву з діалогових вікон; записати дані цього масиву в файл у власну папку на жорсткому диску однією командою (ім'я цього файла – ваше прізвище, а розширення – *txt*); скопіювати програмно створений файл на власний флеш-носії. Використовуючи довільний текстовий редактор, дописати у створений файл на жорсткому диску ще одне прізвище письменника.**

Вимоги до програми:

- значення даних програми ввести з клавіатури;
- при введенні порожнього рядка передбачити його повторне введення.

Наприклад, програма для введення даних п'яти друзів та запису їх у файли може мати вигляд:

```
using System;
using Microsoft.VisualBasic;
using System.Windows.Forms;
using System.IO;

namespace LR17FileTxt
{
    class Program
    {
        static void Main(string[] args)
        {
            string[] friends = new string[5];
            for (int i = 0; i < friends.Length; i++)
            {
                povtorI:
                friends[i] = Interaction.InputBox("Введіть назву " + (i + 1).ToString() + "-го друга", "Введення");
                if (friends[i] == "")
                {
                    if (MessageBox.Show("Назва не введена!\n\nБажаєте повторити?", "Увага!",
                        MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation) == DialogResult.Yes)
                        goto povtorI;
                }
                else
                    return;
            }
        }
    }
}
```

```

    }
    File.WriteAllLines("D:\\kibernetik.txt", friends);
    if (File.Exists("L:\\kibernetik.txt"))
        File.Delete("L:\\kibernetik.txt");
    File.Copy("D:\\kibernetik.txt", "L:\\kibernetik.txt");
    MessageBox.Show("Дані у файли збережено успішно", "Результати", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
}
}

```

2. Створити програму для зчитування даних створеного файлу однією командою та виводу їх у діалогове вікно.

Вимоги до програми:

- перед відкриттям файла передбачити перевірку існування файла. При відсутності файла видати відповідне повідомлення та завершити виконання програми;
- результати виконання вивести в одному діалоговому вікні

Наприклад, програма для зчитування даних друзів з файлу та виводу їх на екран мати вигляд:

```

static void Main(string[] args)
{
    string[] friends;
    if (!File.Exists("D:\\kibernetik.txt"))
        {MessageBox.Show("Файл з даними друзів відсутній.\n\nЗчитування виконати неможливо", "Увага!",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        return;
    }
    friends = File.ReadAllLines("D:\\kibernetik.txt");
    string res = "";
    for (int i = 0; i < friends.Length; i++)
        res += (i + 1).ToString() + ". " + friends[i] + "\n";
    MessageBox.Show("Перелік друзів:\n" + res, "Результати", MessageBoxButtons.OK, MessageBoxIcon.Information);
}

```

З питань тиражування та використання

Лабораторна робота № 21

Тема. Обробка текстових файлів та потоків.

Мета. Формування вмінь і навиків створення та використання текстових файлів і потоків. Закріплення вмінь і навиків використання функцій вводу-виводу та обробки рядків. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання

1. Коли та з якою метою доцільно використовувати файли в програмуванні?
2. Які типи файлів і потоків використовуються в С#? У чому переваги та недоліки кожного типу стосовно форматів записів та швидкості обробки?
3. Що відбувається на рівні операційної системи при відкритті файла?
4. Як створити текстовий файл?
5. Як здійснюється зчитування даних з текстового файла?
6. Як здійснюється перевірка кінця текстового файла і коли вона застосовується?
7. Як закрити файл? Що відбувається на рівні операційної системи при закритті файла?

Завдання

1. **Скласти дві програми: першу – для створення файла послідовного доступу, другу - для опрацювання даних, записаних у цей файл.**

Вимоги до програми:

- значення даних першої програми ввести з клавіатури з відповідною перевіркою коректності типів та даних;
- перед відкриттям файла в другій програмі для обробки передбачити перевірку існування файла. При відсутності файла видати відповідне повідомлення та завершити виконання програми;
- результати виконання другої програми вивести у вікні консольного додатку чи в одному діалоговому вікні.

Варіанти

1. Протокол лижних гонок записати у файл *Ski.txt*. Для кожного учасника вводиться прізвище, час старту (години, хвилини, секунди) та час фінішу. Використовуючи сформований файл, вивести прізвища учасників, які виконали норматив;
2. Сформувати Файл *Book.txt* з прізвищ улюблених письменників учнів класу (учні по черзі вводять по три прізвища письменників). Використовуючи сформований файл, надрукувати прізвища п'яти найпопулярніших письменників;
3. Сформувати файл *Exam.txt* за результатами здачі сесії групою студентів. При цьому для кожного студента вводиться прізвище та три відповідні оцінки. Використовуючи сформований файл, вивести результати сесії у вигляді таблиці. У заголовку таблиці вивести назви предметів;

4. Задана розріджена матриця (розмірності не більш 10x10), яка складається з нулів та одиниць. Сформувати файл *Mlist.txt*, в якому запам'ятати матрицю в такому вигляді: кількість рядків, кількість стовпців, а далі – номери рядків та стовпців, на перетині яких знаходяться ненульові елементи. Використовуючи сформований файл, вивести матрицю на екран у звичному вигляді;
5. Задана розріджена матриця (розмірності не більш 10x10). Сформувати файл *Mlist.txt*, в якому запам'ятати матрицю в такому вигляді: кількість рядків, кількість стовпців, а далі – номери рядків, стовпців та значень ненульових елементів. Використовуючи сформований файл, вивести матрицю на екран у звичному вигляді;
6. Сформувати файл *Friend.txt* з прізвищ та дат (дня, місяця і року) народження Ваших друзів. Використовуючи сформований файл, вивести прізвища тих, хто народився влітку;
7. Сформувати файл *Name.txt*, в якому зберегти список імен. Використовуючи сформований файл, вивести імена, що починаються з букви *A*;
8. Сформувати файл *Stud.txt* з наступною структурою: прізвище студента, стать (одна буква – *Ч* або *Ж*), рік народження. Вивести список студентів чоловічої статі, вказавши їх вік. Після списку вивести середній вік відібраних студентів;
9. Сформувати файл *Car.txt*, що має містити інформацію про автолюбителів у форматі: прізвище водія, марка автомобіля, колір. Використовуючи сформований файл, вивести на екран дисплея відомості про автолюбителів, що мають автомобіль *Жигулі* зеленого кольору;
10. Сформувати файл *Stud.txt* з даними студентів одного курсу (прізвище, стать, рік та місяць народження). Використовуючи сформований файл, вивести прізвища студентів чоловічої статі, що народились влітку;
11. Сформувати файл *Rezklad.txt*, що містить інформацію про поїзди, які відправляються з залізничного вокзалу м. Києва (номер поїзда, станція призначення, час відправлення, час в дорозі). Використовуючи сформований файл, вивести інформацію про поїзди, які відправляються у ковельському напрямі від 20 до 24 годин;
12. Сформувати файл *Play.txt*, що має містити інформацію про побутові магнітофони у форматі: марка магнітофона та його вартість. Використовуючи сформований файл, вивести інформацію про магнітофони вартістю менше 500 гривень;
13. Сформувати файл *Book.txt*, який має містити дані про книги з програмування Вашою особистою бібліотекою (прізвище автора і його ініціали, назва книги, рік видання). Використовуючи сформований файл, вивести на екран дані останньої виданої книги;
14. Сформувати файл *Magazin.txt*, що має містити відомості про магазини Вашого району у форматі: тип товарів магазину (*Універмаг, Продукти, Тканини, ...*), його номер та адресу. Використовуючи сформований файл, вивести інформацію про всі магазини що займаються книгами;

15. Сформувати файл *Kub.txt*, що має містити відомості про кубики у форматі: розмір кожного кубика (довжина ребра в сантиметрах), його колір (червоний, жовтий, зелений чи синій) та матеріал (дерево, метал, картон). Знайти кількість дерев'яних кубиків з ребром 3 см і кількість металевих кубиків з ребром, більшим за 5 см.

Наприклад, програма для формування файлу завдання варіанту № 13 може мати вигляд:

```
struct Book
{
    public string name;
    public string author;
    public int year;

    public Book(bool readData):this()
    {
        if (readData)
        {
            Console.Write("Введіть назву: ");
            this.name = Console.ReadLine();
            Console.Write("Введіть автора: ");
            this.author = Console.ReadLine();
            Console.Write("Введіть рік виходу: ");
            this.year = Convert.ToInt32(Console.ReadLine());
        }
    }

    public Book(string name, string author, int year)
    {this.name =name;
    this.author=author;
    this.year=year;
    }

    public void Info()
    {
        Console.WriteLine("Книга '{0}' (автор(и) {1}) була видана в {2} році", name, author, year);
    }
}
...
Book[] masBook = new Book[] { new Book { name = "Изучаем С#", author = "Єндрю Стілмен, Джениффер Грин" },
    new Book("Кобзар", "Тарас Григорович Шевченко", 1840),
    new Book(true), new Book(false), new Book()};
StreamWriter f = new StreamWriter("I:\\LR18.txt");
int i;
for (i = 0; i < masBook.Count(); i++)
{
    f.WriteLine(masBook[i].author);
    f.WriteLine(masBook[i].name);
    f.WriteLine(masBook[i].year.ToString());
}
f.Close();
```

А програма для розв'язання цього завдання може бути такою:

```
StreamReader f1 = new StreamReader("I:\\LR18.txt");
int countLine=0;
while (!f1.EndOfStream)
{f1.ReadLine();
 countLine++;
}
if (countLine==0)
{
 Console.WriteLine("Дані книг відсутні");
 Console.ReadLine();
}

f1.BaseStream.Seek(0, SeekOrigin.Begin); //позиціонуємо на початок в базовому потоці
f1.DiscardBufferedData();//очищаємо проміжний буфер для коректного зчитування з початку
masBook = new Book[countLine / 3];
i = 0;
while (!f1.EndOfStream)
{masBook[i].author=f1.ReadLine();
 masBook[i].name = f1.ReadLine();
 masBook[i].year = Int32.Parse(f1.ReadLine());
 i++;
}
f1.Close();
Console.WriteLine("Перелік книг:");
foreach (Book cBook in masBook)
 cBook.Info();
int index = 0, lastYear = masBook[0].year;
for (i = 1; i < masBook.Count(); i++)
 if (masBook[i].year > lastYear)
 {
 lastYear = masBook[i].year;
 index = i;
 }
Console.Write("Серед них найсвіжіша: ");
masBook[index].Info();
Console.ReadLine();
```

2. Скласти програму для дозапису даних у файл послідовного доступу, створений в результаті виконання першої програми попереднього завдання. Після дозапису даних у файл знову виконати програму обробки його даних (тобто другу програму попереднього завдання).

Лабораторна робота № 22

Тема. Опрацювання двійкових файлів та потоків послідовного доступу.

Мета. Формування вмінь і навиків створення та використання двійкових файлів і потоків. Закріплення вмінь і навиків використання функцій вводу-виводу та обробки рядків. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання

1. Коли та з якою метою доцільно використовувати двійкові файли?
2. У чому переваги та недоліки послідовних відносно типізованих двійкових файлів?
3. Для чого в класах обробки потоків використовується властивість *BaseStream*?
4. Як створити двійковий файл?
5. Як здійснюється зчитування/запис даних з двійкового файла?
6. Як в двійкових файлах зберігаються рядки? Від чого залежить їх розмір?
7. Як здійснюється перевірка кінця двійкового файла і коли вона застосовується?
8. Як закрити файл? Що відбувається на рівні операційної системи при закритті файла?

Завдання

1. **Скласти дві програми: першу – для створення двійкового файла послідовного доступу, другу - для опрацювання даних, записаних у цей файл.**

Вимоги до програми:

- значення даних першої програми ввести з клавіатури з відповідною перевіркою коректності типів та даних;
- перед відкриттям файла в другій програмі для обробки передбачити перевірку існування файла. При відсутності файла видати відповідне повідомлення та завершити виконання програми;
- результати виконання другої програми вивести у вікні консольного додатку чи в одному діалоговому вікні.

Варіанти

1. Протокол лижних гонок записати у файл *Ski.dat*. Для кожного учасника вводиться прізвище, час старту (години, хвилини, секунди) та час фінішу. Використовуючи сформований файл, вивести прізвища учасників, які виконали норматив;
2. Сформуванати Файл *Book.dat* з прізвищ улюблених письменників учнів класу (учні по черзі вводять по три прізвища письменників). Використовуючи сформований файл, надрукувати прізвища п'яти найпопулярніших письменників;

3. Сформувати файл *Exam.dat* за результатами здачі сесії групою студентів. При цьому для кожного студента вводиться прізвище та три відповідні оцінки. Використовуючи сформований файл, вивести результати сесії у вигляді таблиці. У заголовку таблиці вивести назви предметів;
4. Задана розріджена матриця (розмірності не більш 10x10), яка складається з нулів та одиниць. Сформувати файл *Mlist.dat*, в якому запам'ятати матрицю в такому вигляді: кількість рядків, кількість стовпців, а далі – номери рядків та стовпців, на перетині яких знаходяться ненульові елементи. Використовуючи сформований файл, вивести матрицю на екран у звичному вигляді;
5. Задана розріджена матриця (розмірності не більш 10x10). Сформувати файл *Mlist.dat*, в якому запам'ятати матрицю в такому вигляді: кількість рядків, кількість стовпців, а далі – номери рядків, стовпців та значень ненульових елементів. Використовуючи сформований файл, вивести матрицю на екран у звичному вигляді;
6. Сформувати файл *Frend.dat* з прізвищ та дат (дня, місяця і року) народження Ваших друзів. Використовуючи сформований файл, вивести прізвища тих, хто народився влітку;
7. Сформувати файл *Name.dat*, в якому зберегти список імен. Використовуючи сформований файл, вивести імена, що починаються з букви *A*;
8. Сформувати файл *Stud.dat* з наступною структурою: прізвище студента, стать (одна буква – *Ч* або *Ж*), рік народження. Вивести список студентів чоловічої статі, вказавши їх вік. Після списку вивести середній вік відібраних студентів;
9. Сформувати файл *Car.dat*, що має містити інформацію про автолюбителів у форматі: прізвище водія, марка автомобіля, колір. Використовуючи сформований файл, вивести на екран дисплея відомості про автолюбителів, що мають автомобіль *Жигулі* зеленого кольору;
10. Сформувати файл *Stud.dat* з даними студентів одного курсу (прізвище, стать, рік та місяць народження). Використовуючи сформований файл, вивести прізвища студентів чоловічої статі, що народились влітку;
11. Сформувати файл *Rozklad.dat*, що містить інформацію про поїзди, які відправляються з залізничного вокзалу м. Києва (номер поїзда, станція призначення, час відправлення, час в дорозі). Використовуючи сформований файл, вивести інформацію про поїзди, які відправляються у кovelьському напрямі від 20 до 24 годин;
12. Сформувати файл *Play.dat*, що має містити інформацію про побутові магнітофони у форматі: марка магніфона та його вартість. Використовуючи сформований файл, вивести інформацію про магнітофони вартістю менше 500 гривень;
13. Сформувати файл *Book.txt*, який має містити дані про книги з програмування Вашою особистою бібліотекою (прізвище автора і його ініціали, назва книги, рік видання). Використовуючи сформований файл, вивести на екран дані останньої виданої книги;

14. Сформувати файл *Magazin.dat*, що має містити відомості про магазини Вашого району у форматі: тип товарів магазину (*Універмаг, Продукти, Тканини, ...*), його номер та адресу. Використовуючи сформовану файл, вивести інформацію про всі магазини що займаються книгами;
15. Сформувати файл *Kub.dat*, що має містити відомості про кубики у форматі: розмір кожного кубика (довжина ребра в сантиметрах), його колір (червоний, жовтий, зелений чи синій) та матеріал (дерево, метал, картон). Знайти кількість дерев'яних кубиків з ребром 3 см і кількість металевих кубиків з ребром, більшим за 5 см.

Наприклад, програма для формування файлу завдання варіанту № 13 може мати такий вигляд:

```
Book[] masBook = new Book[] { new Book { name = "Изучаем C#", author = "Ендрю Стіллен, Джениффер Грін" },
                               new Book("Кобзар", "Тарас Григорович Шевченко", 1840),
                               new Book(true), new Book(false), new Book(), book,book1,book2};
FileStream f = new FileStream("I:\\LR19.dat", FileMode.Create);
BinaryWriter w = new BinaryWriter(f,Encoding.UTF8);
int i;
for (i = 0; i < masBook.Count(); i++)
{
    w.Write(masBook[i].author+"");
    Console.WriteLine(w.BaseStream.Position);
    w.Write(masBook[i].name+"");
    Console.WriteLine(w.BaseStream.Position);
    w.Write(masBook[i].year);
    Console.WriteLine(w.BaseStream.Position);
}
w.Close();
```

А програма для розв'язання цього завдання може бути такою:

```
f = new FileStream("I:\\LR19.dat", FileMode.OpenOrCreate);
BinaryReader r = new BinaryReader(f, Encoding.UTF8);
int countZap=0;
while (r.BaseStream.Position<r.BaseStream.Length)
{r.ReadString();
 r.ReadString();
 r.ReadInt32();
 countZap++;
}
if (countZap == 0)
{Console.WriteLine("Дані книг відсутні");
 Console.ReadLine();
}
r.BaseStream.Seek(0,SeekOrigin.Begin); //позиціонуємо на початок в базовому потоці
masBook = new Book[countZap];
i = 0;
while (i < countZap)
{masBook[i].author=r.ReadString();
 masBook[i].name = r.ReadString();
 masBook[i].year = r.ReadInt32();
 i++;
```

```

}
r.Close();
Console.WriteLine("Перелік книг:");
foreach (Book cBook in masBook)
    cBook.Info();
int index = 0, lastYear = masBook[0].year;
for (i = 1; i < masBook.Count(); i++)
    if (masBook[i].year > lastYear)
    {
        lastYear = masBook[i].year;
        index = i;
    }
Console.Write("Серед них найсвіжіша: ");
masBook[index].Info();
Console.ReadLine();

```

2. Скласти програму для дозапису даних у двійковий файл послідовного доступу, створений в результаті виконання першої програми попереднього завдання. Після дозапису даних у файл знову виконати програму обробки його даних (тобто другу програму попереднього завдання). Порівняйте розміри файлів, створених програмами попередньої та цієї лабораторної роботи.

З питань тиражування та використання цього видання звертайтеся на shpotko@ukr.net

Лабораторна робота № 23

Тема. Обробка двійкових файлів та потоків з структурованими даними.

Мета. Формування вмінь і навиків створення та використання файлів з структурованими даними. Закріплення вмінь і навиків використання функцій вводу-виводу та обробки рядків. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання

1. Коли та з якою метою доцільно використовувати файли з структурованими даними в програмуванні?
2. У чому переваги та недоліки файла з структурованими даними стосовно файлів послідовного доступу?
3. Що відбувається на рівні операційної системи при відкритті файла?
4. Для чого визначається структура запису?
5. Як підрахувати кількість байтів поля-рядка у структурованому файлі?
6. Як визначити зміщення у структурованому файлі для зчитування значення вказаного поля визначеного запису?
7. Як закрити файл? Що відбувається на рівні операційної системи при закритті файла?

Завдання

1. **Скласти дві програми: першу – для створення структурованого двійкового файла довільного доступу, який зберігатиме дані списку по варіанту, другу - для опрацювання даних, записаних у цей файл.**

Вимоги до програми:

- значення даних першої програми ввести з клавіатури з відповідною перевіркою коректності типів та даних;
- перед відкриттям файла для обробки в другій програмі передбачити перевірку існування файла. При відсутності файла видати відповідне повідомлення та завершити виконання програми;
- результати виконання другої програми вивести в одному діалоговому вікні.

Варіанти

1. Список студентів містить їх прізвища, імена та по батькові. Вияснити, чи є в даному не впорядкованому за алфавітом списку студентів задане прізвище. Якщо є, то вказати порядкові номери таких прізвищ, відповідні імена та по батькові;
2. Список студентів містить їх прізвища, імена та по батькові. Вияснити, чи є в даному не впорядкованому за алфавітом списку студентів однофамільці. При виявленні однофамільців надрукувати їх дані;
3. Список студентів містить їх прізвища, імена, по батькові та абрєвіатури груп. Визначити, в яких групах зустрічається прізвище *Поліщук* (списки груп не впорядковані за алфавітом);

4. Заданий список, що містить перелік прізвищ студентів і назви вулиць, на яких вони проживають. Визначити, на якій з вулиць пропиває найбільша кількість студентів;
5. Список студентів містить їх прізвища, імена та по батькові. Визначити, яке ім'я зустрічається найчастіше;
6. Список містить прізвища студентів і назви улюблених предметів кожного з них. Визначити, який з предметів вказаний найбільшу кількість разів і для скількох студентів він є улюбленим;
7. Список містить прізвища студентів та суми отриманих балів кожним з них. Впорядкувати список за спаданням суми отриманих балів;
8. Список містить перелік футбольних команд та кількість очок, набраних у чемпіонаті кожною з них. Вказати назву команди-чемпіона. При наявності кількох команд з максимальною кількістю набраних очок надрукувати назви всіх таких команд;
9. Список співробітників містить їх прізвища, імена, по батькові і номери їх домашніх телефонів. Знайти телефон співробітника за прізвищем або вказати на відсутність даних у списку. Якщо для співробітника зазначено декілька телефонів, то вивести їх всі;
10. Заданий список, що містить перелік прізвищ студентів і назви вулиць, на яких вони проживають. Вказати, чи є серед студентів такі, що проживають по вулиці *Соборній*;
11. Заданий список, що містить перелік прізвищ студентів і назви вулиць, на яких вони проживають. Вказати кількість і прізвища студентів, що проживають по вулиці *Дубенській*;
12. Список містить відомості про автомобілі: їх марки, номери і прізвища власників. Вказати кількість автомобілів даної марки;
13. Список містить відомості про автомобілі: їх марки, номери і прізвища власників. Вказати прізвища власників і номери автомобілів даної марки;
14. Список містяться відомості про кожну книгу: прізвище автора, рік та назву видання. Знайти назви книг даного автора;
15. Список містить дані про кожну партію експортованих товарів: назву, країну, що імпортує товар та об'єм партії, що поставляється в штуках. Вказати загальний об'єм експорту даного товару.
16. Список містить дані про книги з програмування (прізвище автора і його ініціали, назва книги, рік видання). Вивести на екран дані останньої виданої книги;

Наприклад, програма для формування файлу завдання варіанту № 16 може мати вигляд:

```
Book[] masBook = new Book[] { new Book { name = "Изучаем С#", author = "Ендрю Стиллмен, Дженифер Грин" },
                               new Book("Кобзар", "Тарас Григорович Шевченко", 1840),
                               new Book(true, new Book(false), new Book(), book,book1,book2);

const string plusStr = " ";
int lenZap = 206;
FileStream f = new FileStream("I:\\LR20.dat", FileMode.Create);
BinaryWriter w = new BinaryWriter(f, Encoding.Unicode);
```

```

int i;
for (i = 0; i < masBook.Count(); i++)
{w.Write((masBook[i].author+plusStr).Substring(0,50));
 Console.WriteLine(w.BaseStream.Position);
 w.Write((masBook[i].name+plusStr).Substring(0,50));
 Console.WriteLine(w.BaseStream.Position);
 w.Write(masBook[i].year);
 Console.WriteLine(w.BaseStream.Position);
}
w.Close();

```

А програма для розв'язання цього завдання може бути такою:

```

f = new FileStream("I:\\LR20.dat", FileMode.OpenOrCreate);
BinaryReader r = new BinaryReader(f,Encoding.Unicode);
int countZap=(int)r.BaseStream.Length/lenZap;
if (countZap == 0)
{Console.WriteLine("Дані книг відсутні");
 Console.ReadLine();
}
r.BaseStream.Seek(202,SeekOrigin.Begin); //позиціонуємо на початок в базовому потоці
int index = 0, year, lastYear = r.ReadInt32();
for (i = 1; i < countZap; i++)
{r.BaseStream.Seek(i*lenZap+202,SeekOrigin.Begin); //позиціонуємо на початок в базовому потоці
 year = r.ReadInt32();
 if (year > lastYear)
 {lastYear = year;
  index = i;
 }
}
//зчитуємо дані найсвіжішої книги
r.BaseStream.Seek(index*lenZap,SeekOrigin.Begin); //позиціонуємо на початок в базовому потоці
book.author=r.ReadString().TrimEnd();
book.name = r.ReadString().TrimEnd();
book.year = r.ReadInt32();
r.Close();
Console.Write("Дані найсвіжішої книги: ");
book.Info();
Console.ReadLine();

```

2. Скласти програму для допису даних у структурований двійковий файл довільного доступу, створений в результаті виконання першої програми попереднього завдання. Після допису даних у файл знову виконати програму обробки його даних (тобто другу програму попереднього завдання). Порівняйте розміри файлів, створених програмами попередньої та цієї лабораторної роботи.

Рекомендована література

1. Ковалюк Т. В. Алгоритмізація та програмування: Підручник. Львів: Магнолія-2006, 2019. 400 с.
2. Ришковець Ю. В., Висоцька В. А. Алгоритмізація та програмування: Навчальний посібник. Львів: Новий світ 2000, 2020. Ч. 1 – 337 с., Ч. 2 – 314 с.
3. Троелсен Ендрю, Джеккс Філіп. Мова програмування C# 7.0 та платформи .NET і .NET Core. 8-е вид. Київ: Діалектика, 2020. 656 с. Рос. мовою.
4. Стиллмен Ендрю, Грин Дженнифер. Изучаем C#. 3-е изд. СПб.: Питер, 2014. 816 с.
5. Кормен Т. Х. Лейзерсон Ч., Ріверс Р., Штайн К. Алгоритми: побудова й аналіз. Київ: Діалектика, 2020. 1324 с.
6. Джозеф Албахари, Бен Албахари. C# 6.0. Справочник. Полное описание языка: Пер. с англ. Москва: Вильямс, 2017. 1040 с.
7. С. А. Абрамов, Г. Г. Гнездилова, Е. Н. Капустина и др. Задачи по программированию. Москва: Наука, 2000. 596 с.

З питань тиражування та використання цього видання звертайтеся на e-mail book@shpntko.com

Для нотаток

З питань тиражування та використання цього видання звертайтеся на e-mail books_shrotko@ukr.net

Для нотаток

З питань тиражування та використання цього видання звертайтеся на e-mail books_sprotko@ukr.net

Навчальне видання

Основи програмування. Лабораторний практикум для студентів денної та заочної форм навчання освітньої програми «Інженерія програмного забезпечення Інтернету речей» першого (бакалаврського) рівня вищої освіти за спеціальністю 121 Інженерія програмного забезпечення галузі знань 12 Інформаційні технології

Укладач

Олександр Володимирович Шпортко

Відповідальний за випуск доц. Ю. Г. Лютюк
Комп'ютерна верстка О. В. Шпортко

Підписано до друку 18.05.2022.

Формат 60x84 1/16. Папір офсетний № 1.

Умовн. друк. арк. 5,1. Наклад 50 примірників.

Віддруковано засобами оперативної поліграфії
ПВНЗ "Міжнародний економіко-гуманітарний університету
імені академіка Степана Дем'янчука".
м. Рівне, вул. С. Дем'янчука, 4.