

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА
ІНСТИТУТ МАТЕМАТИКИ, ЕКОНОМІКИ І МЕХАНІКИ

Л. А. Косирева

**ЗАВДАННЯ ТА МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних робіт з навчальної
дисципліни «Комп'ютерна математика»**

для студентів I курсу напряму підготовки 6.040202 «Механіка»

Одеса
ОНУ
2013

Рекомендовано до друку Вченою радою ІМЕМ ОНУ.
Протокол № 1 від 28 вересня 2012 р.

Рецензенти:

А. В. Камєнєва, кандидат технічних наук, доцент кафедри математичного забезпечення комп'ютерних систем ІМЕМ ОНУ імені І. І. Мечникова;

В. Е. Волков, кандидат фізико-математичних наук, доцент, завідувач кафедрою комп'ютерних систем та управління бізнес-процесами Одеської національної академії харчових технологій.

Косирева Л. А. Завдання та методичні вказівки до виконання лабораторних робіт з навчальної дисципліни «Комп'ютерна математика» (для студентів І курсу спеціальності «Механіка») / Косирева Л. А. - Одеса, «Одеський національний університет імені І. І. Мечникова», 2013. - 44 с.

Зміст

1. <i>Лабораторна робота 1.</i> Програмування алгоритмів з використанням функцій. Створення методів	4
2. <i>Лабораторна робота 2.</i> Обробка рядків.....	9
3. <i>Лабораторна робота 3.</i> Пошук рядка, що задовольняє заданій умові.....	18
4. <i>Лабораторна робота 4.</i> Рядки і текстові файли.....	20
5. <i>Лабораторна робота 5.</i> Робота з текстовими файлами	23
6. <i>Лабораторна робота 6.</i> Робота з бінарними файлами	23
7. <i>Лабораторна робота 7.</i> Створення списків і робота з ними.....	27
8. <i>Лабораторна робота 8.</i> Створення і використання делегатів	33
9. <i>Лабораторна робота 9.</i> Створення класів	35
Список використаної літератури.....	42

Лабораторна робота № 1

Програмування алгоритмів з використанням функцій. Створення методів

Вказівки до виконання роботи.

Ця робота виконується з використанням функцій (методів), створених програмістом.

Формат запису методу наступний:

```
[доступ] тип_повернення ім'я  
(список_параметрів)  
{  
    // тіло методу  
}
```

Тут елемент **доступ** означає модифікатор доступу, який визначає, які частини програми можуть отримати доступ до методу. Модифікатор доступу необов'язковий. Якщо він не вказаний, мається на увазі, що метод закритий (**private**) в рамках класу, де він визначений. Для того, щоб метод міг бути викликаний з будь-якої частини програми, він має бути оголошений як **public**.

За допомогою елемента **тип_повернення** вказується тип значення, поверненого методом. Це може бути будь-який допустимий тип, включаючи типи класів, що створюються програмістом. Якщо метод не повертає ніякого значення, необхідно вказати тип **void**. Ім'я методу задається елементом **ім'я**. В якості імені методу можна використати будь-який допустимий ідентифікатор, відмінний від тих, які вже використані для інших елементів програми в межах поточної зони видимості. Елемент **список_параметрів** є послідовністю пар (складених з типу даних та ідентифікатора), розділених комами. Параметри - це змінні, які набувають значень аргументів, що передаються методу при виклику. Якщо метод не має параметрів, **список_параметрів** залишається порожнім.

Для виклику функції в простому випадку треба вказати її ім'я, за яким в круглих дужках через кому перераховуються імена аргументів, які передаються.

Варіанти завдань.

1. Задані натуральні числа n і m , дійсні числа $a_1, \dots, a_n, b_1, \dots, b_m$. У послідовності a_1, \dots, a_n і в послідовності b_1, \dots, b_m усі члени, що

йдуть за членом з найбільшим значенням (за першим по порядку, якщо їх декілька), замінити на 0,5.

2. Задані натуральні числа n і m , цілі числа $a_1, \dots, a_n, b_1, \dots, b_m, k$. Якщо в послідовності a_1, \dots, a_n немає жодного члена зі значенням k , то перший по порядку член цієї послідовності, не менший усіх інших членів, замінити на значення k . За таким же правилом перетворити послідовність b_1, \dots, b_m стосовно значення 10.

3. Задано натуральне число n , цілі числа a_1, \dots, a_n . Розглянути відрізки послідовності a_1, \dots, a_n (послідовності тих членів, що йдуть підряд), що складаються з повних квадратів. Отримати найбільшу з довжин даних відрізків. Описати функцію, що дозволяє розпізнавати повні квадрати.

4. Задано натуральне число n , цілі числа a_1, \dots, a_n . Розглянути відрізки послідовності a_1, \dots, a_n (послідовності тих членів, що йдуть підряд), що складаються з мір числа п'ять. Отримати найбільшу з довжин даних відрізків. Описати функцію, що дозволяє розпізнавати міру числа п'ять.

5. Задано натуральне число n , цілі числа a_1, \dots, a_n . Розглянути відрізки послідовності a_1, \dots, a_n (послідовності тих членів, що йдуть підряд), що складаються з простих чисел. Отримати найбільшу з довжин даних відрізків. Описати функцію, що дозволяє розпізнавати прості числа.

6. Задано натуральне число n . Серед чисел $1, 2, \dots, n$ знайти все ті, які можна представити у вигляді суми квадратів двох натуральних чисел. Описати функцію, що дозволяє розпізнавати повні квадрати.

7. Задані дійсні числа $x_1, y_1, x_2, y_2, \dots, x_{10}, y_{10}$. Знайти периметр десятикутника, вершини якого мають відповідно координати $(x_1, y_1), (x_2, y_2), \dots, (x_{10}, y_{10})$. Описати функцію обчислення відстані між двома точками, заданими своїми координатами.

8. Описати функцію, результатом роботи якої є символ, заданий при зверненні до процедури, якщо цей символ не є буквою, і

відповідна рядкова (мала) буква інакше. Застосувати створену функцію для набору символів, що вводяться користувачем з клавіатури.

9. Описати функцію, яка замінює в початковому рядку символів усі одиниці нулями і усі нулі одиницями. Заміна повинна виконуватися, починаючи із заданої позиції рядка. Застосувати створену функцію для рядка S і номери позиції n , що вводяться користувачем з клавіатури.

10. Описати функцію, що дозволяє визначити позицію найправішого входження заданого символу в початковий рядок. Якщо рядок не містить символу, результатом роботи функції має бути -1 . Застосувати створену функцію для рядка, що вводиться користувачем з клавіатури.

11. Описати функцію, результатом роботи якої є 0 , якщо символ, заданий при зверненні до функції, - буква, і 1 інакше. Застосувати створену функцію для набору символів, що вводяться користувачем з клавіатури.

12. Описати функцію побудови рядка символів, що являється записом заданого дійсного числа в десятковій системі числення. Рядок повинен містити вказану кількість цифр після коми. Застосувати створену функцію для чисел, що вводяться користувачем з клавіатури.

13. Два натуральні числа називаються дружніми, якщо кожне з них дорівнює сумі усіх дільників іншого, за винятком його самого (такими є, наприклад, числа 220 і 284). Надрукувати усі пари дружніх чисел, що не перевершують заданого натурального числа.

14. Дано парне $n > 2$. Перевірити для цього числа гіпотезу Гольдбаха. Ця гіпотеза (по сьогоднішній день не спростована і повністю не доведена) полягає в тому, що кожне парне n , більше 2, представляється у вигляді суми двох простих чисел. Описати функцію, що дозволяє розпізнавати прості числа.

15. Дано натуральне n . З'ясувати, чи є серед чисел $n, n+1, \dots, 2n$ близнюки, тобто прості числа, різниця між якими рівна двом. Описати функцію, що дозволяє розпізнавати прості числа.

16. Дано натуральне число n , цілі числа a_1, \dots, a_n . Розглянути відрізки послідовності a_1, \dots, a_n (послідовності тих членів, що йдуть підряд), що складаються з досконалих чисел. Отримати найбільшу з довжин даних відрізків. Описати функцію, що дозволяє розпізнавати досконалі числа.
17. Описати функцію, в результаті звернення до якої з першого заданого рядка видаляється кожен символ, що належить і другому заданому рядку. Застосувати створену функцію для рядків, що вводяться користувачем з клавіатури.
18. Описати функцію, що дозволяє визначити позицію першого входження в заданий рядок якого-небудь символу з другого заданого рядка. Результатом роботи функції має бути -1 , якщо перший рядок не містить жодного символу, що належить і другому заданому рядку. Застосувати створену функцію для рядків, що вводяться користувачем з клавіатури.
19. Описати функцію обчислення значення цілого числа по заданому рядку символів, який є десятковим записом цього числа. Зробити перевірку коректності введення рядка: ввести параметр, що повертає 0 , якщо рядок символів є десятковим записом цілого числа, і -1 інакше. Застосувати створену функцію для рядка, що вводиться користувачем з клавіатури.
20. Дані натуральне n , дійсні числа $x_1, y_1, x_2, y_2, \dots, x_n, y_n$. Знайти площу багатокутника, що складається з n сторін, розбиваючи його на трикутники. Вершини багатокутника при деякому послідовному обході мають координати $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. Описати функцію обчислення площі трикутника по координатах його вершин.
21. Описати функцію побудови рядка символів, що являється записом заданого дійсного числа в десятковій системі числення. Рядок повинен містити вказану кількість цифр після коми. Застосувати створену функцію для чисел, що вводяться користувачем з клавіатури.
22. Описати функцію "стискування" початкової послідовності символів: кожна послідовність, що складається з декількох входжень одного і того ж символу, замінюється на текст $x(k)$, де x -

символ, а k - рядок, що є записом числа входжень символу x в початкову послідовність. Застосувати створену функцію для рядка символів, що вводиться користувачем з клавіатури.

23. Дані цілі числа a_1, \dots, a_n . Визначити, чи є ця послідовність періодичною (тобто чи може вона бути отримана повтореннями деякої своєї початкової частини). З усіх періодів вказати найменший.

Лабораторна робота № 2

Обробка рядків

Вказівки до виконання роботи.

Кожне завдання лабораторної роботи припускає наявність введення користувачем рядка (чи рядків) з клавіатури і використання методів класу **String**, призначених для роботи з рядками. Цей клас визначений в просторі імен **System**. Він лежить в основі вбудованого в C# типу **string**, і є частиною середовища .NET Framework.

Подібно до масивів, індексація в рядках розпочинається з нуля. У C# рядки є об'єктами, тому вони не підлягають зміні.

У класі **string** визначена єдина властивість, призначена тільки для читання:

```
public int Length {get;}
```

Властивість **Length** повертає кількість символів, що містяться у рядку.

У класі **string** визначені багато різних методів. При цьому деякі з них мають два або більше переобтяжених форматів.

У таблиці 1 показані методи, що дозволяють обробляти символи і визначати, до якої категорії вони належать.

Нижче, в таблиці 2 містяться методи, призначені для обробки рядків.

Таблиця 1

Методи класифікації символів

Перевіряє приналежність символу до певної множини символів. Повертає true, якщо умова, що перевіряється, виконується	
public static bool IsDigit (char <i>c</i>)	Цифри від 0 до 9
public static bool IsLetter (char <i>c</i>)	Буква алфавіту
public static bool IsLetterOrDigit (char <i>c</i>)	Буква алфавіту або цифра
public static bool IsUpper (char <i>c</i>)	Буква верхнього регістра (A - Z)
public static bool IsLower (char <i>c</i>)	Буква нижнього регістра (a - z)
public static bool IsPunctuation (char <i>c</i>)	Знак пунктуації
public static bool IsControl (char <i>c</i>)	Символ, що управляє
public static bool IsNumber (char <i>c</i>)	Шестнадцятиричні цифри (0-9 або A-F)
public static bool IsSeparator (char <i>c</i>)	Розділовий знак, наприклад, пробіл
public static bool IsSeparator (char <i>c</i>)	Псевдосимвол Unicode
public static bool IsSymbol (char <i>c</i>)	Символічний знак, наприклад, валютний символ
public static bool IsWhiteSpace (char <i>c</i>)	Пропуск, символ табуляції або порожній рядок

Таблиця 2

Методи, визначені в класі string

static string Copy (string <i>str</i>)	Повертає копію рядка <i>str</i>
int CompareTo (string <i>str</i>)	Повертає негативне значення, якщо рядок, який викликає, менше рядка <i>str</i> , позитивне значення, якщо рядок, який викликає, більше рядка <i>str</i> , і нуль, якщо порівнювані рядки рівні
public static int Compare (string <i>str1</i> , string <i>str2</i>)	Порівнює рядок, що адресується параметром <i>str</i> , з рядком, що адресується параметром <i>str2</i> . Повертає позитивне число, якщо рядок <i>str1</i> більше <i>str2</i> , негативне число, якщо <i>str1</i> менше <i>str2</i> , і нуль, якщо рядки <i>str1</i> і <i>str2</i> рівні
string ToLower ()	Замінює усі букви в рядку, який викликає, на рядкові

string Toupper()	Замінює усі букви в рядку, який викликає, на прописні
public static string Concat (string <i>str1</i> , string <i>str2</i>)	Повертає рядок, який містить рядок <i>str2</i> , приєднаний до кінця рядка <i>str1</i>
Пошук рядка	
public bool EndsWith (string <i>str</i>)	Повертає значення <i>true</i> , якщо рядок, який викликає, закінчується підрядком, переданим в параметрі <i>str</i> . Інакше метод повертає значення <i>false</i>
public int IndexOf (char <i>ch</i>)	Відшукує перше входження символу <i>ch</i> в рядок, який викликає, та повертає його індекс. Якщо символу <i>ch</i> в рядку немає, повертається значення <i>-1</i>
public int IndexOf (string <i>str</i>)	Відшукує перше входження підрядка <i>str</i> в рядок, який викликає, та повертає його індекс. Якщо підрядка <i>str</i> в рядку немає, повертається значення <i>-1</i>
public int IndexOf (char <i>ch</i> , int <i>start</i>)	Відшукує перше входження символу <i>ch</i> в рядок, який викликає, та повертає його індекс. Пошук розпочинається з елемента, індекс якого заданий параметром <i>start</i> . Якщо шуканий символ не виявлений, повертається значення <i>-1</i>
public int IndexOf (string <i>str</i> , int <i>start</i>)	Відшукує перше входження підрядка <i>str</i> в рядок, який викликає, та повертає його індекс. Пошук розпочинається з елемента, індекс якого заданий параметром <i>start</i> . Якщо підрядка <i>str</i> в рядку немає, повертається значення <i>-1</i>
public int IndexOf (char <i>ch</i> , int <i>start</i> , int <i>count</i>)	Повертає індекс першого входження символу <i>ch</i> у рядку, який викликає. Пошук розпочинається з елемента, індекс якого заданий параметром <i>start</i> , і охоплює <i>count</i> елементів. Метод повертає значення <i>-1</i> , якщо шуканий символ не виявлений
public int IndexOf (string <i>str</i> , int <i>start</i> , int <i>count</i>)	Повертає індекс першого входження підрядка <i>str</i> у рядку, який викликає. Пошук розпочинається з елемента, індекс якого заданий параметром <i>start</i> , і охоплює <i>count</i> елементів. Метод повертає значення <i>-1</i> , якщо шуканий підрядок не виявлений

<p><code>public int IndexOfAny(char[] a)</code></p>	<p>Повертає індекс першого входження будь-якого символу з послідовності <i>a</i>, який буде виявлений у рядку, який викликає. Повертає значення <i>-1</i>, якщо збіги символів не виявлено</p>
<p><code>public int IndexOfAny(char[] a, int start)</code></p>	<p>Повертає індекс першого входження будь-якого символу з послідовності <i>a</i>, який буде виявлений у рядку, який викликає. Пошук розпочинається з елемента, індекс якого заданий параметром <i>start</i>. Метод повертає значення <i>-1</i>, якщо збіги символів не виявлено</p>
<p><code>public int IndexOfAny(char[] a, int start, int count)</code></p>	<p>Повертає індекс першого входження будь-якого символу з послідовності <i>a</i>, який буде виявлений у рядку, який викликає. Пошук розпочинається з елемента, індекс якого заданий параметром <i>start</i>, і охоплює <i>count</i> елементів. Метод повертає значення <i>-1</i>, якщо збіги символів не виявлено</p>
<p><code>public int LastIndexOf(char ch)</code></p>	<p>Повертає індекс останнього входження символу <i>ch</i> у рядку, який викликає. Повертає значення <i>-1</i>, якщо шуканий символ не виявлений</p>
<p><code>public int LastIndexOf(string str)</code></p>	<p>Повертає індекс останнього входження підрядка <i>str</i> у рядку, який викликає. Повертає значення <i>-1</i>, якщо шуканий підрядок не виявлений</p>
<p><code>public int LastIndexOf (char ch, int start)</code></p>	<p>Повертає індекс останнього входження символу <i>ch</i>, виявленого в межах діапазону рядка, який викликає. Пошук виконується в зворотному порядку, починаючи з елемента, індекс якого заданий параметром <i>start</i>, і закінчуючи елементом з нульовим індексом. Метод повертає значення <i>-1</i>, якщо шуканий символ не виявлений</p>
<p><code>public int LastIndexOf (string str, int start)</code></p>	<p>Повертає індекс останнього входження підрядка <i>str</i>, виявленого в межах діапазону рядка, який викликає. Пошук виконується в зворотному порядку, починаючи з елемента, індекс якого заданий параметром <i>start</i>, і закінчуючи елементом з нульовим індексом. Метод повертає значення <i>-1</i>, якщо шуканий підрядок не виявлений</p>

<p><code>public int LastIndexOf(char <i>ch</i>, int <i>start</i>, int <i>count</i>)</code></p>	<p>Повертає індекс останнього входження символу <i>ch</i>, виявленого в межах діапазону рядка, який викликає. Пошук виконується в зворотному порядку, починаючи з елемента, індекс якого заданий параметром <i>start</i>, і охоплюючи <i>count</i> елементів. Метод повертає значення <i>-1</i>, якщо шуканий символ не виявлений</p>
<p><code>public int LastIndexOf (string <i>str</i>, int <i>start</i>, int <i>count</i>)</code></p>	<p>Повертає індекс останнього входження підрядка <i>str</i>, виявленого в межах діапазону рядка, який викликає. Пошук виконується в зворотному порядку, починаючи з елемента, індекс якого заданий параметром <i>start</i>, і охоплюючи <i>count</i> елементів. Метод повертає значення <i>-1</i>, якщо шуканий підрядок не виявлений</p>
<p><code>public int LastIndexOfAny (char[] <i>a</i>)</code></p>	<p>Повертає індекс останнього входження будь-якого символу з послідовності <i>a</i>, який буде виявлений в рядку, який викликає. Повертає значення <i>-1</i>, якщо збіги символів не виявлено</p>
<p><code>public int LastIndexOfAny (char[] <i>a</i>, int <i>start</i>)</code></p>	<p>Повертає індекс останнього входження будь-якого символу з послідовності <i>a</i>, який буде виявлений в рядку, який викликає. Пошук виконується в зворотному порядку, починаючи з елемента, індекс якого заданий параметром <i>start</i>, і закінчуючи елементом з нульовим індексом. Повертає значення <i>-1</i>, якщо збіги символів не виявлено</p>
<p><code>public int LastIndexOfAny (char[] <i>a</i>, int <i>start</i>, int <i>count</i>)</code></p>	<p>Повертає індекс останнього входження будь-якого символу з послідовності <i>a</i>, який буде виявлений в рядку, який викликає. Пошук виконується в зворотному порядку, починаючи з елемента, індекс якого заданий параметром <i>start</i>, і охоплюючи <i>count</i> елементів. Метод повертає значення <i>-1</i>, якщо збіги символів не виявлено</p>
<p><code>public bool StartsWith(string <i>str</i>)</code></p>	<p>Повертає значення <i>true</i>, якщо рядок, який викликає, починається рядком, переданим в параметрі <i>str</i>. Інакше метод повертає значення <i>false</i></p>

Розбиття і складання рядків	
public string[] Split (params char[] <i>seps</i>)	Метод призначений для розбиття рядка, який викликає, на підрядки, які повертаються методом у вигляді строкового масиву. Символи, які відділяють підрядки один від іншого, передаються в масиві <i>seps</i> . Якщо параметр <i>seps</i> містить <i>null</i> - значення як роздільник підрядків, використовується пропуск.
public string[] Split (params char[] <i>seps</i> , int <i>count</i>)	Метод відрізняється від попереднього тим, що обмежує кількість підрядків, які повертаються значенням <i>count</i> .
public static string Join (string <i>sep</i> , string[] <i>strs</i> , int <i>start</i> , int <i>count</i>)	Метод повертає рядок, який містить конкатеновані рядки, передані в масиві <i>strs</i> . Кожен рядок, що становить результат, відділяється від наступного розділовим рядком, заданим параметром <i>sep</i> .
public static string Join (string <i>sep</i> , string[] <i>strs</i>)	Другий формат дозволяє зібрати рядок, який міститиме <i>count</i> конкатенованих рядків, переданих в масиві <i>strs</i> , починаючи з рядка <i>strs[start]</i> . Кожен рядок, що становить результат, відділяється від наступного розділовим рядком, заданим параметром <i>sep</i> .
Видалення символів і доповнення ними рядків	
public string Trim ()	Призначений для видалення початкових і кінцевих пробелів із рядка, який викликає. Повертається рядок, що містить результат цієї операції.
public string Trim (params char[] <i>chrs</i>)	Дозволяє видалити початкові і кінцеві символи, задані параметром <i>chrs</i> . Повертається рядок, що містить результат цієї операції.
public string PadLeft (int <i>len</i>)	Призначений для доповнення рядка з лівого боку пробілами в такій кількості, щоб загальна довжина рядка, який викликає, стала рівною заданому значенню <i>len</i> . Повертається рядок, що містить результат цієї операції.
public string PadLeft (int <i>len</i> , char <i>ch</i>)	Відрізняється від попередньої тим, що для доповнення рядка замість пробілу використовується символ, заданий параметром <i>ch</i> . Повертається рядок, що містить результат цієї операції.
public string PadRight (int <i>len</i>)	Доповнює рядок з правого боку пробілами в такій кількості, щоб загальна довжина рядка, який викликає, стала рівною заданому значенню <i>len</i> . Повертається рядок, що містить результат цієї операції.
public string	Відрізняється від попередньої тим, що для

PadRight (int <i>len</i> , char <i>ch</i>)	доповнення рядка замість пробілу використовується символ, заданий параметром <i>ch</i> . Повертається рядок, що містить результат цієї операції.
Вставка, видалення і заміна	
public string Insert (int <i>start</i> , string <i>str</i>)	У рядок, який викликає, вставляється рядок <i>str</i> , починаючи з індексу <i>start</i> .
public string Remove (int <i>start</i> , int <i>count</i>)	Видаляє з рядка <i>count</i> символів, починаючи з позиції <i>start</i> .
public string Replace (char <i>ch1</i> , char <i>ch2</i>)	Дозволяє замінити в рядку, який викликає, усі входження символу <i>ch1</i> символом <i>ch2</i> .
public string Replace (string <i>str1</i> , string <i>str2</i>)	Служить для заміни в рядку, який викликає, усіх входжень рядка <i>str1</i> рядком <i>str2</i> .
public string Substring (int <i>idx</i>)	Повертається підрядок, починаючи з позиції <i>idx</i> , і до кінця рядка, який викликає,.
public string Substring (int <i>idx</i> , int <i>count</i>)	Повертається підрядок, починаючи з позиції <i>idx</i> , і що складається з <i>count</i> символів

Завдання

1. Заданий рядок, що містить декілька слів, розділених пробілами. Необхідно надрукувати кількість слів в рядку і довжину кожного слова.
2. Заданий рядок, що містить декілька слів, розділених пробілами. Необхідно надрукувати усі слова, що розпочинаються з гласної букви.
3. Заданий рядок, що містить декілька слів, розділених пробілами. Необхідно видалити з рядка слова, що складаються з *n* букв. Число *n* задається користувачем.
4. Заданий рядок, що містить декілька слів, розділених пробілами. Необхідно видалити з рядка слово з найбільшим входженням заданої користувачем букви.
5. Перевірте, чи можна з букв, що входять в слово А, побудувати слово В. Букви початкового слова можна переставляти, але не можна кожену букву використати більше одного разу.

6. Заданий рядок, що містить декілька слів, розділених пробілами. Необхідно обернути кожне слово рядка.
7. Підрахувати, скільки разів цифра 7 зустрічається в десятковому запису заданого чотиризначного числа n .
8. Визначити найбільшу з цифр, використуваних в десятковому запису заданого чотиризначного натурального числа n .
9. Заданий рядок, що містить декілька слів, розділених пробілами. Необхідно провести наступне перетворення рядка: якщо слово розпочинається із заданої користувачем букви, переставити її в кінець слова.
10. Автоморфними називаються числа, які містяться в останніх розрядах їх квадрата. Наприклад, $5^2 = 25$, $25^2 = 625$. Складіть алгоритм для знаходження декількох автоморфних чисел.
11. Заданий рядок, що містить декілька слів, розділених пробілами, і має як рядкові, так і прописні букви. Необхідно перетворити усі рядкові букви в прописні і вивести на друк слово, в якому зроблено максимальну кількість перетворень.
12. З'ясувати, чи вірно, що в заданому рядку є усі букви, що входять в задане користувачем слово.
13. Заданий рядок символів. З'ясувати, чи вірно, що в цьому рядку більше груп букв, ніж груп знаків.
14. Заданий рядок символів. Видалити з кожної групи цифр, що йдуть підряд, і в якій більше двох цифр та якій передуює точка, усі цифри, починаючи з третьої (наприклад, $ab+0.1973-1.1$ перетворюється в $ab+0.19-1.1$).
15. У рядку є слова трьох типів: що складаються тільки з букв, складаються тільки з цифр, складаються із суміші букв і цифр. Підрахувати кількість слів, що відносяться до кожного типу.

16. Заданий рядок, що містить декілька слів, розділених пробілами. Надрукувати слово, яке містить максимальну кількість заданої користувачем букви.
17. Заданий рядок символів. Видалити з нього усі символи, що не є буквами та розділовими знаками.
18. Заданий рядок символів. Перетворити його, видаливши з кожної групи цифр, якій не передує точка, усі початкові нулі (окрім останнього, якщо за ним йде точка).
19. Заданий рядок, що містить декілька слів, розділених пробілами. Видалити з неї слова, що зустрічаються більше одного разу.
20. Заданий рядок, що складається з декількох слів. З'ясувати, чи є в цьому рядку два слова, кожне з яких виходить перевертанням іншого.
21. Заданий рядок, що складається з декількох слів. Знайти найбільшу довжину слів-паліндромів, якщо вони є.
22. Заданий текст російською мовою, записаний рядковими буквами. Слова розділені пробілами або розділовими знаками. Визначити частоту входження кожного символу.

Лабораторна робота № 3

Пошук рядка, що задовольняє заданій умові

Вказівки до виконання роботи.

Кожне завдання лабораторної роботи припускає наявність введення користувачем рядків з клавіатури. Обробка довільного рядка згідно з умовою завдання повинна здійснюватися за допомогою створеного студентом методу.

Дані 3 рядки. Знайти:

1. В якому з них знаходиться щонайдовше слово?
2. В якому з них більше слів, що містять цифри?
3. В якому з них кожне парне слово більше за кожне непарне?
4. В якому з них більше слів однієї довжини?
5. В якому з них непарні слова відсортовані по неубуванню?
6. В якому з них слова розташовані по наростанню їх довжини?
7. В якому з них більше слів, довжина яких дорівнює заданому користувачем числу?
8. В якому з них слово, що не містить задану користувачем букву, має найбільшу довжину?
9. В якому з них слова відсортовані за збільшенням?
10. В якому з них більше слів з різною сумою кодів символів?
11. В якому з них немає однакових слів?
12. В якому з них більше слів, що завершуються заданою користувачем буквою?
13. В якому з них більше слів, код третьої букви яких дорівнює заданому користувачем значенню?

14. В якому з них більше слів, що містять задану користувачем букву не менше k раз?
15. В якому з них слово, що містить дві задані користувачем букви, має найбільшу довжину?
16. В якому з них знаходиться слово, сума кодів символів якого максимальна?
17. В якому з них довжина k -го слова більше довжини слова з номером n ?
18. В якому з них менше слів, що не мають двох заданим користувачем букв?
19. В якому з них більше слів, що розпочинаються із заданої користувачем букви, і що завершуються буквою, код якої більше заданої величини на n ?
20. В якому з них менше слів з однаковою сумою кодів їх символів?

Лабораторна робота № 4

Рядки і текстові файли

Вказівки до виконання роботи.

Кожне завдання лабораторної роботи припускає наявність текстового файлу, що містить по одному реченню в кожному рядку. Кількість рядків у файлі довільна.

Для читання тексту з текстового файлу в C# призначений клас **StreamReader**, визначений в просторі імен **System.IO**. Читання з файлу рядків (по одному по порядку) можна зробити за допомогою оператора:

```
while (( s = f.ReadLine()) != null) { ... }
```

Варіанти завдань.

1. Написати програму, яка прочитує з текстового файлу три речення і виводить їх в зворотному порядку.
2. Написати програму, яка прочитує текст з файлу і виводить на екран тільки речення, що містять введене з клавіатури слово.
3. Написати програму, яка прочитує текст з файлу і виводить на екран тільки ті рядки, що містять двозначні числа.
4. Написати програму, яка прочитує англійський текст з файлу і виводить на екран слова, що розпочинаються з гласних букв.
5. Написати програму, яка прочитує текст з файлу і виводить його на екран, міняючи місцями кожні два сусідні слова.
6. Написати програму, яка прочитує текст з файлу і виводить на екран тільки речення, що не містять ком.
7. Написати програму, яка прочитує текст з файлу і визначає, скільки в нім слів, що складаються з не більше ніж чотирьох букв.
8. Написати програму, яка прочитує текст з файлу і виводить на екран тільки цитати, тобто. речення, поміщені в лапки.

9. Написати програму, яка прочитує текст з файлу і виводить на екран тільки речення, що складаються із заданої кількості слів.
10. Написати програму, яка прочитує англійський текст з файлу і виводить на екран слова тексту, що починаються і закінчуються на гласні букви.
11. Написати програму, яка прочитує текст з файлу і виводить на екран тільки рядки, що не містять двозначних чисел.
12. Написати програму, яка прочитує текст з файлу і виводить на екран тільки речення, що розпочинаються з тире, перед яким можуть знаходитися тільки пробільні символи.
13. Написати програму, яка прочитує англійський текст з файлу і виводить його на екран, замінивши кожен першу букву слів, що розпочинаються з гласної букви, на прописну.
14. Написати програму, яка прочитує текст з файлу і виводить його на екран, замінивши цифри від 0 до 9 на слова " нуль", " один", ..., "дев'ять", розпочинаючи кожне речення з нового рядка.
15. Написати програму, яка прочитує текст з файлу, знаходить щонайдовше слово і визначає, скільки разів воно зустрілося в тексті.
16. Написати програму, яка прочитує текст з файлу і виводить на екран спочатку питальні, а потім окличні речення.
17. Написати програму, яка прочитує текст з файлу і виводить на екран, додаючи після кожного речення, скільки разів трапилося в ній введене з клавіатури слово.
18. Написати програму, яка прочитує текст з файлу і виводить на екран усі його речення в зворотному порядку.
19. Написати програму, яка прочитує текст з файлу і виводить на екран спочатку речення, що починаються з одного буквених слів, а потім усі інші.

20. Написати програму, яка прочитає текст з файлу і виводить на екран речення, що містять максимальну кількість знаків пунктуації.
21. Написати програму, яка прочитає текст з файлу і виводить на екран перший з найкоротших його рядків.

Лабораторна робота № 5

Робота з текстовими файлами

Створити файл, що містить базу даних з відомостями про студентів групи, вказавши в якості характеристик прізвище, ім'я та по батькові, рік народження, адресу, номер телефону, місце проживання, колір очей чи які-небудь інші (4 - 5 пунктів по вибору). Для зберігання параметрів кожного студента використати тип даних "структура". Для збереження записів використати файли класу **FileStream**.

Програма повинна вміти розсортувати елементи файлу по одній з ознак. Передбачити можливість додавання в список нових елементів.

Лабораторна робота № 6

Робота з бінарними файлами

Вказівки до виконання роботи.

Програма на ЕОМ цієї лабораторній роботі повинна містити наступні пункти:

1. Створення бінарного файлу програмним шляхом і можливість занесення або додавання інформації в цей файл користувачем з клавіатури. Цей пункт припускає наявність інтерфейсу для роботи користувача з клавіатурою.
2. Роботу з файлом згідно з постановкою завдання.
3. Запис результатів роботи програми в файл.
4. Виведення результатів роботи програми на екран у вигляді таблиці.

Бінарні файли на мові C# створюються за допомогою класів **BinaryWriter** і **BinaryReader**.

Варіанти завдань

1. Створити файл, що містить відомості про деяку кількість кубиків: розмір кожного кубика (довжина ребра в сантиметрах), його колір (червоний, жовтий, зелений або синій) і матеріал (дерев'яний,

металевий, картонний). Знайти кількість кубиків кожного з перерахованих кольорів і їх сумарний об'єм.

2. Створити файл, що містить відомості про речовини: вказується назва речовини, його питома вага і провідність (провідник, напівпровідник, ізолятор). Знайти питомі ваги і назви усіх напівпровідників.
3. Створити файл, що містить різні дати. Кожна дата - це число, місяць і рік. Знайти усі весняні дати.
4. Відомості про автомобіль складаються з його марки, номера і прізвища власника. Створити файл, що містить відомості про декілька автомобілів. Знайти прізвища власників і номери автомобілів марки, що цікавить користувача.
5. Відомості про учня складаються з його імені, прізвища і назви класу (року навчання і букви), в якому він вчиться. Створити файл, що вміщає відомості про учнів школи. З'ясувати, чи є в школі учні з однаковими прізвищами.
6. Створити файл KURS1, що містить відомості про студентів першого курсу групи механіки: прізвище, ім'я, по батькові студента, екзаменаційні оцінки по алгебрі, математичному аналізу, програмуванню за 4-хбальною шкалою. Написати програму, яка залишає у файлі KURS1 відомості тільки про тих студентів, які успішно склали усі іспити, і виводить на друк відомості про студентів, що мають хоч би одну заборгованість: друкує їх прізвища і ініціали та кількість не складених іспитів.
7. Відомості про автомобіль складаються з його марки, номера і прізвища власника. Створити файл, що містить відомості про декілька автомобілів. Знайти кількість автомобілів кожної марки.
8. Створити файл, що містить відомості про кубики: розмір кожного кубика (довжина ребра в сантиметрах), його колір (червоний, жовтий, зелений або синій) і матеріал (дерев'яний, металевий, картонний). Знайти кількість дерев'яних кубиків з ребром 3 см і кількість металевих кубиків з ребром, більшим, ніж 5 см.

9. Описати функцію $Zap(d, s, w)$ з трьома параметрами - імена файлів, що містять дати, які складаються з числа і місяця. Функція повинна з файлу d переписувати у файл s усі літні дати, а у файл w - усі зимові дати. Створити файл, що містить дати днів народження студентів своєї групи і розсортувати їх за допомогою вищеописаної функції.
10. Створити файл, що вміщає відомості про речовини: вказується назва речовини, його питома вага і провідність (провідник, напівпровідник, ізолятор). Вибрати дані про провідників і упорядкувати їх по убутанню питомих ваги.
11. Створити файл, що вміщає наступну інформацію: покупець, товар, кількість одиниць, вартість одиниці товару. Скласти таблицю, що містить відомості про вибрані користувачем товари: кількість проданих одиниць і їх загальну вартість, і вивести ці відомості у файл.
12. Відомості про учня складаються з його імені, прізвища і назви класу (року навчання і букви), в якому він вчиться. Створити файл, що вміщає відомості про учнів школи. З'ясувати, чи є учні з однаковими прізвищами в яких-небудь паралельних класах.
13. Відомості про учня складаються з його імені, прізвища і назви класу (року навчання і букви), в якому він вчиться. Створити файл, що вміщає відомості про учнів школи. З'ясувати, чи є в школі учні, у яких співпадають і ім'я, і прізвище.
14. Створити файл, що містить наступну інформацію: покупець, товар, кількість одиниць, вартість одиниці товару. Скласти таблицю, що вміщає відомості про вибраних користувачем покупців: товар, скільки куплено одиниць і загальну вартість купівлі, і вивести ці відомості у файл.
15. Відомості про учня складаються з його імені, прізвища і назви класу (року навчання і букви), в якому він вчиться. Створити файл, що вміщає відомості про учнів школи. З'ясувати, чи є учні з однаковими прізвищами в якому-небудь класі.

16. Відомості про учня складаються з його імені, прізвища і назви класу (року навчання і букви), в якому він вчиться. Створити файл, що вміщає відомості про учнів школи. З'ясувати, на скільки чоловік у восьмих класах більше, ніж в десятих.
17. Створити файл, що містить різні дати. Кожна дата - це число, місяць і рік. Знайти найпізнішу дату.

Лабораторна робота № 7

Створення списків і робота з ними

Деякі теоретичні відомості.

При програмуванні на будь-якій мові широко застосовуються списки, ґрунтовані на послідовному і на пов'язаному розподілі пам'яті. При послідовному розподілі елементи списку (вузли) розміщуються послідовно, один за іншим. При пов'язаному розподілі пам'яті місце розташування кожного елемента заздалегідь невідоме - блок пам'яті, який відводиться для розміщення окремого елемента, виділяється з однієї більшої області пам'яті по спеціальних алгоритмах. Найчастіше використовуються наступні однозв'язні списки:

Стік - лінійний список, в якому усі включення і виключення (і зазвичай всякий доступ) робляться на одному кінці списку. Механізм функціонування стека добре відбитий в іншій його назві - список типу LIFO (last in first out - "останнім увійшов - першим вийшов"). Схематично його побудову можна зобразити таким чином:



Черга - лінійний список, в якому усі включення робляться на одному кінці списку, а усі виключення (і зазвичай всякий доступ) робляться на іншому його кінці. Це список типу FIFO (first in first out - "першим увійшов - першим вийшов").



У мові C# для побудови списків можна використовувати клас **List**.

Варіанти завдань

1. Дані послідовності дійсних чисел x_1, x_2, \dots та p_1, p_2, \dots (елементи списку визначаються користувачем в процесі введення, їх кількість заздалегідь не відома), що визначають систему матеріальних точок на прямій: x_i - координата, p_i - вага i -тої точки ($i = 1, 2, \dots$).

Вказати номер точки, найближче розташованої до центру тяжіння системи. Якщо таких точок декілька, то узяти будь-яку з них.

2. "Лічилка". Дані натуральні n, m . Передбачається, що n чоловік встають в круг і отримують номери, які призначаються проти годинникової стрілки, $1, 2, \dots, n$. Потім, починаючи з першого, також проти годинникової стрілки, відлічується m -а людина (оскільки люди стоять по колу, то за n -ю людиною йде перша). Ця людина виходить з колу, після чого, починаючи з наступної, знову відлічується m -а людина і так до тих пір, поки з усього колу не залишається одна людина. Визначити її номер.
3. Створити список, що складається з символів s_1, s_2, \dots (елементи списку визначаються користувачем в процесі введення, їх кількість заздалегідь не відома). Створити новий список, що складається з символів, які входять в початковий список по одному разу, прочитати і вивести його вміст на екран.
4. Створити список з цілих чисел a_1, a_2, \dots (елементи списку визначаються користувачем в процесі введення, їх кількість заздалегідь не відома). Необхідно отримати послідовність $x_1, y_1, x_2, y_2, \dots, x_k, y_k$, де x_1, \dots, x_m - узяті в порядку дотримання парні члени початкової послідовності, y_1, \dots, y_p - непарні члени, де $k = \min(m, p)$.
5. Створити список, що складається з символів s_1, s_2, \dots (елементи списку визначаються користувачем в процесі введення, їх кількість заздалегідь не відома). Створити новий список, що містить тільки останні входження до нього кожного символу зі збереженням взаємного порядку цих входжень, прочитати і вивести його вміст на екран.
6. Створити 3 списки різної довжини, що складаються з символів $s_1, s_2, \dots, t_1, t_2, \dots, u_1, u_2, \dots$ (елементи списку визначаються користувачем в процесі введення, їх кількість заздалегідь не відома). Створити новий список, що містить ті символи, які входять одночасно в усі три послідовності, прочитати і вивести його вміст на екран.

7. Створити список з цілих чисел a_1, a_2, \dots (елементи списку визначаються користувачем в процесі введення, їх кількість заздалегідь не відома). Перетворити цю послідовність, розташувавши спочатку від'ємні члени, а потім - додатні. При цьому порядок тих і інших чисел змінити на зворотний.
8. Створити список з цілих чисел a_1, a_2, \dots (елементи списку визначаються користувачем в процесі введення, їх кількість заздалегідь не відома). Перетворити цю послідовність, розташувавши спочатку від'ємні члени, а потім - додатні. При цьому порядок тих і інших чисел зберегти колишнім.
9. Створити список з цілих чисел a_1, a_2, \dots (елементи списку визначаються користувачем в процесі введення, їх кількість заздалегідь не відома). Перетворити цю послідовність, розташувавши спочатку від'ємні члени, а потім - додатні. При цьому порядок від'ємних чисел змінити на зворотний, а порядок додатних зберегти колишнім.
10. Створити список з цілих чисел a_1, a_2, \dots (елементи списку визначаються користувачем в процесі введення, їх кількість заздалегідь не відома). Перетворити цю послідовність, розташувавши спочатку від'ємні члени, а потім - додатні. При цьому порядок від'ємних чисел зберегти колишнім, а порядок додатних змінити на зворотний.
11. Заданий рядок, що містить декілька слів, розділених пробілами. Створити список, що складається із слів рядка. Видалити слово, розташоване перед щонайдовшим словом рядка.
12. Створити список з декількох елементів типу структури з ім'ям TRAIN, що містить наступні поля (елементи списку визначаються користувачем в процесі введення, їх кількість заздалегідь не відома):
 - назва пункту призначення;
 - номер потягу;
 - час відправлення.Написати програму, що виконує наступні дії:
 - упорядкувати елементи в списку за часом відправлення потягу;

- вивести на екран монітора інформацію про потяги, що спрямовуються в пункт, назва якого введена з клавіатури; якщо таких потягів немає, вивести відповідне повідомлення.

13. Створити список з декількох елементів типу структури з ім'ям EXPORT, що містить наступні поля (елементи списку визначаються користувачем в процесі введення, їх кількість заздалегідь не відома):

- назва товару;
- країна-виробник товару;
- об'єм партії, що поставляється;
- ціна одиниці товару.

Написати програму, що виконує наступні дії:

- упорядкувати записи в алфавітному порядку по країнах-виробниках;
- вивести на екран монітора інформацію про експортера, значення якого введене з клавіатури; якщо такого немає, вивести відповідне повідомлення.

14. Створити список з декількох елементів типу структури з ім'ям PHONE, що містить наступні поля (елементи списку визначаються користувачем в процесі введення, їх кількість заздалегідь не відома):

- прізвище, ім'я;
- номер телефону;
- адреса абонента.

Написати програму, що виконує наступні дії:

- упорядкувати записи по номерах телефонів;
- вивести на екран монітора інформацію про абонента, прізвище якого введене з клавіатури; якщо такого немає, вивести відповідне повідомлення.

15. Створити список з декількох елементів типу структури з ім'ям STUDENT, що містить наступні поля (елементи списку визначаються користувачем в процесі введення, їх кількість заздалегідь не відома):

- прізвище і ініціали;
- номер групи;
- успішність (масив з 5 елементів).

Написати програму, що виконує наступні дії:

- упорядкувати записи за абеткою;
- вивести на екран монітора прізвища і номери груп усіх студентів, що мають хоч би одну оцінку 2; якщо таких студентів немає, вивести відповідне повідомлення.

16. Створити список з декількох елементів типу структури з ім'ям NOTE, що містить наступні поля (елементи списку визначаються користувачем в процесі введення, їх кількість заздалегідь не відома):

- прізвище, ім'я;
- номер телефону;
- дата народження (масив з трьох чисел).

Написати програму, що виконує наступні дії:

- упорядкувати записи по датах народження;
- вивести на екран монітора інформацію про людину, номер телефону якого введений з клавіатури; якщо такого немає, вивести відповідне повідомлення.

17. Створити список з декількох елементів типу структури з ім'ям AVTO, що містить наступні поля (елементи списку визначаються користувачем в процесі введення, їх кількість заздалегідь не відома):

- номер автомобіля;
- марка автомобіля;
- прізвище власника.

Написати програму, що виконує наступні дії:

- упорядкувати записи в зростаючому порядку по номерах автомобілів;
- вивести на екран монітора інформацію про автомобілі, марка яких введена з клавіатури; якщо таких немає, вивести відповідне повідомлення.

18. Створити список з декількох елементів типу структури з ім'ям TOY, що містить наступні поля (елементи списку визначаються користувачем в процесі введення, їх кількість заздалегідь не відома):

- назва іграшки;
- її вартість;

- для якого віку призначена.

Написати програму, що виконує наступні дії:

- упорядкувати записи в убуючому порядку за вартістю товару;
- вивести на екран монітора інформацію про іграшки, ціни яких не перевищують тієї, яка введена з клавіатури; якщо таких іграшок немає, вивести відповідне повідомлення.

19. Створити список з декількох елементів типу структури з ім'ям `WORKER`, що містить наступні поля (елементи списку визначаються користувачем в процесі введення, їх кількість заздалегідь не відома):

- прізвище і ініціали працівника;
- назва займаної посади;
- рік вступу на роботу.

Написати програму, що виконує наступні дії:

- упорядкувати записи в алфавітному порядку по прізвищах;
- вивести на екран монітора прізвища працівників, чий стаж роботи в організації перевищує значення, введене з клавіатури; якщо таких працівників немає, вивести відповідне повідомлення.

20. Створити список з декількох елементів типу структури з ім'ям `PRICE`, що містить наступні поля (елементи списку визначаються користувачем в процесі введення, їх кількість заздалегідь не відома):

- назва товару;
- назва магазину, в якому продається товар;
- вартість товару в гривнях.

Написати програму, що виконує наступні дії:

- упорядкувати записи в алфавітному порядку по назві товарів;
- вивести на екран монітора інформацію про товар, назва якого введена з клавіатури; якщо таких товарів немає, вивести відповідне повідомлення.

Лабораторна робота № 8 Створення і використання делегатів

Створити функцію, яка обчислює наближене значення інтеграла $I = \int_a^b f(x)dx$ із заданою точністю ε . У список формальних параметрів функції включити межі інтегрування a і b , точність обчислення ε і посилання на функцію $f(x)$ у вигляді делегата. Обчислити інтеграл, що відповідає своєму варіанту завдання, по одній їх нижче приведених формул.

Деякі теоретичні відомості.

Для обчислення інтеграла відрізок інтегрування $[a, b]$ розбивають на n рівних частин точками $x_i = x_0 + i \cdot h$, де $i = 0 \div n$, $x_0 = a$, $x_n = b$, $h = (b - a)/n$. Позначимо $y_i = f(x_i)$, $i = 0 \div n$. Тоді наближене значення інтеграла можна обчислити по одній з наступних формул:

а) формула лівих прямокутників

$$I \approx h (y_0 + y_1 + \dots + y_{n-1});$$

б) формула правих прямокутників

$$I \approx h (y_1 + y_2 + \dots + y_n);$$

в) формула середніх прямокутників

$$I \approx h (y_{1/2} + y_{1+1/2} + \dots + y_{(n-1)+1/2}), \text{ де } y_{i+1/2} = f(x_i + h/2);$$

г) формула трапеції

$$I \approx h [(y_0 + y_n)/2 + (y_1 + y_2 + \dots + y_{n-1})]$$

д) формула парабол (Сімпсона)

$$I \approx \frac{h}{3} [y_0 + y_n + 2(y_2 + y_4 + \dots + y_{n-2}) + 4(y_1 + y_3 + \dots + y_{n-1})],$$

де n – парне;

е) формула “трьох восьмих”

$$I \approx \frac{3h}{8} [(y_0 + y_n) + 2(y_3 + y_6 + \dots + y_{n-3}) + 3(y_1 + y_2 + y_4 + y_5 + \dots + y_{n-2} + y_{n-1})],$$

де n – кратне 3.

При обчисленні інтеграла з деякою точністю ε розрахунок по одній з наближених формул повторюють, увесь час збільшуючи в два рази число розбиття відрізка інтегрування, до тих пір, поки не буде виконано умову досягнення заданої точності ε :

$$|I(k) - I(2k)| < \varepsilon, \text{ де } k = n, 2n, \dots$$

Варіанти завдань:

- 1) $\int_0^{1.2} \frac{dx}{\sqrt[3]{x+1}}$; 2) $\int_0^3 \sqrt{4+x^2} dx$; 3) $\int_0^1 \sin x \times \exp(-x^2) dx$;
- 4) $\int_1^2 \frac{\sin(x^2+1)}{\sqrt{x^2+x+1}} dx$; 5) $\int_0^{10} (x-5)^2(10-x) dx$; 6) $\int_1^7 \frac{e^x}{x} dx$;
- 7) $\int_0^{\pi/6} \sqrt{\operatorname{tg} x} dx$; 8) $\int_0^{\pi} \sqrt{1+\cos^2 x} dx$; 9) $\int_0^1 \sin(x^4+2x^3+x^2) dx$
- 10) $\int_0^6 \frac{dx}{\sqrt{1+x^4}}$; 11) $\int_0^1 e^{2x} \sin(2x^2+1) dx$; 12) $\int_0^2 \frac{e^{2x} \sin x}{x^2+1} dx$;
- 13) $\int_0^2 e^{-x} \cos \frac{\pi x}{4} dx$; 14) $\int_{0.2}^{0.56} \arccos(\exp(-\sqrt{3x})) dx$; 15) $\int_0^1 e^{\sin x} dx$;
- 16) $\int_0^{\pi/4} \frac{dx}{\sqrt{1-0.25 \sin^2 x}}$; 17) $\int_0^5 e^x \sin x^2 dx$; 18) $\int_0^{\pi} \sin(2 \cos x) dx$;
- 19) $\int_0^{\pi/4} \ln(1+\cos x) dx$; 20) $\int_0^1 \frac{dt}{\sqrt{(1+t^2)(4+t^2)}}$; 21) $\int_{\pi/2}^{\pi} \frac{\cos x}{x} dx$.

Лабораторна робота № 9

Створення класів

Вказівки до виконання роботи.

Кожен клас, що розробляється, повинен, як правило, містити наступні елементи: приховані поля, конструктори з параметрами і без параметрів, методи; перевантажені операції. Функціональні елементи класу повинні забезпечувати несуперечливий, повний зручний інтерфейс класу. При виникненні помилок повинні викидатися виключення.

У програмі повинна виконуватися перевірка усіх розроблених елементів класу.

1. Створити клас «круг». Передбачити наявність методів знаходження довжини кола і площі круга. Використати цей клас в програмі для визначення площі і довжини екземпляра класу «круг».
2. Створити клас «двооперандна арифметична операція», в якому передбачається використання двох арифметичних операцій над цілими числами: суми і добутку. Використати клас для обчислення суми і добутку двох цілих чисел.
3. Створити клас «книга», яка характеризується прізвищем автора, назвою, роком видання, ціною від видавництва. У магазині відпускна ціна книги більше, ніж ціна від видавництва, на 20%. У класі передбачити наявність методу обчислення відпускної ціни. Використати цей клас в програмі для виведення повної інформації про декілька екземплярів книг.
4. Створити клас «учитель», зарплата якого залежить від педагогічного стажу. У цьому класі передбачити метод підрахунку зарплати за правилом: якщо стаж більше 3 років, то зарплата збільшується на 10%, якщо стаж більше 10 років, то на 20%, якщо стаж більше 20 років, то на 30%. Використати цей клас в програмі для підрахунку зарплати конкретного учителя.
5. Створити клас «футбольна команда». Упродовж футбольного сезону команда бере участь в деякій кількості матчів. Деякі вона

виграла, деякі програна, а деякі зіграла внічию. За виграш команді нараховується 3 очки, за нічию - 1 очко, за програш - 0 очок. У класі передбачити наявність підрахунку кількості очок, набраних командою за сезон. Використати цей клас для підрахунку кількості очок, набраних певною командою упродовж футбольного сезону.

6. Створити клас «товар» для представлення товарів на складі, які характеризуються своєю назвою, кількістю і ціною однієї одиниці. Передбачити в класі метод, який підраховував би, на яку суму грошей на складі є товар заданого виду. Використати цей клас для виводу повної інформації про наявність товару заданого виду на складі.
7. Побудувати систему класів для опису плоских геометричних фігур: круга, квадрата, прямокутника. Передбачити методи для створення об'єктів, переміщення на площині, зміни розмірів і обертання на заданий кут.
Написати програму, що демонструє роботу з цими класами. Програма повинна містити меню, що дозволяє здійснити перевірку усіх методів класу.
8. Скласти опис класу для представлення комплексних чисел. Забезпечити виконання операцій складання, віднімання і множення комплексних чисел.
Написати програму, що демонструє роботу з цими класами. Програма повинна містити меню, що дозволяє здійснити перевірку усіх методів класу.
9. Скласти опис класу для об'єктів-векторів, що задаються координатами кінців в тривимірному просторі. Забезпечити операції складання і віднімання векторів з отриманням нового вектору (суми або різниці), обчислення скалярного добутку двох векторів, довжини вектору, косинуса кута між векторами.
Написати програму, що демонструє роботу з цими класами. Програма повинна містити меню, що дозволяє здійснити перевірку усіх методів класу.

10. Скласти опис класу прямокутників із сторонами, паралельними осям координат. Передбачити можливість переміщення прямокутника на площини, зміну розмірів, побудову найменшого прямокутника, що містить два задані прямокутники, і прямокутники, що є загальною частиною (перетином) двох прямокутників.
Написати програму, що демонструє роботу з цими класами. Програма повинна містити меню, що дозволяє здійснити перевірку усіх методів класу.
11. Створити клас «студентська група». Передбачити можливість роботи зі змінним числом студентів, пошуку студента за якою-небудь ознакою (наприклад, по прізвищу, даті народження або номеру телефону), додавання і видалення записів, сортування по різних полях. Написати програму, що демонструє роботу з цими класами. Програма повинна містити меню, що дозволяє здійснити перевірку усіх методів класу.
12. Створити клас для роботи з одновимірним масивом цілих чисел (вектором). Забезпечити наступні можливості:
- завдання довільних цілих меж індексів при створенні об'єкту;
 - звернення до окремого елемента масиву з контролем виходу за межі масиву;
 - виконання операцій поелементного складання і віднімання масивів з однаковими межами індексів;
 - виконання операцій множення і ділення усіх елементів масиву на скаляр;
 - виведення на екран елемента масиву по заданому індексу і усього масиву.
- Написати програму, що демонструє усі розроблені елементи класу.
13. Створити клас многочленів від однієї змінної, що задаються ступенем многочлена і масивом коефіцієнтів. Забезпечити наступні можливості:
- обчислення значення многочлена для заданого аргументу;
 - операції складання, віднімання і множення многочлена з отриманням нового об'єкту-многочлена;
 - отримання коефіцієнта, заданого по індексу;

- виведення на екран опису многочлена.

Написати програму, що демонструє усі розроблені елементи класу.

14. Створити клас, що забезпечує представлення матриці довільного розміру з можливістю зміни числа рядків і стовпців, виведення на екран підматриці будь-якого розміру і усієї матриці, доступу по індексах до елемента матриці.

Написати програму, що демонструє усі розроблені елементи класу.

15. Створити клас, що реалізовує тип даних «дійсна матриця» і роботу з ним. Клас повинен реалізовувати наступні операції над матрицями:

- складання, віднімання (як з іншою матрицею, так і з числом);
- комбіновані операції присвоєння(+=, -=);
- операції порівняння на рівність і нерівність;
- методи обчислення зворотної і транспонованої матриці;
- доступ до елемента по індексах.

Написати програму, що демонструє усі розроблені елементи класу.

16. Створити клас «колода карт», що включає закритий масив елементів класу «карта». У карті зберігаються масть і номер. Забезпечити можливість виведення карти по номеру, виведення усіх карт, перемішування колоди і видача усіх карт з колоди поодиночці і по 6 штук у випадковому порядку.

Написати програму, що демонструє усі розроблені елементи класу.

17. Створити клас «поїзд», що містить наступні закриті поля:

- назва пункту призначення;
- номер поїзду (може містити букви і цифри);
- час відправлення.

Передбачити властивості для отримання стану об'єкту.

Описати клас «вокзал», що містить закритий масив поїздів. Забезпечити наступні можливості:

- висновок інформації про поїзд по номеру за допомогою індексу;

- висновок інформації про поїзди, що вирушають після введеного з клавіатури часу;
- перевантажену операцію порівняння, що виконує порівняння часу відправлення двох поїздів;
- висновок інформації про поїзди, що вирушають в заданий пункт призначення.

Інформація має бути відсортована за часом відправлення. Написати програму, що демонструє усі розроблені елементи класу.

18. Створити клас «товар», що містить наступні закриті поля :

- назва товару;
- назва магазину, в якому продається товар;
- вартість товару в гривнях.

Передбачити властивості для отримання стану об'єкту.

Описати клас «склад», що містить закритий масив товарів.

Забезпечити наступні можливості:

- висновок інформації про товар по номеру за допомогою індексу;
- виведення на екран інформації про товар, назва якого введена з клавіатури; якщо таких товарів немає, видати відповідне повідомлення;
- сортування товарів по назві магазину, по найменуванню і за ціною;
- переобтяжену операцію складання товарів, що виконує складання їх цін.

Написати програму, що демонструє усі розроблені елементи класу.

19. Створити клас «запис», що містить наступні закриті поля:

- прізвище, ім'я;
- номер телефону;
- дата народження (масив з трьох чисел).

Передбачити властивості для отримання стану об'єкту.

Описати клас «записник», що містить закритий масив записів.

Забезпечити наступні можливості:

- виведення на екран інформації про людину, номер телефону якої введений з клавіатури; якщо такої немає, видати на екран відповідне повідомлення;

- пошук людей, день народження яких сьогодні або в заданий день;
- пошук людей, день народження яких буде наступного тижня;
- пошук людей, номер телефону яких починається на 3 заданих цифри.

Написати програму, що демонструє усі розроблені елементи класу.

20. Створити абстрактний клас `Vehicle` (транспортний засіб). На його основі реалізувати класи `Plane` (літак), `Car` (автомобіль) і `Ship` (корабель). Класи повинні мати можливість задавати і отримувати координати і параметри засобів пересування (ціна, швидкість, рік випуску і т. п.) за допомогою властивостей. Для літака має бути визначена висота, для літака і корабля - кількість пасажирів, для корабля - порт приписки. Динамічні характеристики задати за допомогою методів.

21. Створити базовий клас «рядок». Обов'язкові поля класу:

- поле для зберігання символів рядка;
- значення типу `word` для зберігання довжини рядка у байтах.

Реалізувати обов'язкові методи наступного призначення:

- конструктор без параметрів;
- конструктор, що приймає як параметр строковий літерал;
- конструктор, що приймає як параметр символ;
- метод отримання довжини рядка;
- метод очищення рядка (зробити рядок порожнім).

Описати похідний від класу «рядок» клас «комплексне число».

Рядки цього класу складаються з двох полів, розділених символом `i`. Перше число задає значення дійсної частини числа, друге - значення уявної. Кожне поле може містити тільки символи десяткових цифр і символи '-' та '+', задаючи знак числа. Символи '-' або '+' можуть знаходитися тільки в першій позиції числа, причому символ '+' може бути відсутнім, в цьому випадку число вважається позитивним. Якщо у складі рядка, який буде ініціалізувати, будуть зустрінуті будь-які символи, відмінні від допустимих, клас «комплексне число» повинен приймати нульове значення. Приклади рядків:

`33i12`

`-7i100`

$$+5i-21$$

Для класу «комплексне число» визначити наступні методи:

- перевірка на рівність;
- складання чисел;
- множення чисел.

Список використаної літератури

1. Павловская Т. А. С#. Программирование на языке высокого уровня. Учебник для вузов. – СПб.: Питер, 2009.
2. Шилдт Герберт. Полный справочник по С#.: Пер с англ. – М. : Вильямс, 2004.
3. Косирева Л. А., Рачинська А. Л. Збірник завдань для лабораторних робіт по купсах «Програмування» та «Практикум на ЕОМ» для студентів 1 курсу спеціальності «Механіка», частина 1- Одеса, ОНУ, 2011, 32с.
4. Павловская Т. А., Щупак Ю. А. «С/С++. Программирование на языке высокого уровня. Структурное программирование. Практикум». Питер, 2002.

Наукове видання

Косирева Ліаліна Анатоліївна

**ЗАВДАННЯ ТА МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ
лабораторних робіт з навчальної дисципліни
«Комп'ютерна математика»**

для студентів I курсу напрямку підготовки 6.040202 «Механіка»

Комп'ютерне верстання – О. І. Карлічук

Підп. до друку 24.01.2014. Формат 60x84/16.

Умов.-др.арк. 2,44. Тираж 25.

Зам. № 799

Видавець и виготовлювач

Одеський національний університет

імені І. І. Мечникова

Свідоцтво суб'єкта видавничої справи

ДК № 4215 от 22.11.2011 р.

Україна, 65082, м. Одеса, вул.. Єлісаветинська, 12

Тел. (048) 723-28-39, E-mail: druk@onu.edu.ua