

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КЕРУЮЧИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

Кафедра спеціалізованих комп'ютерних систем

МЕТОДИЧНІ ВКАЗІВКИ

**до лабораторних та самостійних робіт
з дисципліни**

«ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

Частина 1

Харків – 2019

Методичні вказівки розглянуто і рекомендовано до друку

на засіданні кафедри спеціалізованих комп'ютерних систем
25 лютого 2019 р., протокол № 9.

Тематика занять охоплює основну частину 1 семестру курсу, роботу з інтегрованим середовищем програмування Eclipse. Пропонується при розв'язанні задач програмувати їх реалізацію засобами популярної мови Java.

Призначено для студентів факультету ІКСТ зі спеціальності 123 «Комп'ютерна інженерія» першого освітнього рівня освіти (бакалавр) усіх форм навчання.

Укладачі:

доценти Є. П. Павленко,
В. М. Бутенко,
старш. викл. О. В. Головка

Рецензент

проф. С. І. Доценко

МЕТОДИЧНІ ВКАЗІВКИ
до лабораторних та самостійних робіт
з дисципліни
«ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

Частина 1

Відповідальний за випуск Павленко Є. П.

Редактор Еткало О. О.

Підписано до друку 27.03.19 р.

Формат паперу 60x84 1/16. Папір писальний.

Умовн.-друк. арк. 2,5. Тираж 50. Замовлення №

Видавець та виготовлювач Український державний університет
залізничного транспорту,
61050, Харків-50, майдан Фейербаха, 7.
Свідоцтво суб'єкта видавничої справи ДК № 6100 від 21.03.2018 р.

ЗМІСТ

Вступ.....	4
ЛАБОРАТОРНА РОБОТА 1. Робота в інтегрованому середовищі програмування Eclipse для Java Developers	5
ЛАБОРАТОРНА РОБОТА 2. Вивчення елементарних конструкцій мови Java	15
ЛАБОРАТОРНА РОБОТА 3. Вивчення циклічних операторів мови Java	28
ЛАБОРАТОРНА РОБОТА 4. Робота з одновимірними масивами	35
ЛАБОРАТОРНА РОБОТА 5. Робота з двовимірними масивами.....	41
ЛАБОРАТОРНА РОБОТА 6. Робота з рядками	44
Завдання для самостійної роботи студентів	54
Запитання для самостійної підготовки до модульного контролю.....	55
Список літератури.....	61

ВСТУП

Методичні вказівки призначені допомогти студенту розібратися в прийомах самостійного програмування елементів загальнотехнічного спрямування на прикладі класичних та спеціалізованих систем залізничного транспорту [1].

Мова Java – об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. З 2009 року мовою займається компанія «Oracle», яка того року придбала «Sun Microsystems». В офіційній реалізації Java-програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

«Oracle» надає компілятор Java та віртуальну машину Java, які задовольняють специфікації Java Community Process, під ліцензією GNU General Public License [2].

Синтаксис мови схожий із синтаксисом C та C++. Зокрема взято за основу об'єктну модель C++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста, та полегшено процес розробки об'єктно-орієнтованих програм. Ряд дій, які в C/C++ повинні здійснювати програмісти, доручено віртуальній машині. Передусім Java розроблялась як платформо незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням, що порівняно, наприклад, з C++ зменшує швидкість роботи програм. За необхідності таких дій Java дає змогу викликати підпрограми, написані іншими мовами програмування [2].

У лабораторіях кафедри СКС встановлено Eclipse (Kepler 2013) версії 4.3, що дає змогу отримати необхідні навички програмування кросплатформенною, об'єктно-орієнтованою мовою високого рівня Java.

ЛАБОРАТОРНА РОБОТА 1

Робота в інтегрованому середовищі програмування Eclipse для Java Developers

Мета роботи: отримання навичок роботи в інтегрованому середовищі програмування Eclipse (Kepler 2013) версії 4.3, ознайомлення з командами основного меню, основними командами редактора.

1 Завдання та порядок виконання

1.1 Вивчити теоретичний матеріал.

1.2 Підготувати відповіді на контрольні запитання.

1.3 Запустити Eclipse. У вікні редактора набрати приклад навчальної програми. Випробувати команди редактора Eclipse. Запустити програму. Додати аргументи командного рядка. Додати відповідні команди в програму. Запустити програму. Проаналізувати отриманий результат.

Контрольні запитання

1 Чому середовище для розробки Eclipse-програм називається інтегрованим?

2 З яких частин складається екран основного меню? Їх призначення.

3 Призначення опцій основного меню.

4 Як додати аргументи командного рядка для запуску програми ?

5 Вказати команди для запуску програми.

2 Зміст звіту

2.1 Номер роботи, її назва, визначення мети.

2.2 Стислий зміст теоретичного матеріалу та відповіді на контрольні запитання.

2.3 Результати виконання завдання: схеми алгоритмів, програми, результати роботи програм.

2.4 Висновки до роботи.

3 Навчальний матеріал

3.1 Запуск Eclipse. Для запуску Eclipse слід знайти на робочому столі ярлик з підписом Eclipse і подвійно натиснути на стрілку. При цьому на екрані з'явиться впливаюче вікно (рисунок 1).

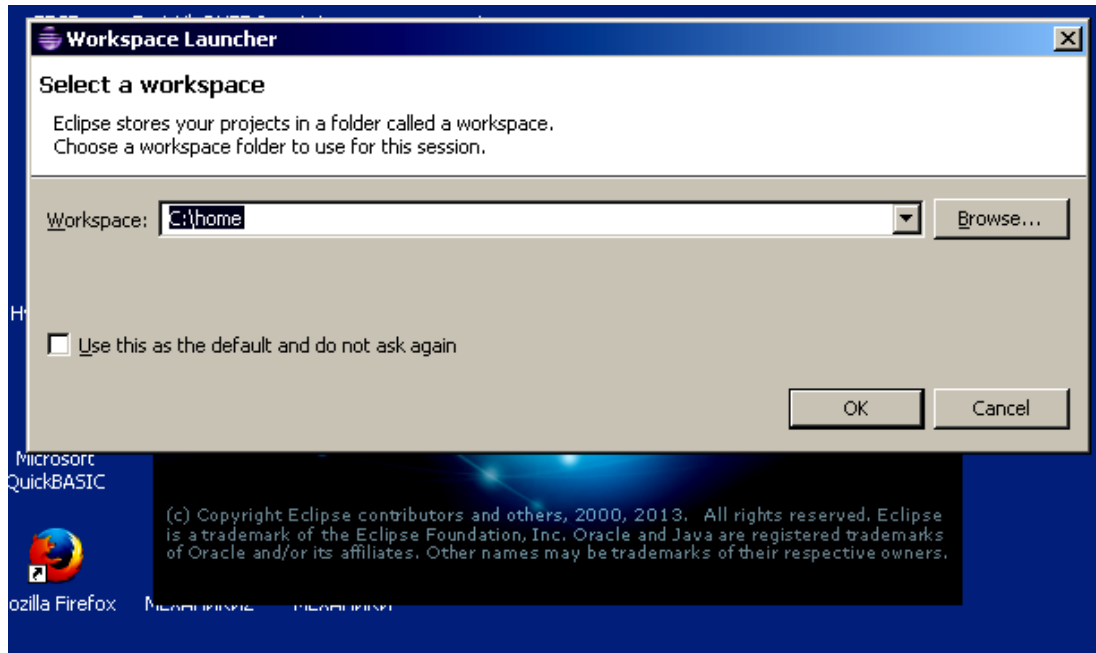


Рисунок 1

На ньому вказується папка, у якій Eclipse зберігає ваші проекти. Якщо є необхідність, виберіть іншу папку робочої області, яку потрібно використовувати для цього. Перед вами з'явиться робоче вікно Eclipse, яке складається з п'яти частин:

- основного меню (верхня частина екрана, кнопки і значки);
- Package Explorer (буквально – дослідження пакетів), списку відкритих пакетів – проектів Java – розміщується зліва;
- вікна редактора із закладками відкритих файлів Java – розміщується справа;
- вікна із закладками повідомлень (рисунок 2);
- рядка підказки чи вказівки дій функціональних кнопок.

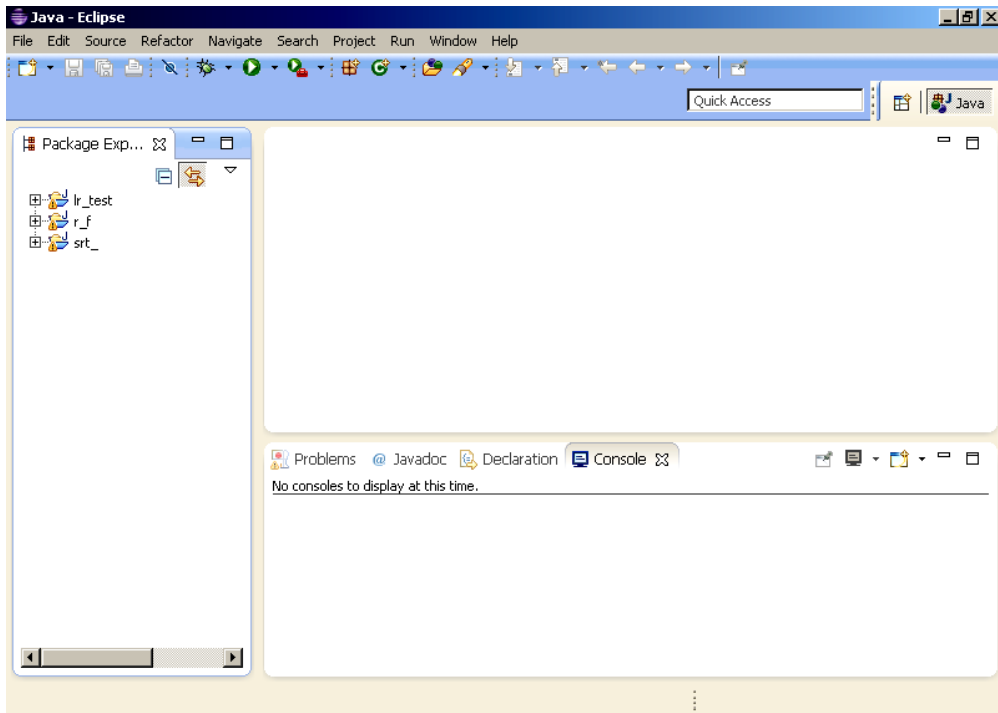


Рисунок 2

Створення, запуск і налагодження консольного проекту
 Для створення консольного проекту необхідно вибрати меню File -New - Java Project (рисунок 3).

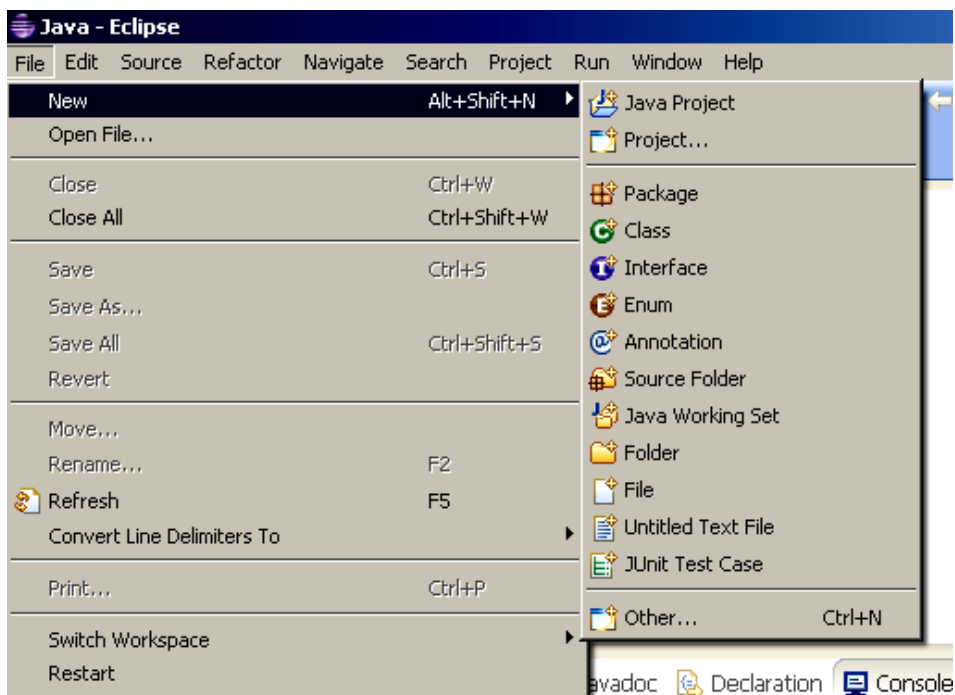


Рисунок 3

У результаті вибору відкриється вікно створення проекту (рисунок 4).

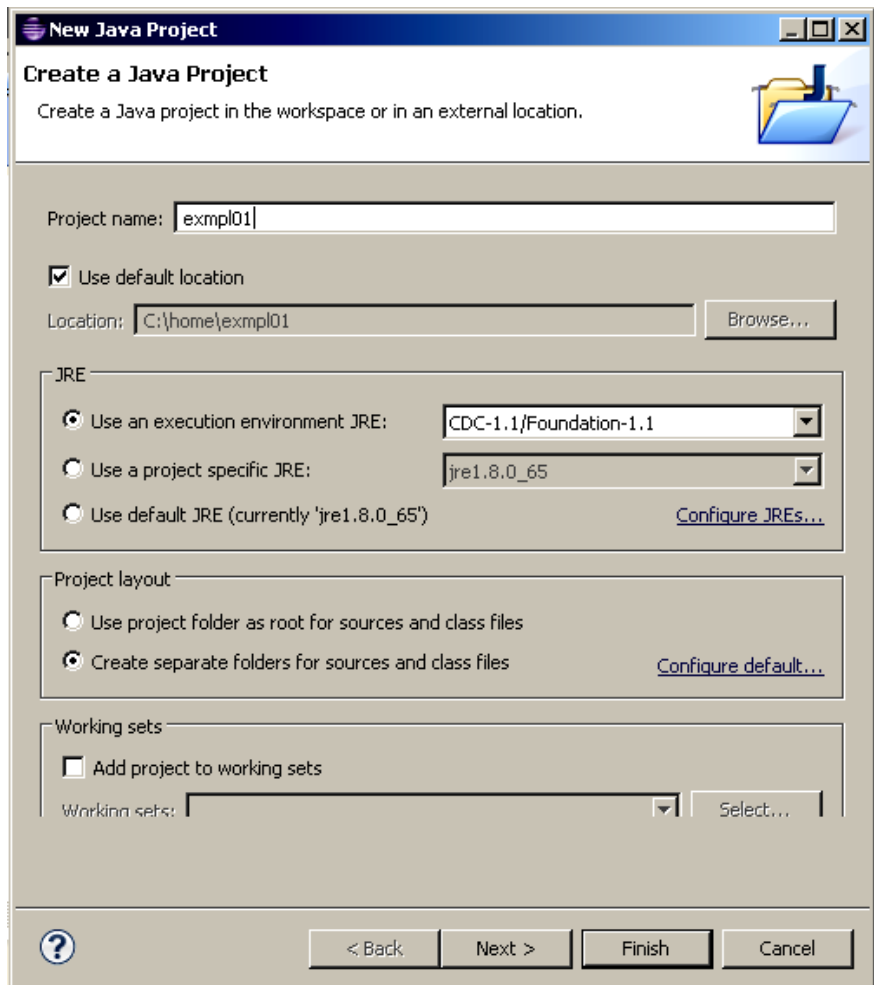


Рисунок 4

Далі вводиться ім'я проекту, за необхідності вибирається JRE. Можливо задати деякі додаткові параметри, натискаючи кнопку Next, можна залишити все інше за замовчуванням і натиснути кнопку Finish для створення проекту. У Package explorer зліва з'явиться щойно створений проект (рисунок 5).

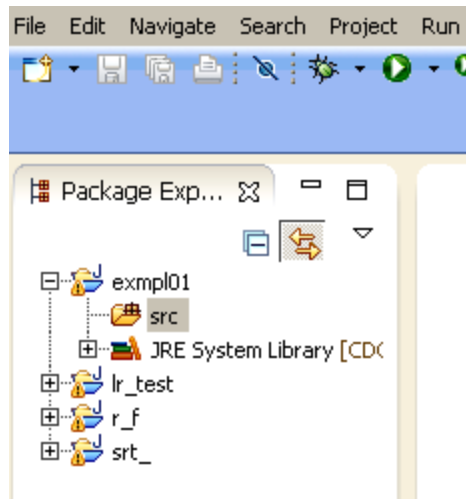


Рисунок 5

Щоб створити пакет у новому проекті, необхідно натиснути правою кнопкою миші на назву проекту в Package Explorer або вибрати в головному меню File - New - Package. Потім ввести ім'я пакета і натиснути кнопку Finish. після цього в папці src з'явиться піктограма порожнього пакета з ім'ям. Якщо це не буде зроблено, буде створено пакет з ім'ям Default Package (Стандартний пакет). За допомогою File - New можна створити клас (Class), інтерфейс (Interface), перерахування (Enum) або інший елемент проекту (рисунок 6).

При створенні класу програми необхідно як мінімум ввести його ім'я. Так само в Майстрі створення класу можна додати до класу інтерфейси, вказати модифікатор доступу, повідомити, що клас буде містити метод main() та ін. (рисунок 6).

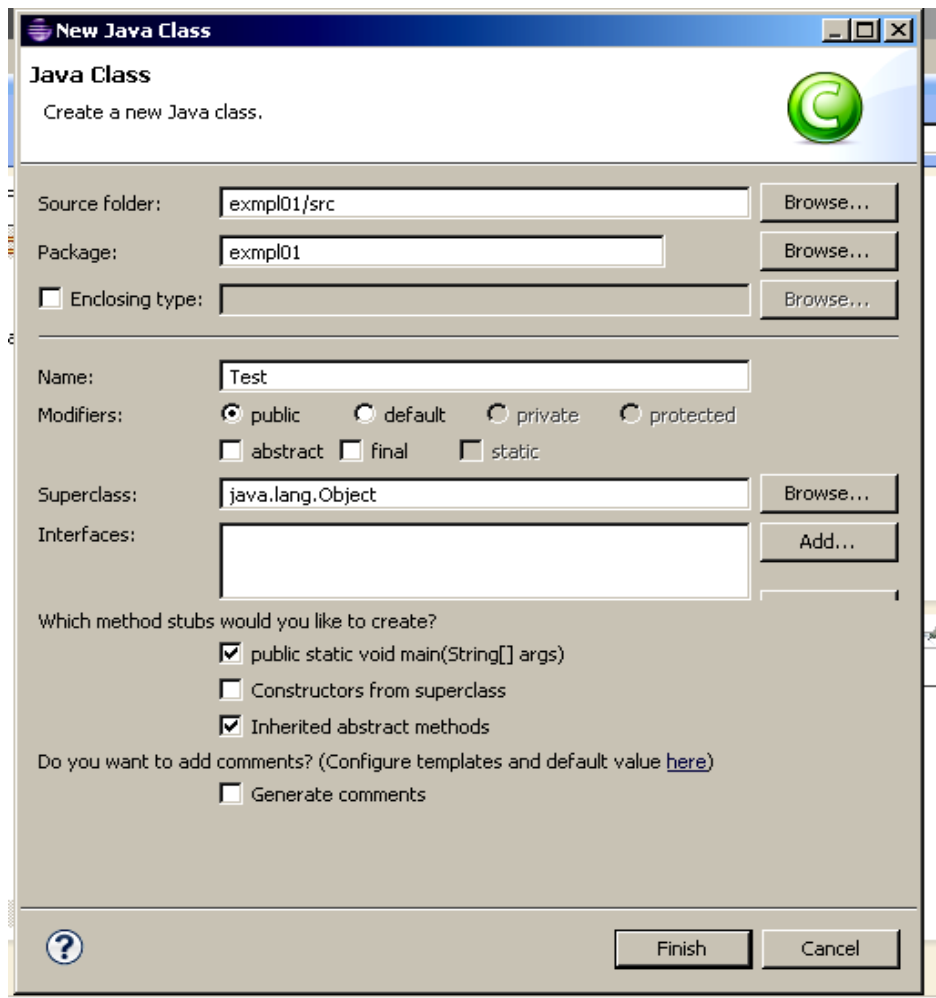


Рисунок 6

Створений клас буде мати вигляд як на рисунку 7. На початку програми додамо рядок
`import java.util.Scanner;`
Це дасть можливість організувати введення з клавіатури та виведення на екран значень змінних (рисунок 7).

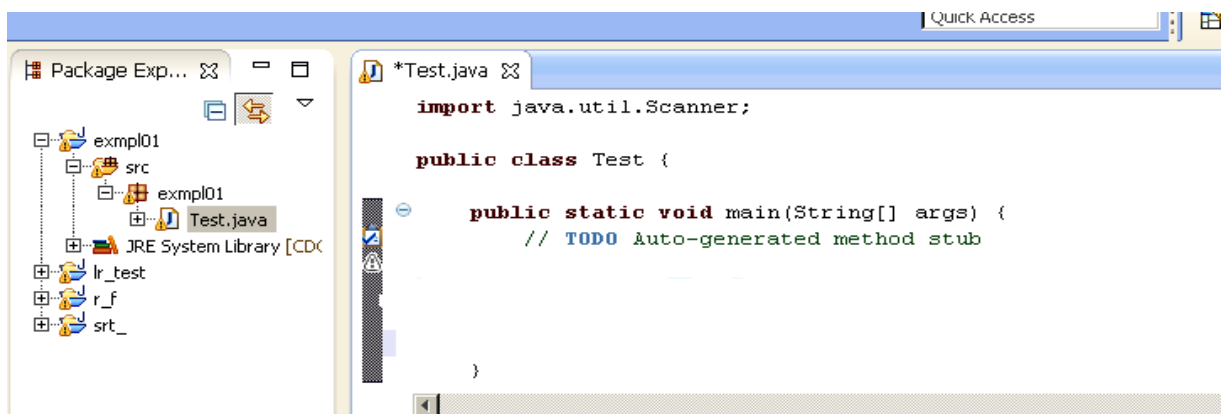


Рисунок 7

Допишемо декілька операторів (рисунок 8) , а саме:

```
Scanner in = new Scanner(System.in);
```

(оголошує вхідний потік – можливість введення з клавіатури)

```
int j = in.nextInt();
```

(оголошує змінну j та дає можливість введення її значення)

```
System.out.println("helo" +j);
```

(виводить напис на екран)

Запустимо програму.

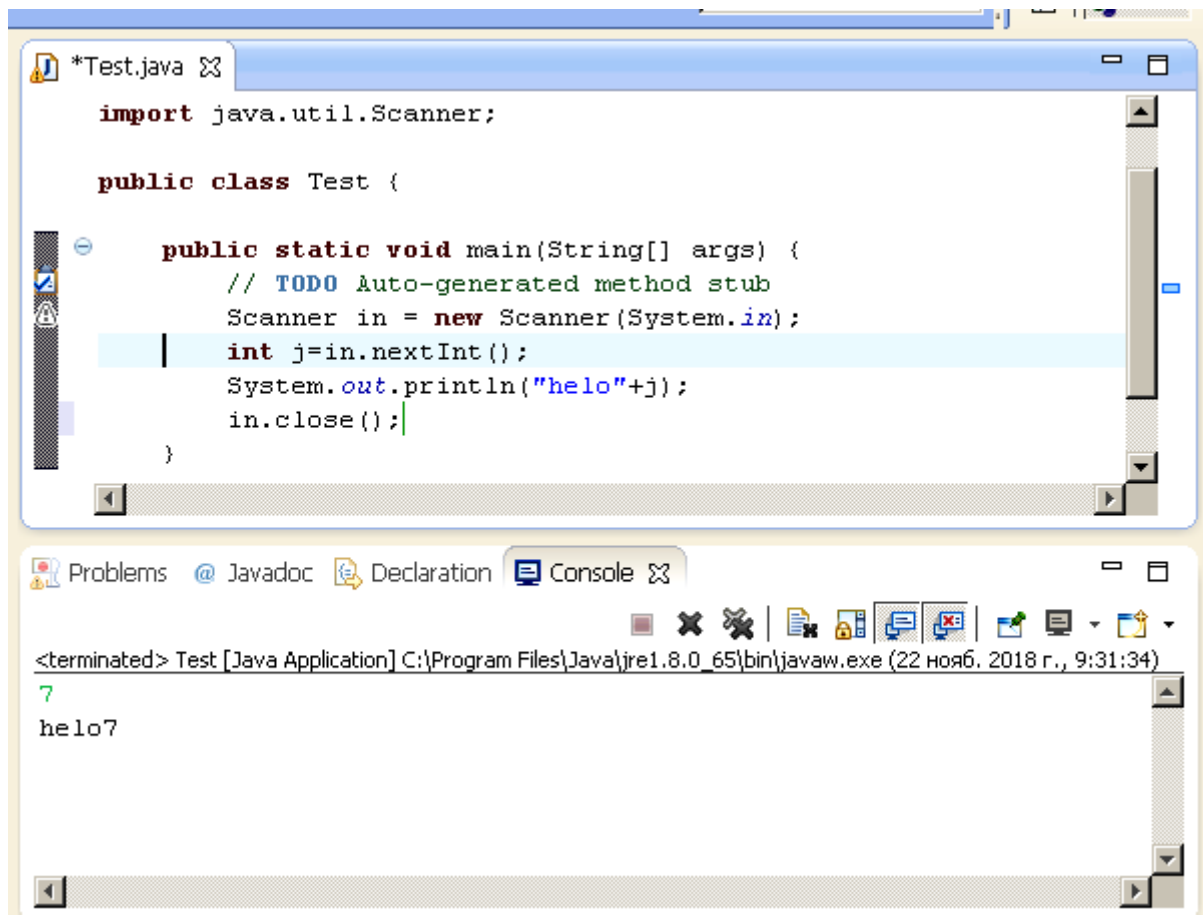


Рисунок 8

Внизу рисунка 8 показано консоль. Зеленим кольором відображено 7 – значення, введене з клавіатури в змінну j, та текст, що виведе програма: helo7.

Проект можна запустити натисненням комбінації кнопок Ctrl + F11 або кнопкою Run на панелі інструментів (зелене коло з білим трикутником усередині). У папці src проекту може розміщуватися тільки java-код і файли з розширенням properties.

Усі інші ресурси проекту завжди повинні міститися в проекті, але поза папкою src.

При розробці може знадобитися тимчасово або постійно закоментувати ділянку коду. Виконується вказана дія після виділення фрагмента коду натисканням сполучення кнопок Ctrl-/. Дія зі зняття коментарю з виділеного фрагмента здійснюється цим же поєднанням кнопок.

Код java має для кращого візуального сприйняття східчасту структуру, вигляд якої можна змінити в налагодженнях редактора. Але в процесі розробки вона може бути порушена програмістом, тоді для додання коду форматovanого виду слід натиснути кнопки Ctrl-Shift-F або вибрати Source - Format.

Для передачі аргументів на метод main() необхідно вибрати пункт меню Run-Run Configurations, відкриється вікно (рисунок 9). Далі на вкладці Arguments в полі Program arguments вказати параметри. Параметри розділяються пропуском або кнопкою Enter (рисунок 10).

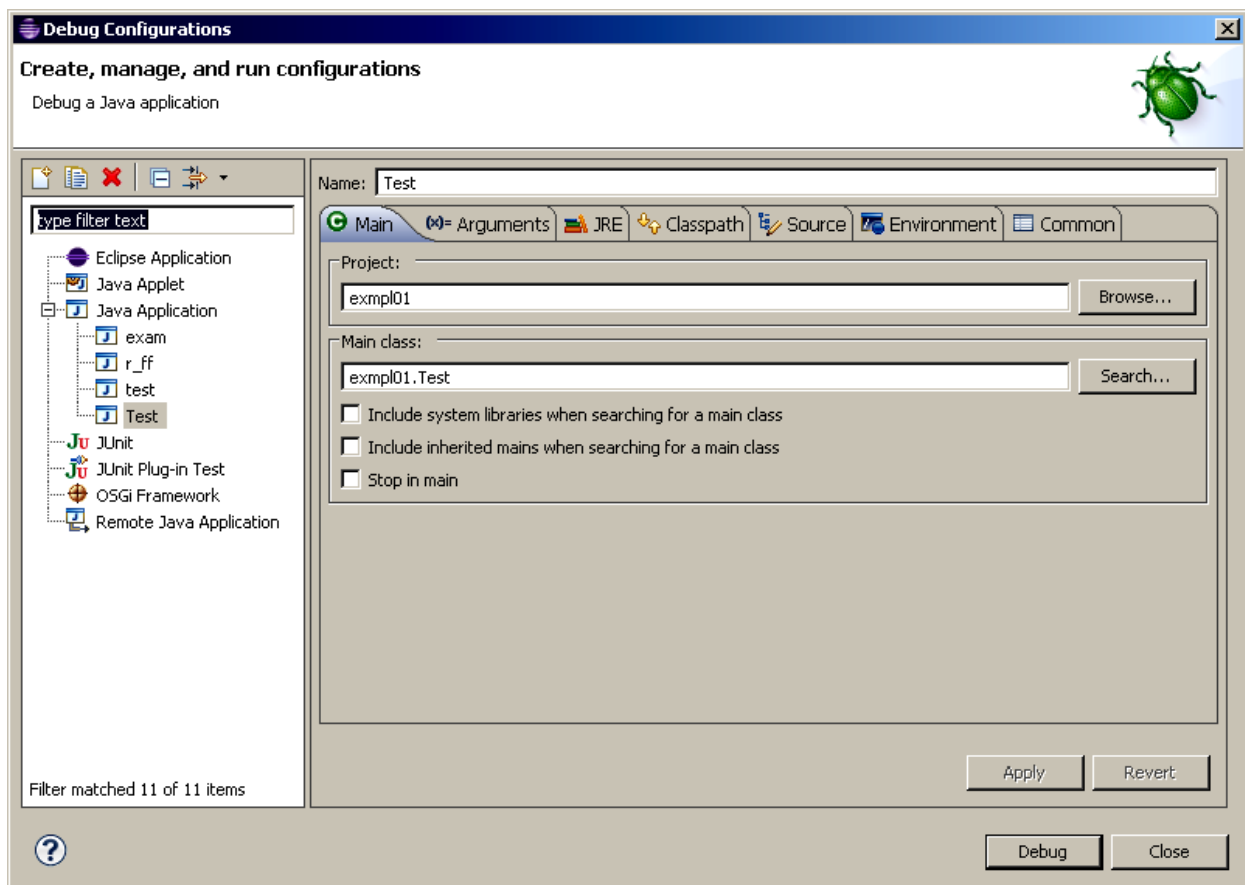


Рисунок 9

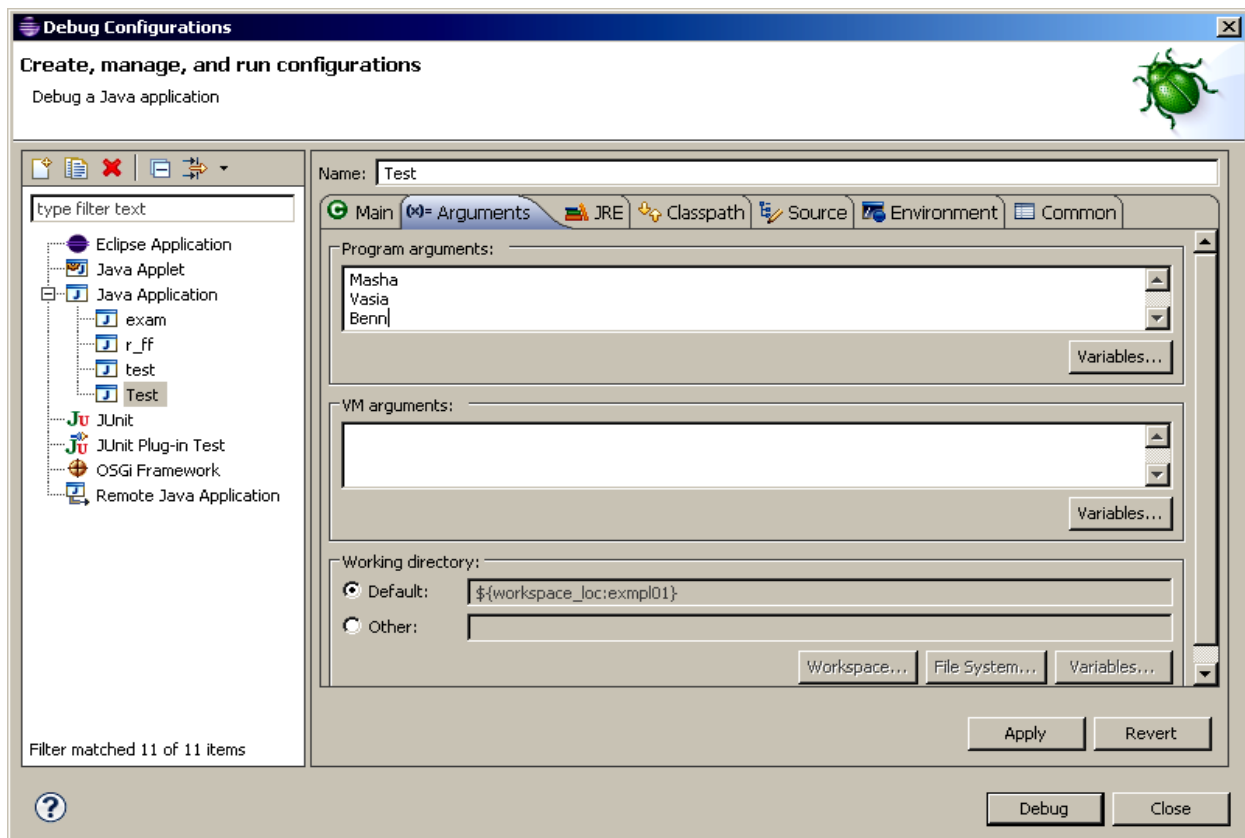


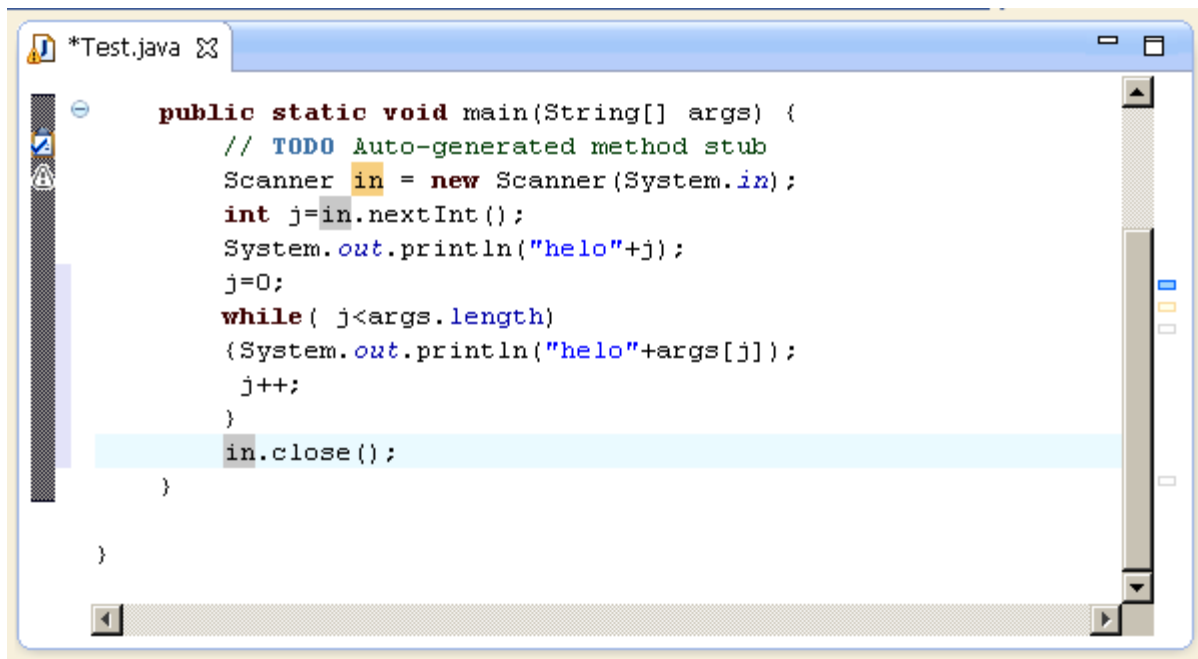
Рисунок 10

Для налагодження проект запускається кнопкою F11 клавіатури або кнопкою Debug панелі інструментів (кнопка із зображенням жука). Точки зупинки (Breakpoint) ставляться подвійним натисканням миші на сірому полі зліва від коду класу. Точка зупинки графічно виглядає як маленька блакитна крапка.

Для прикладу роботи з аргументами програми додамо до програми Test декілька рядків:

```
j=0;
while(j <args.length)
{ System.out.println ( "helo" + args [j]);
j++;
}
```

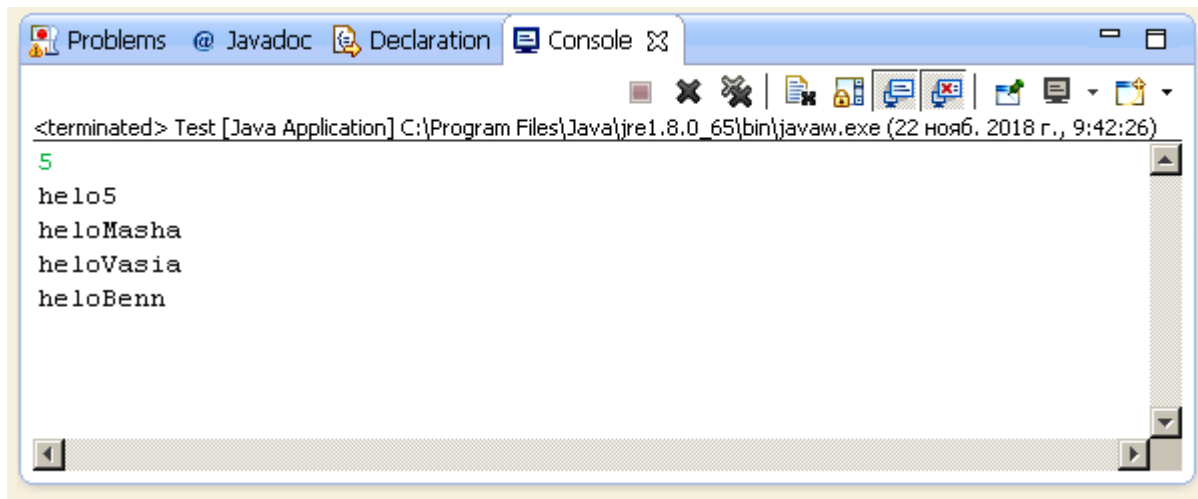
На рисунку 11 зображено лістинг програми. Це цикл while(), у якому друкуються аргументи командного рядка, які записані в масив args [], у кількості args.length .



```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    Scanner in = new Scanner(System.in);
    int j=in.nextInt();
    System.out.println("helo"+j);
    j=0;
    while( j<args.length)
    {System.out.println("helo"+args[j]);
      j++;
    }
    in.close();
}
}
```

Рисунок 11

Введемо аргументи для запуску програми як було вказано раніше. У вкладці «консоль» після запуску отримаємо вікно як на рисунку 12.



```
<terminated> Test [Java Application] C:\Program Files\Java\jre1.8.0_65\bin\javaw.exe (22 нояб. 2018 г., 9:42:26)
5
helo5
heloMasha
heloVasia
heloBenn
```

Рисунок 12

Java – це повністю об'єктно-орієнтована мова, тому в кожній програмі має бути наявне оголошення класу, ім'я якого повинне збігатися з ім'ям файлу програми. Специфікатор `public` перед ім'ям класу означає, що цей клас доступний для інших класів і

об'єктів поза цим пакетом. Без цього специфікатора наша програма не може бути виконана.

Метод `main` повинен бути наявним у кожній автономній програмі. З нього починається виконання програми. Параметр `String args []` передає в програму масив рядків – аргументів з командного рядка. Розмір масиву визначається змінною `args.length`. Ми в циклі формуємо рядки " helo j-й елемент командного рядка" і виводимо їх на екран (точніше в потік `out` - аналог `stdout` в `C / C ++`). Експериментуйте з цією програмою, задаючи різні аргументи командного рядка і спробуйте її модифікувати, щоб вона виводила рядки типу "j-й аргумент командного рядка, go out !!! "

ЛАБОРАТОРНА РОБОТА 2

Вивчення елементарних конструкцій мови Java

Мета роботи: вивчення правил запису елементарних конструкцій мови програмування Java, змінних, констант, операторів, виразів. Створення програми з їх використанням.

1 Завдання та порядок виконання

1.1 Вивчити теоретичний матеріал.

1.2 Підготувати відповіді на контрольні запитання.

1.3 Записати мовою Java математичні вирази.

1.4 Записати мовою Java програму згідно з варіантом завдання.

Контрольні запитання

1 Визначити поняття «ідентифікатор». Правила складання ідентифікаторів.

2 Змінні. Типи, діапазони їх значень, правила оголошення змінних.

3 Константи. Види констант.

4 Основні класи операторів.

5 Визначити поняття «блок».

6 Вказати оператор умовного переходу.

2 Зміст звіту

2.1 Номер роботи, її назва, визначення мети.

2.2 Стислий зміст теоретичного матеріалу та відповіді на контрольні запитання.

2.3 Текст програми, вхідні дані та результати виконання завдання з рівнів 1 та 2.

2.4 Висновки до роботи.

3 Навчальний матеріал

3.1 Ідентифікатори – це імена, що використовуються для звернення до змінних, функцій, міток та інших визначених користувачем об'єктів. Ідентифікатор може містити від одного до декількох символів. Перший символ обов'язково повинен бути літерою латинського алфавіту чи літерою підкреслення. У мові Java великі та малі літери трактуються як різні. Наприклад, ідентифікатори count, Count і COUNT представляють три різні змінні. Ідентифікатор змінної не повинен збігатися з ключовим словом, з ім'ям бібліотечної функції чи функції користувача.

3.2 Змінні. Усі змінні перед використанням повинні бути явно оголошені, тобто вказано тип даних, до якого належить змінна.

Найпростіший тип даних у мові Java – булевий. Булеві змінні мають два можливих значення: true або false. Довжина 1 біт.

Приклад оголошення булевої змінної:

```
boolean ab = true;
```

Цілі змінні мають такі типи:

byte – 8-бітове ціле число зі знаком у діапазоні -128 +127;

short – 16-бітове ціле число зі знаком у діапазоні -32768 +32767;

int – 32-бітове ціле число зі знаком у діапазоні $-2^{31} + 2^{31} - 1$;

long – 64-бітове ціле число зі знаком у діапазоні $-2^{63} + 2^{63} - 1$.

Створення цілих змінних виконується, наприклад, таким оператором:

```
int j = 20;
```


Символьні змінні створюються так само, як і цілочислові і булеві змінні:

```
char C = 'b';
```

Вони можуть оброблятися як 16-розрядні цілі без знака (діапазон 0-65536) або \u 0000- \uFFFF).

Числа з плаваючою точкою подані двома типами: float і double. Обидва типи відповідають стандартній специфікації для чисел з плаваючою точкою, що забезпечує переносимість Java – програм:

float – 32-бітове число з плаваючою точкою звичайної точності;

double – 64-бітове число з плаваючою точкою подвійної точності.

Константи в Java можуть бути будь-якого вищезгаданого типу. Константи спеціального виду: 8-ковий (повинні починатися з 0), 16-ковий (повинні починатися з 0x), Unicode – символи (повинні починатися з \u).

Приклад:

```
043
```

```
0x4A7
```

```
\u 0002
```

3.3 Математичні функції в Java. Для обчислень математичних виразів використовуються математичні функції, реалізовані в публічному класі Math, що успадковує клас Object.

Клас Math містить методи для виконання таких основних числових операцій, як елементарна експоненція, логарифм, квадратний корінь та тригонометричні функції.

Опис методів класу Math наведено в таблиці 1.

Таблиця 1

Модифікатор і тип	Метод та опис
1	2
static double	<u>abs</u> (double a) Повертає абсолютне значення double значення
static double	<u>acos</u> (double a) Повертає дуговий косинус значення; повернутий кут міститься в діапазоні від 0.0 до π
static int	<u>addExact</u> (int x, int y) Повертає суму своїх аргументів, викидаючи виключення, якщо результат перепоовнює int.
static long	<u>addExact</u> (long x, long y) Повертає суму своїх аргументів, викидаючи виключення, якщо результат перепоовнює long
static double	<u>asin</u> (double a) Повертає дуговий синус значення; повернутий кут міститься в діапазоні $-\pi / 2$ до $\pi / 2$
static double	<u>atan2</u> (double y, double x) Повертає кутовий <i>тема</i> від перетворення прямокутних координат (x, y) до полярних координат (r, <i>theta</i>)
static double	<u>atan</u> (double a) Повертає дотичну дугу значення; повернутий кут міститься в діапазоні $-\pi / 2$ до $\pi / 2$
static double	<u>cbrt</u> (double a) Повертає корінь куба double значення
static double	<u>ceil</u> (double a) Повертає найменше (найближче до від'ємної нескінченності) double значення, яке більше або дорівнює аргументу і дорівнює математичному цілому
static double	<u>copySign</u> (double magnitude, double sign) Повертає перший аргумент з плаваючою точкою зі знаком другого аргументу з плаваючою точкою

Продовження таблиці 1

1	2
static double	<u>cos</u> (double a) Повертає тригонометричний косинус під кутом
static double	<u>cosh</u> (double x) Повертає гіперболічний косинус double значення
static double	<u>exp</u> (double a) Повертає число Ейлера e у значенні типу double
static double	<u>random</u> () Повертає double значення з додатним знаком, більшим або рівним 0.0 і меншим 1.0
static double	<u>rint</u> (double a) Повертає double значення, найближче до аргументу значення, і дорівнює математичному цілому
static long	<u>round</u> (double a) Повертає найближче до аргументу значення
static int	<u>round</u> (float a) Повертає найближчий int до аргументу, коли зв'язки округлюються до додатної нескінченності
static float	<u>signum</u> (float f) Повертає функцію signum аргументу; нуль, якщо аргумент дорівнює нулю, 1.0f, якщо аргумент більший від нуля, -1.0f, якщо аргумент менший від нуля
static double	<u>sin</u> (double a) Повертає тригонометричний синус під кутом
static double	<u>sinh</u> (double x) Повертає гіперболічний синус double значення
static double	<u>sqrt</u> (double a) Повертає правильний округлений додатний квадратний корінь із double значенням

Приклади запису математичного виразу в Java:
У математиці:

$$y = ax^2 + 3.45x + c,$$

в Java:

$$y = a * \text{pow}(x, 2) + 3.45 * x + c;$$

У математиці:

$$s = \cos^4(a+b) \sqrt{(a^2 + f^2)} + \ln(a^4),$$

в Java:

$$s = \text{pow}(\cos(a+b), 4) * \text{sqrt}(\text{pow}(a, 2) + \text{pow}(f, 2)) + \log(\text{pow}(a, 4)).$$

3.4 Введення даних з клавіатури і виведення на екран. Найбільш простим способом взаємодії з користувачем є консоль: ми можемо виводити на консоль деяку інформацію або, навпаки, зчитувати з консолі дані. Для взаємодії з консоллю в Java застосовується клас `System`, а його функціональність власне забезпечує консольне введення і виведення. Оператор

```
import java.util.Scanner;
```

розміщується на початку програми. Це дасть можливість організувати введення з клавіатури та виведення на екран значень змінних.

`Scanner` – це клас у пакеті `java.util`, який використовується для введення примітивних типів, таких як `int`, `double` та ін., і рядків. Це найпростіший спосіб читати вхідні дані в програмі Java.

Для створення об'єкта класу `Scanner` ми зазвичай передаємо попередньо визначений об'єкт `System.in`, який являє собою стандартний вхідний потік.

```
Scanner sc = new Scanner(System.in);
```

Ми можемо передавати об'єкт класу Файл, якщо ми хочемо читати вхідні дані з файлу:

- щоб прочитати числові значення певного типу даних XYZ, функція, яку слід використовувати, буде nextXYZ (). Наприклад, щоб прочитати значення типу short, ми можемо використовувати nextShort();
- щоб прочитати рядки, ми використовуємо nextLine();
- щоб прочитати один символ, ми використовуємо next() charAt (0). Функція next () повертає наступну послідовність символів на вході, як рядок, а функція charAt (0) повертає перший символ у цьому рядку.

Виведення на консоль

Для створення потоку виведення в клас System визначено об'єкт out. У цьому об'єкті визначено метод println(), який дає змогу вивести на консоль деяке значення з подальшим переведенням консолі на наступний рядок.

Розглянемо фрагмент коду, щоб прочитати дані різних типів даних.

```
import java.util.Scanner;
public class Lr1 {

    public static void main(String[] args) {
        double y, s, a, b, f;
        Scanner sc = new Scanner(System.in);
        String name = sc.nextLine();
        char gender = sc.next().charAt( 0 );
        int age = sc.nextInt();
        long mobileNo = sc.nextLong();
        double cgpa = sc.nextDouble();

        // Print the values to check if input was
        correctly obtained.
        System.out.println( "Name: " +name);
        System.out.println( "Gender: " +gender);
        System.out.println( "Age: " +age);
        System.out.println( "Mobile Number: "
+mobileNo);
        System.out.println( "CGPA: " +cgpa);
    }
}
```

```
}
```

Вхідні дані програми:

Masha

f

18

0686000400

23,78

Вихідний результат

Name: Masha

Gender: f

Age: 18

Mobile Number: 686000400

CGPA: 23.78

Недолік цієї програми в тому, що, відправивши її на виконання, ми отримуємо порожню консоль, яка щось чекає. Ми маємо пам'ятати послідовність, у якій вводяться значення різних типів, що не завжди зручно і призводить до помилок при виконанні. Бажано перед кожним оператором введення помістити виведення підказки. Наприклад таким чином:

```
Scanner sc = new Scanner(System.in);

System.out.print( "input Name: " );
String name = sc.nextLine();
System.out.print( "input Gender: " );
char gender = sc.next().charAt( 0 );
System.out.print( "input Age: " );
int age = sc.nextInt();
System.out.print( "input Mobile Number: " );
long mobileNo = sc.nextLong();
System.out.print( "input CGPA: " );
double cgpa = sc.nextDouble();
```

Тоді введення даних матиме вигляд:

input Name: Sid

input Gender: f

input Age: 23

input Mobile Number: 0681234000

input CGPA: 67,78

Відповідний вихідний результат:

Name: Sid

Gender: f

Age: 23
Mobile Number: 681234000
CGPA: 67.78

3.5 Оператори передачі управління Java. Розгалуженим називається обчислювальний процес, у якому вибір послідовності дій залежить від значень вихідних даних чи проміжних результатів.

Алгоритми обчислювальних процесів, що розгалужуються, подані у вигляді структур, що мають кілька варіантів обчислень. Кожен варіант поданий окремою обчислювальною гілкою. Вибір гілки обчислень заздалегідь не очевидний, він може залежати від вихідних значень величин, отриманих у процесі введення, чи від значень величин, отриманих у процесі обчислень. При виконанні розгалуженого алгоритму на ПЕОМ (реалізованого у вигляді програми алгоритмічною мовою) обчислення буде виконане за однією з гілок, тією, яка відповідає даному набору величин. Вибір гілки виконання алгоритму здійснюється шляхом виконання однієї чи послідовності перевірок значень, що гарантує для визначеного набору даних виконання дій за єдиною гілкою. Природно, що при розробці алгоритму розгалуженого обчислювального процесу необхідно передбачити всі можливі гілки обчислень, які залежать від варіантів наборів даних, що можуть вплинути на послідовність дій в алгоритмі.

Для реалізації розгалуження в програмі служать оператори передчі управління. Найчастіше використовується оператор if. Оператор if має такий вигляд:

```
if (вираз) оператор1;  
else оператор2;
```

Гілки else може не бути. Приклад оператора if:

```
if (j != 2)  
j++;  
else {  
j--;  
System.out.println (j); }
```

У частині (вираз) перевіряються деякі умови, варіантами яких є операції відношення (>, >=, ==, !=, <, <=). У той же час

умови можуть бути описані більш складними структурами – логічними виразами. Результатом виконання операції відношення будуть «істина» (при задоволенні зазначеної умови) чи «неправда» (при невиконанні). Якщо результат «істина», то буде виконуватись оператор1, якщо «неправда», то буде виконуватись оператор 2.

Приклад 1. Для закріплення викладеного матеріалу напишемо програму для розрахунку за двома формулами. Перед цим приготуйте тестові приклади за другою формулою з допомогою калькулятора (результат обчислення за першою формулою має збігатися з результатом за другою).

$$z_1 = \left(\frac{1+a+a^2}{2a+a^2} + 2 - \frac{1-a+a^2}{2a-a^2} \right) / (10-4a^2),$$

$$z_2 = 1/(4-a^2).$$

Складемо схему алгоритму до цієї задачі. Вхідні дані – змінна А. Для полегшення обчислень введемо допоміжні змінні у і b, результатом будуть Z1 і Z2. Напишемо програму таким чином, щоб було видно, якому блоку який оператор відповідає.

Розв'язання

Лістинг програми

```

package Lin2;
import java.util.Scanner;
public class trig_fun {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        double A,y,b,Z1,Z2; // Об'явлення змінних
        System.out.print( "input A: " ); //
        Запрошення до введення.
        A = sc.nextDouble(); // Ввід значення в
        змінну А.

        y=(1+A+Math.pow(A,2))/ (2*A+Math.pow(A,2)); //
        Допоміжні
        b=(1-A+Math.pow(A,2))/ (2*A-Math.pow(A,2));
        // розрахунки.
        //===== Розрахунок по першій формулі,
    }
}

```



```

Z1=(y+2-b) / (10-4*A*A);
//===== по другій формулі.
Z2=1 / (4-A*A);
/* Виведення результатів на екран*/
System.out.println(" Z1="+Z1+"\n Z2="+Z2);

}

}

```

Вхідні дані:

input A: 5,6

Результат:

Z1=-0.036549707602339186

Z2=-0.036549707602339186

Приклад 2. Скласти програму обчислення значення функції

$$y = \begin{cases} x + 4x - 7 & , \text{при } x < 2 \\ \frac{1}{x + 4x - 7} & , \text{при } x \geq 2 \end{cases}$$

Підібрати вхідні дані для обчислення за кожною гілкою.

Розв'язання

Ця задача досить традиційна за постановкою: є деяка величина x , що може набувати будь-яких значень. Але якщо x має значення до 2, то розрахунок величини y треба виконати за першою формулою: $y = x + 4x - 7$; якщо x має значення більше чи рівне 2, то

розрахунок повинен виконуватися за другою формулою: $y = \frac{1}{x + 4x - 7}$.

```

package rozgal;
import java.util.Scanner;
public class lr2 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double x, y, b; // Об'явлення

```

змінних

```

        System.out.print( "input x: " ); //
Запрошення до введення.
        x = sc.nextDouble();
        // Допоміжні розрахунки.
        b=x+4*x-7;
        //== Розрахунок y,
        if(x<2) y=b;
        else y=1/b;
        /* Виведення результатів на екран*/
        System.out.println(" y="+y);

    }

}

```

Вхідні дані:

input x: 5,7

Результат:

y=0.046511627906976744

Обчислення виконане за другою гілкою обчислень.

Вхідні дані:

input x: 1,5

Результат:

y=0.5

Обчислення виконане за першою гілкою обчислень.

Варіанти завдань

Напишіть програму для розрахунку за двома формулами. Перед цим приготуйте тестові приклади за другою формулою з допомогою калькулятора (результат обчислення за першою формулою має збігатися з результатом за другою).

Рівень 1

Варіант 1

$$z_1 = 2 \sin^2(3\pi - 2\alpha) \cos^2(5\pi + 2\alpha),$$

$$z_2 = \frac{1}{4} - \frac{1}{4} \sin\left(\frac{5}{2}\pi - 8\alpha\right).$$

Варіант 2

$$z_1 = \cos \alpha + \sin \alpha + \cos 3\alpha + \sin 3\alpha,$$

$$z_2 = 2\sqrt{2} \cos \alpha \sin \left(\frac{\pi}{4} \pi + 2\alpha \right) \cdot$$

Варіант 3

$$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha + 1 - 2 \sin^2 2\alpha},$$

$$z_2 = 2 \sin \alpha \cdot$$

Варіант 4

$$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha - \cos 3\alpha + \cos 5\alpha},$$

$$z_2 = \operatorname{tg} 3\alpha \cdot$$

Варіант 5

$$z_1 = \cos \alpha + \cos 2\alpha + \cos 6\alpha + \cos 7\alpha,$$

$$z_2 = 4 \cos \frac{\alpha}{2} * \cos \frac{5}{2} \alpha * \cos 4\alpha \cdot$$

Рівень 2

Варіант 1

$$f = \begin{cases} -4, & x < 0, \\ x^2 + 3x + 4, & 0 \leq x < 1, \\ (x^2 + 3x + 4)^2, & x \geq 1. \end{cases}$$

Варіант 2

$$y = \begin{cases} x, & \text{якщо } x = g \\ 2 - \frac{x^2}{9} - g, & \text{якщо } x < g \text{ або } x > g \end{cases}; \quad g = \begin{cases} am^2, & a < m \\ m^3 + e^{-am}, & a \geq m \end{cases}$$

Варіант 3

$$t = \begin{cases} a \sin^2(x) + \sqrt{|\cos(x)|}, & \text{якщо } x > 0 \\ \frac{x^2 + ab}{a} + \operatorname{tg} \frac{a}{b}, & \text{якщо } x \leq 0 \text{ і } a \neq 0 \\ a^2 x^2 + 2ax + b^2, & \text{в інших випадках} \end{cases}; \quad x = \frac{1}{a} + 0,5; \quad a = \sqrt{b^2 + c^2}$$

Варіант 4

$$y = \begin{cases} 2,3 & \text{если } x > 2,3 \\ d - \frac{x^2}{2} & \text{если } x \leq 2,3 \end{cases}; \quad x = \begin{cases} a^5 + b & \text{если } b < 0 \\ a^2 + \frac{a}{a-b} & \text{если } 0 \leq b < 1,5 \text{ и } a \neq b \\ a^3 + ba^2 + b^2 & \text{если } 1,5 \leq b < 2,5 \\ a^4 & \text{если } 2,5 \leq b \end{cases}$$

Варіант 5

$$s = \begin{cases} x + \sqrt{12,5 + r^2}, & 0 < x < 3 \text{ и } r \neq 0 \\ 0,14t^2 + b^5x, & 3 \leq x < 5 \text{ и } a \leq 0; \\ r\frac{a}{c} + \sqrt[5]{\frac{r}{c}}, & \text{в остальных случаях} \end{cases}; \quad r = \begin{cases} \frac{b}{x}, & \text{если } b < 0 \\ \sqrt{\frac{a}{c}} + \sqrt{\frac{b}{c}}, & \text{если } b \geq 0 \end{cases}$$

Варіант 6

$$p = \begin{cases} 2,45x^2 + e^{-ax}, & a < 0 \\ \frac{\sqrt{a+b}}{\log(a^2 + b^2)}, & a \geq 0 \end{cases}; \quad a = 0,73t - \sin x; \quad x = \begin{cases} 2zb - b^2, & b < \frac{z}{2} \\ z^2 + b^2, & b \geq \frac{z}{2} \end{cases}$$

ЛАБОРАТОРНА РОБОТА 3

Вивчення циклічних операторів мови Java

Мета роботи: вивчення правил запису циклічних операторів мови програмування Java. Створення програми з їх використанням.

1 Завдання та порядок виконання

- 1.1 Вивчити теоретичний матеріал.
- 1.2 Підготувати відповіді на контрольні запитання.
- 1.3 Записати мовою Java програму згідно з варіантом завдання.

Контрольні запитання

- 1 Які бувають цикли? У чому їх відмінність?
- 2 Які оператори призначені для організації циклів мовою Java?
- 3 Формат запису циклу for.
- 4 Засоби задання приросту у циклі for.

5 Формат запису циклу while.

6 Формат запису циклу do-while.

2 Зміст звіту

2.1 Номер роботи, її назва, визначення мети.

2.2 Стислий зміст теоретичного матеріалу та відповіді на контрольні запитання.

2.3 Текст програми, вхідні дані та результати виконання завдання з рівнів 1 та 2.

2.4 Висновки до роботи.

3 Навчальний матеріал

3.1 Циклічні обчислення. Циклічний алгоритм – це алгоритм, що містить багаторазово виконувану послідовність операцій. Використання циклів дає змогу значно скоротити схему алгоритму і розмір відповідної йому програми алгоритмічною мовою. За своєю організацією цикли можуть мати різну структуру.

Арифметичними називаються цикли, кількість повторень яких задана змінною, яка називається параметром циклу. Отже, кількість повторень арифметичного циклу може бути визначена до його виконання. Границі параметра циклу і крок його зміни можуть бути задані константами, змінними або виразами.

Наприклад, параметр x змінюється в інтервалі від 1 до 3 із кроком 0,5. Формально це можна подати так: $x \in [1,3]$, $h_x=0,5$. Другий приклад: параметр z змінюється в інтервалі від z_0 до z_{\max} із кроком h : $z \in [z_0, z_{\max}]$, $h_z=h$.

Кількість повторень циклу розраховується за формулою

$$N = \left[\frac{x_m - x_0}{h} \right] + 1,$$

де $[]$ – ціла частина числа;

x_0 – початкове значення параметра циклу;

x_m – кінцеве значення параметра циклу;

h – крок зміни параметра.

Арифметичні цикли з одним параметром називаються простими. Якщо цикл містить у собі кілька інших циклів, він називається вкладеним (кратним).

Для організації простого арифметичного циклу необхідно виконати такі дії:

- сформувати вихідні дані;
- задати початкове значення параметра циклу;
- виконати задані обчислення і надрукувати проміжні результати;
- змінити параметр циклу на величину кроку;
- перевірити, чи є нове значення параметра припустимим, тобто чи не перевищує обчислене значення кінцеве значення параметра. Якщо не перевищує, то провадяться чергові обчислення, процес повторюється; у протилежному випадку здійснюється вихід з циклу. Остання дія є умовою закінчення циклу, тому що, якщо її не задати, робота циклу буде нескінченною.

3.2 Оператори циклу. Оператор `while` перевіряє деякий вираз `i`, якщо він істинний, то виконується оператор або блок знову і знову до тих пір, поки вираз не стане хибним. Коли вираз стане хибним, управління передається оператору, наступному за оператором `while`.

```
while (вираз)  
оператор;
```

Приклад циклу `while`:

```
while (i == j)  
{  
i ++;  
}
```

Оператор `do`

Спочатку виконується оператор, що стоїть у циклі, потім обчислюється вираз `while`. Якщо результат правдивий, то оператор повторюється до тих пір, поки вираз не стане хибним:

```
do
```

оператор;
while (вираз);

Основною відмінністю оператора do від циклу while є те, що оператор циклу do виконується хоча б раз, незалежно від значення виразу. Тому оператор do називається циклом з постумовою.

Синтаксис циклу for такий:
for (ініціалізація; вираз; крок)
 оператор;

Спочатку в циклі for виконуються оператори ініціалізації, а потім обчислюється вираз. Якщо вираз істинний, то виконується оператор циклу, параметр збільшується на крок, а потім перевіряється вираз.

```
for (int ball = 0; ball <4; ball ++)  
{  
    System.out.println (ball);  
}
```

Приклад 1. Для закріплення викладеного матеріалу напишемо програму для розрахунку значень функції на відрізку:

$$t = \cos(x + ax^2); \quad a = \sqrt[3]{b + d^2} - e^{-x}; \quad x \in [-3c, 4c], \quad h_x = 0,5c.$$

Розв'язання

Текст програми:

```
package for1;  
import java.util.Scanner;  
public class for1 {  
  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        double a,b,c,d,x,t; // Об'явлення  
змінних  
        System.out.print( "input c: " ); //  
Запрошення до введення.  
        c = sc.nextDouble(); // Ввід значення в  
змінну c.  
        System.out.print( "input b: " ); //  
Запрошення до введення.
```

```

        b = sc.nextDouble(); // Ввід значення в
змінну b.
        System.out.print( "input d: " ); //
Запрошення до введення.
        d = sc.nextDouble(); // Ввід значення в
змінну d.
        for (x=-3*c; x<=4*c; x=x+0.5*c)
        {
            a=Math.pow(b+d*d,1./3)-Math.exp(-x);
            t= Math.cos(x +a*Math.pow(x,2));
            /* Виведення результатів на екран*/
            System.out.println(" a="+a+ "    t="+t+"
x="+x);
        }

    }

}

```

Вхідні дані:

```

input c: 2
input b: 1
input d: 4

```

Результат:

```

a=-400.85751190207685    t=-0.31324786183369396    x=-6.0
a=-145.84187751191837    t=0.8696643655610252    x=-5.0
a=-52.026868442486    t=0.7205166603239245    x=-4.0
a=-17.514255332529434    t=-0.918283756756924    x=-3.0
a=-4.817774508272415    t=-0.7517724015349309    x=-2.0
a=-0.14700023780080995    t=0.411223673641774    x=-1.0
a=1.5712815906582351    t=1.0    x=0.0
a=2.203402149486793    t=-0.9980904011794335    x=1.0
a=2.4359463074216223    t=0.6803286361168241    x=2.0
a=2.521494522290371    t=0.8468780361335202    x=3.0
a=2.552965951769501    t=0.6485197951238791    x=4.0
a=2.5645436436591496    t=0.9999989526796338    x=5.0
a=2.568802838481569    t=-0.46467398958920914    x=6.0
a=2.5703697086926804    t=0.5392717091765766    x=7.0
a=2.5709461280303327    t=-0.9696395738843033    x=8.0

```

Приклад 2. Напишемо програму для розрахунку нижченаведеної задачі. Дано функцію

$$z = x^4 + \sqrt[5]{a * s - b^2}$$

$$s = \cos(x+a) + b^3 \quad x \in [-T; T] \quad h_x = 3.$$

Знайти МАХ у і відповідний йому аргумент.

Розв'язання

Позначимо максимальне значення у – МАХ , а відповідний йому аргумент – хmax.

Текст програми:

```

package for2;
import java.util.Scanner;
public class for2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double a,b,Z,S,x,T; // Об'явлення
змінних
        System.out.print( "input T: " ); //
Запрошення до введення.
        T = sc.nextDouble(); // Ввід значення в
змінну T.
        System.out.print( "input b: " ); //
Запрошення до введення.
        b = sc.nextDouble(); // Ввід значення в
змінну b.
        System.out.print( "input a: " ); //
Запрошення до введення.
        a = sc.nextDouble(); // Ввід значення в
змінну a.
        double MAX, xmax;
        x=-T;
        S= Math.cos(x+a)+Math.pow(b,3);
        Z=Math.pow(x,4) +Math.pow(a*S-b*b,1./5);
        MAX=Z; xmax=x;
        for (x=-T; x<=T; x=x+3)
        {
            S= Math.cos(x+a)+Math.pow(b,3);
            Z=Math.pow(x,4) +Math.pow(a*S-
b*b,1./5);
            /* Виведення значень функції на
екран*/
            System.out.println(" S="+S+" Z="+Z+"
x="+x);
            if (Z>MAX) {MAX=Z; xmax=x;}
        }
    }
}

```

```

    }
    /* Виведення результатів на екран*/
    System.out.println(" MAX"+MAX+"
xmax="+xmax);
}

```

Вхідні дані:

```

input T: 11
input b: 22
input a: 1

```

Результат:

```

S=10647.160928470923    Z=14647.330029886498    x=-11.0
S=10648.753902254344    Z=4102.330228307827    x=-8.0
S=10647.346356379136    Z=631.3300529847378    x=-5.0
S=10648.540302305868    Z=22.330201703193232    x=-2.0
S=10647.583853163453    Z=7.330082568560343    x=1.0
S=10648.283662185464    Z=262.3301697371606    x=4.0
S=10647.854499966192    Z=2407.330116281046    x=7.0
S=10648.004425697987    Z=10006.330134955884    x=10.0
MAX14647.330029886498    xmax=-11.0

```

Варіанти завдань

Рівень 1

- 1 $y = \frac{ax^2 + b}{\sqrt{|a+b|}}$; $x \in [c, d]$, $h_x = h$.
- 2 $z = \frac{ax}{by}$; $x \in [a, b]$, $h_x = 0,5$.
- 3 $x = a - \sin(\omega + \varphi)$; $a \in [k, l]$, $h_a = h$.
- 4 $y = \sin^2(\alpha + \beta)$; $\alpha \in [0, 0,5]$, $h_\alpha = 0,1$.
- 5 $z = \frac{x\sqrt{\ln(a)+1}}{0,25\sin(\alpha)}$; $x \in [3, c]$, $h_x = 0,5$.
- 6 $y = 2\sin(\alpha - \beta) + \cos(\gamma - \omega)$; $\alpha \in [0, 1]$, $h_\alpha = 0,2$.
- 7 $y = \frac{a}{\ln(\sin^2 \alpha)}$; $a \in [c, 0]$, $h_a = 0,25c$.
- 8 $r = 0,14t^2 + b^5x$; $t \in [-10x, 10x]$, $h_t = \frac{b}{x}$.
- 9 $t = \cos(x + ax^2)$; $a = b + d^2 - e^{-x}$; $x \in [-3, 4]$, $h_x = 0,1$.

Рівень 2

1 Дано функцію

$$y = \cos^3(z+a) - x$$

$$z = x^5 + ax + b^3 \quad x \in [0; T] \quad hx = 0,2.$$

Знайти МАХ у і відповідний йому аргумент.

2 Дано функцію

$$z = x^5 + ay + b^3$$

$$y = x^5 + bx + b^3 \quad x \in [0; 10] \quad hx = 2.$$

Знайти МІН z і відповідний йому аргумент.

3 Дано функцію

$$z = x^5 + a*s + b^3$$

$$s = \cos(x+a) + b^3 \quad x \in [-T; T] \quad hx = 3.$$

Знайти МАХ і МІН .

4 Дано функцію

$$y = \cos^3(z+a) - x$$

$$z = x^5 + ax + b^3 \quad a \in [1; 2] \quad ha = 0,2.$$

Знайти МАХ у і відповідний йому аргумент.

5 Дано функцію

$$z = x^5 + a*s + b^3$$

$$s = \cos(x+a) + b^3 \quad a \in [1; 2] \quad ha = 0,2.$$

Знайти МІН z і відповідний йому аргумент.

ЛАБОРАТОРНА РОБОТА 4

Робота з одновимірними масивами

Мета роботи: отримання навичок обробки одновимірних масивів.

1 Завдання та порядок виконання

1.1 Вивчити теоретичний матеріал.

1.2 Підготувати відповіді на контрольні запитання.

1.3 Написати програму згідно з варіантом індивідуального завдання. Відладити програму і запустити на різних наборах вхідних даних.

Контрольні запитання

1 Який вигляд має опис масиву стандартного типу даних?

2 Як оголошується символічний масив? Як звернутись до елемента масиву з номером 3 (третього символу у рядку)?

3 Які функції змінюють регістр символів?

4 Які функції визначають, чи є символ цифрою; літерою?

2 Зміст звіту

2.1 Номер роботи, її назва, визначення мети.

2.2 Стислий зміст теоретичного матеріалу та відповіді на контрольні запитання.

2.3 Результати виконання завдання: схеми алгоритмів, програми, результати роботи програм.

2.4 Висновки до роботи.

3 Навчальний матеріал

3.1 Оголошення і створення масиву. Масив – об'єднання однорідних елементів.

Масиви можуть бути багатовимірними, наприклад:

```
int [] [] dim = new int [3] [8];
```

Масиви можна створювати не тільки з базових типів, але і з довільних об'єктів.

Масиви в Java – це структура даних, яка зберігає впорядковані колекції фіксованого розміру елементів потрібного типу. В Java масив використовується для зберігання колекції даних, але часто буває корисно думати про масив, як про сукупність змінних одного типу.

Щоб використовувати масив у програмі, необхідно оголосити змінну для посилання на масив і вказати тип масиву, який може посилатися на змінну. Синтаксис для оголошення змінної масиву:

```
dataType [] arrayRefVar;
```

Приклади оголошення посилання на масив стандартних типів:

```
char[] buf;
```

```
int[] Arr;
```

Приклад оголошення посилання на масив об'єктів типу (класу) Monstr:

```
Monstr[] stado;
```

У Java є три способи присвоєння значень елементам масиву.

1 Оголошення масиву й ініціалізація його елементів. Це може бути застосовано тільки для базових типів.

```
int fib [] = {1,1,2,3,5};
```

2 Резервування місця для масиву і визначення його розміру. Для цього потрібно використовувати ключове слово `new`, за яким ідуть тип змінної та її розмір:

```
IQ = new int [500];
```

3 Запам'ятовування даних у масиві. Елементи масивів базових типів ініціалізуються нульовим значенням. Оператор, який присвоює значення четвертому елементу масиву:

```
IQ [4] = 12;
```

Індекси масиву повинні або мати тип `int` або перетворюватися до нього.

Щоб використовувати масив, його потрібно створити, тобто виділити місце в пам'яті і розмістити там масив. В Java створити масив можна оператором `new` таким чином:

```
arrayRefVar = new dataType [arraySize];
```

де `arrayRefVar` – ім'я масиву; `dataType` – тип масиву; `arraySize` – розмір (кількість елементів) масиву.

Приклади створення масивів стандартних типів:

```
buf = new char[180];
```

```
Arr=new int[num];
```

де змінна `num` на момент виконання цього оператора має конкретне значення – розмір масиву.

Приклад створення масиву об'єктів класу `Monstr` з п'яти об'єктів:

```
stado = new Monstr[5];
```

Нумерація елементів у масиві йде від нульового елемента. Наприклад, елементи масиву `buf`:

```
buf[0], buf[1], buf[2], buf[3], . . . buf[178], buf[179].
```

Також створити масив і наповнити його значеннями можна при оголошенні з допомогою ініціалізації. Приклад:

```
double[] Lt = {1.9, 2.9, 3.4, 3.5};
```

масив `Lt` матиме чотири значення типу `double`.

Отримати значення розміру масиву можна, звернувшись до його властивості `length`. Наприклад: `buf.length`.

Обробка і виведення масивів відбувається з використанням циклів.

3.2 Приклад розв'язання задачі. Для прикладу розглянемо нижченаведену задачу.

Задано масив значень $A=\{a_i\}$ з 8 елементів і змінні m і k . Знайти кількість елементів масиву, більших, ніж k ; кількість елементів масиву, більших, ніж $2*m$, знайти індекс мінімального елемента. Знайти положення (індекс) мінімального елемента.

Розв'язання

Назвемо масив `Arr`. Позначимо змінною `num` розмір масиву, значення введемо з клавіатури за допомогою `in.nextInt()`. Заповнення проводитимемо генератором випадкових чисел `Math.random()`, що повертає значення типу `double` з діапазону $[0; 1]$.

Для отримання цілих значень з діапазону $(-6; 14)$ використовуємо формулу:

```
Arr[i]=(int)(Math.random()*20)-6;
```

Кількість елементів масиву, більших, ніж k , записуємо в змінну `nk`, а кількість елементів масиву, більших, ніж $2*m$, – у змінну `n2m`. Індекс мінімального елемента масиву присвоюємо змінній `ind`.

```
import java.util.Scanner;
public class Test {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Input a number: ");
        int num = in.nextInt();
        int[] Arr=new int[num];
        /*заповнення масива генератором випадкових чисел */
        System.out.print("Your number: \n"+ num);
        for(int i=0;i<num;i++)
            {Arr[i]=(int) (Math.random() *20) -6;
            }
        /*=====Розрахунок кількості ел-тів більших k в nk,
        *та більших 2*m в змінну n2m */
        System.out.print(" \n k=");
        int k=in.nextInt(), nk=0;
        System.out.print(" \n m=");
```

```

    int m=in.nextInt(),n2m=0;
    System.out.print(" \n Arr={");
    for(int i=0;i<num;i++)
        {if(Arr[i]>k)nk++;
         if(Arr[i]>2*m)n2m++;
          /* виведення масиву*/
          System.out.print(" "+ Arr[i]);
        }
    System.out.println("} ");
/* =====Обчислення індекса мін ел-та масива*/
    int ind=0;
    for(int i=0;i<num;i++)
        if(Arr[i]<Arr[ind])ind=i;
/*дружк результатів*/
    System.out.println("min= "+ Arr[ind]);
    System.out.println(">k "+ nk);
    System.out.println("ind= "+ ind);
    System.out.println(">2*m= "+n2m);
}
}

```

Результат роботи програми:

Input a number: 12

k=3

m=2

Arr={ 0 13 -5 6 3 3 -2 4 2 7 12 -6}

min= -6

>k 5

ind= 11

>2*m= 4

Варіанти завдань

Рівень 1

Усі елементи масивів є цілими числами (int), якщо в завданні не обговорене інше. Усі введені масиви і всі знайдені значення необхідно після обробки вивести на екран.

1 Задано масив значень $A=\{a_i\}$ з 8 елементів і змінні m і k . Знайти кількість елементів масиву, більших, ніж k ; кількість елементів масиву, більших, ніж $2*m$, знайти індекс мінімального елемента.

2 Задано масив значень $A=\{a_i\}$ з 12 елементів. Подвоїти додатні елементи, а від'ємні замінити на їх модуль.

3 Задано масив значень $A=\{a_i\}$ з 12 елементів. Знайти по два максимальних і мінімальних елементи.

4 Задано масив значень $A=\{a_i\}$ з 10 елементів. Знайти максимальний і мінімальний елементи, поміняти їх місцями.

5 Задано масив значень $A=\{a_i\}$ з 12 елементів. Знайти суму всіх додатних елементів і добуток від'ємних.

6 Задано масив значень $A=\{a_i\}$ з 12 елементів. Знайти суму абсолютних значень відхилень елементів від середнього значення.

7 Задано масив значень $A=\{a_i\}$ з 8 елементів. Додати елемент на позицію з індексом $k=4$.

8 Задано масив значень $A=\{a_i\}$ з 10 елементів. Знайти максимальний і мінімальний елементи і видалити їх.

Рівень 2

1 Задано масив цілих чисел. У ньому – послідовність чисел, розділених одним або декількома нулями. Залишити по одному нулю як роздільник.

2 Видалити з масиву нульові елементи, які містяться в ньому, а від'ємні замінити на їх модуль.

3 Задано масив з 10 дійсних чисел. Визначити, чи утворюють вони послідовність, яка зростає або спадає.

4 Задано масив цілих чисел. Визначити, скільки з них набуває найбільшого значення.

5 Задано масив цілих чисел. Визначити довжину найбільшої підпослідовності нулів.

6 Задано масив дійсних чисел. Визначити, скільки з них є більшими від «сусідів», тобто попереднього і наступного чисел.

7 Задано послідовність ненульових цілих чисел, за якою йде 0. Визначити, скільки разів у цій послідовності змінюється знак.

8 Задано масив цілих чисел. Визначити, скільки елементів масиву відмінні від останнього і першого чисел.

9 Задано масив цілих чисел. Надрукувати спочатку всі від'ємні числа, потім усі нулі, потім додатні числа.

10 Задано масив цілих чисел. Визначити, яке число зустрічається в ньому найчастіше.

ЛАБОРАТОРНА РОБОТА 5

Робота з двовимірними масивами

Мета роботи: отримання навичок обробки двовимірних масивів.

1 Завдання та порядок виконання

1.1 Вивчити теоретичний матеріал.

1.2 Підготувати відповіді на контрольні запитання.

1.3 Написати програму згідно з варіантом індивідуального завдання. Відладати програму і запустити на різних наборах вхідних даних.

Контрольні запитання

1 Який вигляд має опис масиву стандартного типу даних?

2 Як оголошується двовимірний масив? Як звернутись до елемента масиву, розташованого на перетині третього рядка і другого стовця?

3 У якому порядку відбувається заповнення масиву, якщо зовнішній цикл організований по рядках, а внутрішній – по стовпцях?

4 Як за допомогою циклів організувати обробку елементів, розташованих у рядках з парним індексом; у стовпцях з непарним індексом?

2 Зміст звіту

2.1 Номер роботи, її назва, визначення мети.

2.2 Стислий зміст теоретичного матеріалу та відповіді на контрольні запитання.

2.3 Результати виконання завдання: схеми алгоритмів, програми, результати роботи програм.

2.4 Висновки до роботи.

3 Навчальний матеріал

Для прикладу розглянемо нижченаведену задачу.

До двовимірного масиву Z , що складається із n рядків та m стовпців однорідних змінних, вводяться довільні числа. Треба

обчислити та вивести добуток усіх елементів із парних рядків масиву Z , що мають значення, більші за число a та менші за число b (вважається, що $a < b$), та середнє арифметичне від усіх додатних значень елементів цього масиву.

Розв'язання

Позначимо змінною n кількість рядків, m – кількість стовпців, значення введемо з клавіатури за допомогою `in.nextInt()`. Заповнення проводимо генератором випадкових чисел `Math.random()`, що повертає значення типу `double` з діапазону $[0; 1)$. Для отримання цілих значень з діапазону $(-6; 14)$ використовуємо формулу

$$Z[i][j] = (\text{int})(\text{Math.random()} * 20) - 6;$$

Суму і кількість додатних елементів масиву позначимо відповідно S і K , а добуток елементів із парних рядків масиву Z , що мають значення, більші за число a та менші за число b , в змінну P .

```
import java.util.Scanner;
```

```
public class Mas2 {  
    public static void main(String[] args) {  
  
        Scanner in = new Scanner(System.in);  
        System.out.print("Input n: ");  
        int n = in.nextInt();  
        System.out.print("Input m: ");  
        int m = in.nextInt();  
        int[][] Z = new int[n][m];  
        System.out.print("Input a: ");  
        int a = in.nextInt();  
        System.out.print("Input b(a<b): ");  
        int b = in.nextInt();  
        System.out.println("Your mas Z=: ");  
        for (int i = 0; i < n; i++) {  
            for (int j = 0; j < m; j++) {  
                Z[i][j] = (int) (Math.random() * 20)  
- 6;  
  
                System.out.print(" " + Z[i][j]);  
            }  
            System.out.print(" \n");  
        }  
        long P = 1, S = 0, K = 0;
```

```

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++)
            if (0 < Z[i][j]) {
                S = S + Z[i][j];
                K++;
            }
    }
    for (int i = 0; i < n; i += 2) {
        for (int j = 0; j < m; j++) {
            if (a < Z[i][j] && Z[i][j] < b)
                P = P * Z[i][j];
        }
    }
    System.out.print(" \nP=" + P);
    if (0 < K) {
        double R = (double) S / K;
        System.out.print(" \nR=" + R + "   S=" + S
+ " K=" + K);
    } else
        System.out.print("\n додатніх елементів
немає");
}
}

```

Результат роботи програми:

```

Input  n: 5
Input  m: 7
Input  a: 1
Input  b(a<b): 5
Your mas Z=:
-5 5 -6 -2 2 8 12
 4 -3 13 1 8 -4 -4
 9 12 8 -6 6 -4 0
-4 5 1 -1 8 4 -5
-4 12 4 1 2 -5 -6

```

P=16

R=6.25 S=125 K=20

Варіанти завдань

Скласти програму розв'язання нижченаведеної задачі.

1 До двовимірного масиву N , що складається із 4 рядків та 5 стовпців однорідних змінних, вводяться довільні числа. Треба підрахувати та вивести у вигляді вектора (одновимірного масиву)

добутки значень елементів кожного стовпця із цього масиву, значення у яких менші за 55 та більші або дорівнюють 10.

2 До двовимірного масиву K , що складається із 5 рядків та 4 стовпців однорідних змінних, вводяться довільні числа. Треба знайти та вивести різницю між сумою елементів масиву K , що за значеннями менші за 7, та сумою елементів масиву K , які за значеннями більші, ніж 7.

3 До двовимірного масиву Z , що складається із n рядків та m стовпців однорідних змінних, вводяться довільні числа. Треба обчислити та вивести добуток усіх елементів із парних рядків масиву Z , що мають значення, більші за число a та менші за число b (вважається, що $a < b$), та середнє арифметичне від усіх додатних значень елементів цього масиву.

4 До двовимірного масиву X , що складається із n рядків та m стовпців однорідних змінних, вводяться довільні числа. Треба підрахувати та вивести у вигляді вектора (одновимірного масиву) середні арифметичні додатних значень елементів кожного стовпця цього масиву.

5 До двовимірного масиву Q , що складається із n рядків та m стовпців однорідних змінних, вводяться довільні числа. Знайти та вивести різницю між сумою додатних та добутком від'ємних значень елементів цього масиву. Треба також обчислити кількість елементів у масиві, що мають значення нуль.

ЛАБОРАТОРНА РОБОТА 6

Робота з рядками

Мета роботи: отримання навичок обробки символічних рядків.

1 Завдання та порядок виконання

1.1 Вивчити теоретичний матеріал.

1.2 Підготувати відповіді на контрольні запитання.

1.3 Написати програму згідно з варіантом індивідуального завдання. Відладити програму і запустити на різних наборах вхідних даних.

Контрольні запитання

- 1 Які типи даних використовуються для роботи із символьними даними?
- 2 Як оголошується символьний масив? Як звернутись до елемента масиву з номером 3 (третього символу в рядку)?
- 3 Які функції змінюють регістр символів?
- 4 Які функції визначають чи є символ цифрою; літерою?

2 Зміст звіту

- 2.1 Номер роботи, її назва, визначення мети.
- 2.2 Стислий зміст теоретичного матеріалу та відповіді на контрольні запитання.
- 2.3 Результати виконання завдання: схеми алгоритмів, програми, результати роботи програм.
- 2.4 Висновки до роботи.

3 Навчальний матеріал

3.1 Типи даних для роботи з рядками в мові Java. У мові Java робота з рядками організована аналогічно C ++: створений клас рядків String. Цей клас дає змогу досить легко працювати з текстовими рядками в програмах.

Рядок розглядається як масив символів, при цьому індекс цього масиву починається з нуля. Потрібно знати, де рядок закінчується. Існують різні розв'язання цієї проблеми. У Паскалі, наприклад, довжина рядка додається перед символами, а в C ++ і Java в кінці рядка повинен бути нульовий символ (символ з кодом 0).

В Java рядки подаються трьома класами.

String. Кращий для рядкових констант.

StringBuffer. Використовується для рядків, що змінюються в програмі.

StringTokenizer. Ділить рядок на лексеми.

Працюючи з класом String, можна шукати, порівнювати та об'єднувати символи; проте не можна вставляти нові символи в рядок або змінювати довжину рядка.

Усередині об'єкта класу String створюється масив символів, який багато в чому нагадує той, що використовується для рядків

у програмах на C ++. Оскільки цей масив «захований» у клас, до нього можна звертатися тільки за допомогою методів класу. Така інкапсуляція даних (одне з ключових властивостей об'єктно-орієнтованого програмування) забезпечує правильне поводження з рядками відповідно до правил класу.

Якщо з рядком C ++ програма може маніпулювати як завгодно, незалежно від того, чи мають ці операції сенс і чи не призведуть вони до фатальної помилки, то рядок Java захищений методами класу.

3.2 Створення рядків. Об'єкт String може створюватися явно або неявно.

Приклад неявного завдання:

```
g.drawString ( "This is a string", 50, 50);
```

Інший спосіб створити об'єкт String – явно отримати екземпляр класу String. Клас String має сім конструкторів. Найбільш очевидний приклад:

```
String str = new String ( "This is a string " );
```

Можна також оголосити об'єкт типу String і потім поставити його значення в програмі:

```
String str = "This is a string";
```

Створення порожніх рядків.

```
String str = new String ();
```

```
String str = "";
```

Це означає створення об'єкта класу String, у якому рядок не містить ніяких символів.

```
String str;
```

Об'єкт не створюється, маємо тільки посилання на об'єкт.

Деякі конструктори класу String:

```
public String () – створює порожній рядок;
```

```
public String (String value) – створює об'єкт String з іншого об'єкта String, разом з рядковою константою;
```

```
public String (char value []) – створює об'єкт String з масиву символів;
```

```
public String (char value [], int offset, int count) – створює об'єкт String з частини масиву символів, починаючи з символу під номером offset. Count – кількість символів;
```

`public String (StringBuffer buffer)` – створює об'єкт `String` з об'єкта `StringBuffer`.

3.3 Отримання інформації про об'єкт `String`. Для визначення довжини рядка можна викликати метод `length ()`:

```
int len = str.length ();
```

Якщо потрібно визначити, чи починається рядок з підрядка, то використовується метод `startsWith ()`:

```
String str = "This is a string";  
boolean result = str.startsWith ("This");
```

У цьому випадку булева змінна `result` дорівнює `true`, оскільки рядок `str` дійсно починається із символів "This".

Аналогічний метод `endsWith ()` визначає, чи закінчується рядковий об'єкт заданими символами.

Якщо потрібно визначити положення першого входження деякого символу в рядку, то використовується метод `indexOf ()`:

```
String str = "Hello, Serg";  
int index = str.indexOf ('l');
```

3.4 Приклад розв'язання задачі. Завдання 1. Для прикладу розглянемо задачу. Дано текст. Надрукувати його, замінивши у ньому всі літери `a` на літеру `o`.

```
public class zad_1 {  
    public static void main(String args[]) {  
        String s=new String("Пусть жгучая тоска всю ночь  
тебя терзает,\r\n" +  
            "Минута - от тоски не будет и  
следа,\r\n" +  
            "И счастье тебя охватит, засияет...");  
        /* ==== виведення на екран рядка */  
        System.out.println(s);  
        System.out.print(s.replaceAll("a", "o"));  
    }  
}
```

Результат на консолі:

Пусть жгучая тоска всю ночь тебя терзает,

Минута – от тоски не будет и следа,
 И счастье тебя охватит, засияет...
 Пусть жгучая тоска всю ночь тебя терзает,
 Минута – от тоски не будет и следа,
 И счастье тебя охватит, засияет...

Завдання 2. Написати програму, яка в заданому тексті знаходить найбільше слово. Також порахувати кількість слів довжиною 5 символів.

Розв'язання

Позначимо k_5 кількість слів довжиною 5 символів, dl – довжину поточного слова, $maxl$ – довжину слова з максимальною кількістю символів, nu – індекс, з якого починається поточне слово, $numax$ – індекс, з якого починається слово з максимальною кількістю символів.

```
import java.util.Scanner;
public class strin {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner in = new Scanner(System.in);
        /* =====ініціалізація змінної рядка s текстом
        */
        String s=new String("Зелень розы, вино мне
        судьбою даны. Только нету тебя в этом блеске весны.
        ");
        /* ===== виведення на екран рядка */
        System.out.print(s);
        int k5=0,kr=0, dl=0,maxl=0,nu,numax=0;
        for(int i=0;i< s.length();i++){
            dl=0;nu=i;
            /*=====визначення довжини слова */
            while(!(s.charAt(i)=='
            '||s.charAt(i)=='.'||s.charAt(i)==' ',' ')) {
                i++;dl++;
            }
            /*=====виведення довжини слова і номера символа,
            яким воно закінчується*/
            System.out.print(" \nlond =" + dl + " i=" + i);
            /*=====обчислення кількості слів з 5 символів*/

            if(dl==5)k5++;else kr++;
        }
    }
}
```



```

    /*визначення мах довжини слова і номера символа
    яким це слово починається*/
    if (maxl<dl) {maxl=dl; numax=nu;}
    }
    System.out.print("Your number: \n"+
k5+"ALL="+kr);
    /* Виведення слова мах довжини*/
    System.out.print("\n Max long word: "+ maxl);
    for(int i=numax;i<numax +maxl;i++){
        System.out.print(" "+s.charAt(i));
    }
} }

```

Результат програми:

Зелень розы, вино мне судьбою даны. Только нету тебя
в этом блеске весны.

```

lond =6 i=6
lond =4 i=11
lond =0 i=12
lond =4 i=17
lond =3 i=21
lond =7 i=29
lond =4 i=34
lond =0 i=35
lond =6 i=42
lond =4 i=47
lond =4 i=52
lond =1 i=54
lond =4 i=59
lond =6 i=66
lond =5 i=72
lond =0 i=73Your number:
1ALL=15

```

Max long word: 7 с у д ь б о ю

Результат для тексту: String s=new String("Любовь,
как море - не боясь плыви. Спасет Аллах достойного
любви. ");

Любовь, как море - не боясь плыви. Спасет Аллах
достойного любви.

```

lond =6 i=6
lond =0 i=7
lond =3 i=11
lond =4 i=16
lond =1 i=18

```

```

lond =2 i=21
lond =5 i=27
lond =5 i=33
lond =0 i=34
lond =6 i=41
lond =5 i=47
lond =10 i=58
lond =5 i=64
lond =0 i=65Your number:
4ALL=10
    Max long word: 10 д о с т о й н о г о

```

У завданнях другого рівня необхідно внести зміни у вхідний рядок (переставляються, додаються або видаляються символи) . Якщо використовувати рядок типу `String`, то це не можливо для випадків, непередбачених стандартними функціями. Це можна обійти різними способами. Наприклад, використовувати тип `StringBuffer`.

Приклад:

Дано текст. Перетворити слова за таким правилом: видалити повторні входження всіх букв.

```

import java.util.Scanner;
public class mmm {
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);
        System.out.println("Введіть символну строку: ");
        String s = in.nextLine();
        StringBuffer str = new
StringBuffer(s.subSequence(0, s.length()));

        for(int i=0;i<str.length();i++){
            char c1=str.charAt(i);
            for(int j=str.length()-1;j>i;j--){
                char c2=str.charAt(j);
                if(c1 == c2){
                    str.deleteCharAt(j);
                }
            }
        }

        System.out.println("Строка, в которой удалены
повторяющиеся буквы "+str);

```

```
}  
  
}
```

Результат наведений на рисунку 13

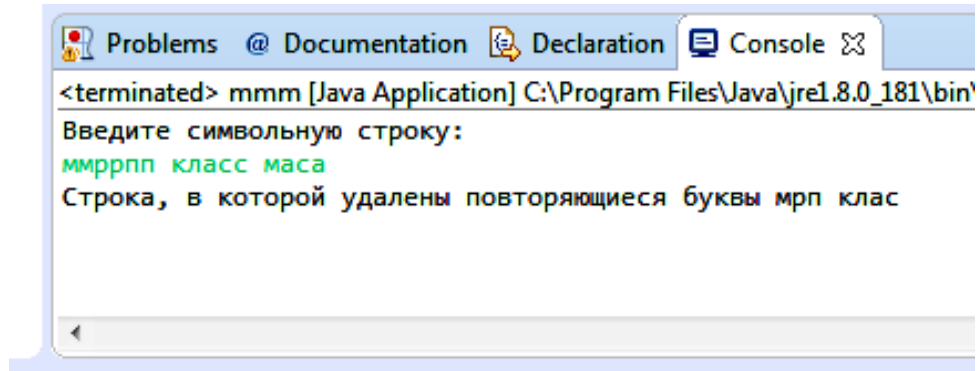


Рисунок 13

Нехай результат займає b_j символів. Тоді виведення можна організувати двома способами. Перший – посимвольно вивести масив `buf` від 0 до b_j-1 елемента. Другий – створити рядок `buf3` типу `String`, використовуючи конструктор `String (char value [], int offset, int count)`, який створює об'єкт `String` з частини масиву символів. Обидва способи подані при розв'язанні нижченаведеної задачі.

Дано текст. Претворити слова за таким правилом: видалити зі слова всі попередні входження останньої літери.

```
import java.util.Scanner;  
import java.io.*;  
import java.util.*;  
public class examp_str_buf {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        Scanner in = new Scanner(System.in);  
  
        System.out.print("Input : ");  
        char[] buf = new char[70];  
        /* =====ініціалізація змінної рядка з текстом */  
        String s=new String("Любовь, как море - не боясь  
        плыви. Спасет Аллах достойного любви. ");  
        char ch,ch1 ;
```

```

        int bj=0, dl=0, nu;
        for(int i=0; i< s.length(); i++){
            dl=0; nu=i;
            /*=====проходимо по s до кінця слова charAt(i)
            - звернення до i-го символу та перетворення його в
            тип char */
            while(!(s.charAt(i)=='
            '||s.charAt(i)=='.'||s.charAt(i)==' ','')){
                i++; dl++;
            }
            /*=====зберігаємо в ch останню букву слова
            s.charAt(i-1), а в ch1 розділовий символ після
            слова s.charAt(i) */
            ch=s.charAt(i-1); ch1=s.charAt(i);
            /*=====перепишуємо слово з s в buf пропускаючи букви
            рівні останій букві */
            for(int j=nu; j<i; j++){
                if (ch!=s.charAt(j)){
                    buf[bj]= s.charAt(j);
                    bj++;
                }
            }
            /*=====дописуємо buf останню букву та пробел */
            buf[bj]= ch; bj++;
            if(ch1==' '){buf[bj]= ch1;    bj++;
            }
        }
        /*=====виводим заповнену частину buf посимвольно */
        System.out.print("\n rez buf");
        for(int i=0; i<bj; i++){
            System.out.print(""+buf[i]);
        }
        /*===формуем рядок buf3 з заповненої частини buf (від
        0 символа, bj символів)*/
        String buf3= new String(buf, 0, bj);
        System.out.print("\n rez buf3"+buf3);
    }
}

```

Результат

Input : Input : Mas

Любовь, как море - не боясь пльви. Спасет Аллах
достойного любви.

rez buf2

rez bufЛюбовь, ак море – не боясь пльви. Спасет Аллах
дстйнго любви.

rez buf3Любовь, ак море – не боясь пльви. Спасет
Аллах дстйнго любви.

Завдання

Рівень 1

1 Визначити, чи є заданий рядок паліндромом (приклад паліндрома: «А роза упала на лапу Азора»).

2 Дано текст, який містить лише цифри і літери. Визначити, скільки слів містить більше 6 символів.

3 Дано текст. Надрукувати його, замінивши в ньому всі літери а на літеру о.

4 Дано текст. Визначити найдовше слово.

5 Дано текст. Визначити кількість слів, які починаються з літери а.

Рівень 2. Варіанти завдань на рядкові буфери

1 Дано текст. Замінити в рядках всі входження слова «cat» словом «dog».

2 Значенням рядка служить слово з дефісом. Поміняти місцями частини слова до і після дефіса.

3 Дано кілька рядків. Видалити з рядків всі знаки "-", які містяться в них.

4 Дано текст. Надрукувати його, видаливши з нього всі літери b, безпосередньо перед якими стоїть літера c.

5 Дано текст. Надрукувати його, видаливши всі подвоєні літери.

6 Дано текст. Видалити всі входження th.

7 Дано текст. Надрукувати його, видаливши повторні входження слів.

8 Дано текст. Перетворити слова за таким правилом: перенести першу літеру в кінець слова.

9 Дано текст. Перетворити слова за таким правилом: перенести останню літеру в початок слова.

10 Дано текст. Перетворити слова за таким правилом: видалити першу літеру слова.

ЗАВДАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ

1 Задано масив цілих чисел. Написати функцію, що визначає кількість парних і непарних чисел у ньому і повертає істину, якщо парних більше.

2 Задано масив цілих чисел. Визначити, яке число зустрічається в ньому найчастіше.

3 Задано масив цілих чисел. Надрукувати ті з них, які входять у масив по одному разу.

4 Задано масив значень $A = \{a_i\}$ з 8 елементів і змінні m і k . Знайти кількість елементів масиву, більших, ніж k ; кількість елементів масиву, більших, ніж $2 * m$, знайти індекс мінімального елемента.

5 Задано масив значень $A = \{a_i\}$ з 12 елементів. Знайти по два максимальних і мінімальних елементи.

6 Задано масив значень $A = \{a_i\}$ з 10 елементів. Знайти максимальний і мінімальний елементи, поміняти їх місцями.

7 Задано масив значень $A = \{a_i\}$ з 12 елементів. Знайти суму всіх додатних елементів і добуток від'ємних.

8 Задано масив значень $A = \{a_i\}$ з 12 елементів. Знайти суму абсолютних значень відхилень елементів від середнього значення.

9 Задано масив значень $A = \{a_i\}$ з 8 елементів. Додати елемент на позицію з індексом $k = 4$.

10 Задано масив значень $A = \{a_i\}$ з 10 елементів. Знайти максимальний і мінімальний елементи і видалити їх.

11 Масив значень $A = \{a_i\}$ складається з літер латинського алфавіту. Ввести індексний масив B . Вивести елементи основного масиву в порядку, заданому в масиві B .

12 Задано масив значень $A = \{a_i\}$ з 10 елементів. Знайти по два максимальних і мінімальних елементи і підрахувати середнє значення інших елементів.

13 Задано масив значень $A = \{a_i\}$ з 10 елементів. Знайти максимальний елемент і замінити всі елементи масиву різницею між знайденим максимальним значенням і поточним.

14 Задано масив значень $A = \{a_i\}$ з 12 елементів. Знайти середнє значення елементів, які містяться на парних позиціях.

15 Задано масив значень $A = \{a_i\}$ з 12 елементів. Знайти середнє значення елементів, які містяться на позиціях, кратних 3.

ЗАПИТАННЯ ДЛЯ САМОСТІЙНОЇ ПІДГОТОВКИ ДО МОДУЛЬНОГО КОНТРОЛЮ

1 Який мінімально необхідний набір утиліт існує для початку роботи з Java?

2 Починаючи з якої версії мови Java підтримується використання Javadoc-коментарів?

3 Починаючи з якої версії мови Java підтримується Unicode 6.0?

4 Дайте характеристику пакета Java Standart Edition.

5 Дайте характеристику пакета Java Enterprise Edition.

6 Дайте характеристику пакета Java Micro Edition.

7 Дайте характеристику пакета JavaCard.

8 Який пакет є інтегрованим середовищем розробки з використанням графічного інтерфейсу?

9 Наведіть особливість байт-кодів.

10 Назвіть особливість вихідного коду Java-програми.

11 Який тип даних Java є базовим?

12 Який тип даних Java є типом посилання?

13 Який цілий тип даних Java має довжину 8 бітів?

14 Який цілий тип даних Java має довжину 32 біта?

15 Яких значень набудуть змінні після виконання фрагменту програми:

```
<br>int x,y,z,w; <br> x=0;y=0; <br>z=x++; w=++y;
```

x=1, z=0, y=1, w=1

x=0, z=0, y=1, w=1

x=1, z=0, y=0, w=1

x=1, z=0, y=1, w=0

x=1, z=1, y=1, w=1

16 Розглянемо фрагмент програми

```
<br>int i=5; <br>double d=7.00; <br>
```

Як коректно виконати присвоювання в змінну i значення змінної d?

V1 i=d;

V2 i=(int)d;

V3 i=(float)d;

V4 i=(double)d;

V5 Правильної відповіді не наведено.

17 Який оператор містить оголошення масиву та ініціалізацію його елементів?

V1 `int fib[]= {1,1,2,3,5};`

V2 `IQ = new int[500];`

V3 `IQ[4] = 127;`

V4 `int [][] dim = new int[3][8];`

V5 Правильної відповіді не наведено.

18 Який оператор містить умовну (тернарну) операцію?

V1 `double d=7.00;`

V2 `x = (int)(x/y)*y;`

V3 `static final double pi=3.14159;`

V4 `max=(a<=b) ? b : a;`

V5 `int fib[]= {1,1,2,3,5}.`

19 Який оператор містить явне перетворення типів?

V1 `double d=7.00;`

V2 `x = (int)(x/y)*y;`

V3 `static final double pi=3.14159;`

V4 `max=(a<=b) ? b : a;`

V5 `int fib[]= {1,1,2,3,5}.`

20 Яким буде значення змінної `y` в результаті виконання фрагменту програми?

```
<br>float y, x=5; <br>if (x<2) y = x-2; <br>else y = 4*x;
```

V1 7;

V2 20;

V3 10;

V4 5;

V5 3.

21 Наведіть характеристику оператора циклу `while`.

22 Наведіть характеристику оператора циклу `do`.

23 Наведіть характеристику оператора циклу `for`.

24 Скільки разів буде виконано цикл?

```
<br>int i=2, j=5; <br>while (i<j) <br>{<br>i++;<br>}
```


- V1 3;
- V2 2;
- V3 5;
- V4 0;
- V5 7.

25 Що станеться після компіляції коду і запуску програми без аргументів?

```
<br>public static void main(String[] args) {<br>  
    System.out.println(args.length); <br> }
```

- V1 код не скомпілюється;
- V2 код скомпілюється, але під час виконання буде викинуто виключення `NullPointerException`;
- V3 код скомпілюється, а на консоль буде виведено число "0";
- V4 код скомпілюється, але нічого не виведе на консоль;
- V5 правильної відповіді не наведено.

26 Скільки разів буде виконано цикл?

```
<br>int top, bot; <br>for(top=0, bot=10; top<=bot; top++, bot--)<br>  
{<br>оператори<br>}
```

- V1 4;
- V2 5;
- V3 6;
- V4 10;
- V5 0.

27 Що буде виведено на екран у результаті виконання програми?

```
<br>int i=2; <br>switch (i) {<br>case 1: System.out.println ("one");  
break; <br>case 2: System.out.println ("two");<br>break; <br>  
case default: System.out.println("default");<br>break; <br>}
```

- V1 one;
- V2 two;
- V3 default;
- V4 two default;
- V5 правильної відповіді не наведено.

28 Що буде виведено на екран у результаті виконання програми, якщо запустити її командою "java Test 1 two 3"?

```
<br>public class Test { <br> static public void main(String[] args) {  
<br> for (int i=0;i<3;i++) { <br> int k = Integer.parseInt(args[i]);  
<br> System.out.println(k); <br>}<br>}
```

V1 1;

V2 1 two;

V3 1 two 3;

V4 повідомлення про помилку;

V5 код не скомпілюється.

Q29 Виберіть варіант коментарю в Java-програмі, який не приведе до помилки:

V1 // This is a valid comment in java

V2 <!-- This is a valid comment in java -->

V3 /* /* This is a valid comment in java. */ */

V4 || This is a valid comment in java

V5 правильної відповіді не наведено.

Q30 Виберіть варіант коментарю в Java-програмі, який приведе до помилки:

V1 /** This is a valid comment in java. */

V2 <!-- This is a valid comment in java -->

V3 /* This is a valid comment in java. */

V4 /* This is a valid comment in java. **/

V5 // This is a valid comment in java.

Q31 Що буде виведено на екран у результаті виконання програми?

```
<br>int i = 0; <br>i++; <br>System.out.print(i); <br>i = i++; <br>System.out.println(i);
```

V1 12;

V2 01;

V3 11;

V4 10;

V5 21.

Q32 Якими будуть значення змінних після виконання фрагмента програми?

```
<br>int i = 1; <br>int j = 1; <br>j = i++;
```

V1 j=1, i = 1;

V2 j=1, i = 2;

V3 j=2, i = 2;

V4 j=2, i = 1;

V5 правильної відповіді не наведено.

Q33 Що станеться під час компіляції і виконання цього фрагмента коду?

```
<br>int i = 0,j =5;<br>tp: for (;;) {<br>i++;<br>for(;;){<br>if(i--<br>j){<br>break tp;<br>}<br>}<br>System.out.println("i="+i+",j="+j);<br>}
```

V1 буде виведено "i = 1, j = 0";

V2 буде виведено "i = 1, j = 4";

V3 буде виведено "i = 3, j = 4";

V4 буде виведено "i = 3, j = 0";

V5 помилка компіляції.

Q34 Який рядок або рядки коду містять помилку?

```
<br>int i = 1;//1 <br>i = -(10 + 2 + i);//2 <br>++i--;//3 <br>System.out.println(i);
```

V1 1;

V2 2;

V3 3;

V4 2,3;

V5 код не містить помилок.

35 Що станеться в результаті виконання нижченаведеного коду?

```
<br>byte b=0; <br>while (++b>0); <br>System.out.println(b);
```

V1 виведеться 255;

V2 виведеться -128;

V3 виведеться 0;

V4 виведеться 127;

V5 відбудеться зациклення програми.

36 Що станеться в результаті виконання нижченаведеного коду?

```
<br>byte b=0; <br>while (++b>0); <br>System.out.println(b);
```

V1 помилка компіляції;

V2 виведеться -128;

V3 відбудеться зациклення програми;

V4 виведеться 127;

V5 правильної відповіді не наведено.

37 Який буде результат компіляції і запуску нижченаведеної програми?

```
<br>int i=1; <br>long j=1; <br>if (i==1) { <br>int k=1; <br>if (i==k){ <br>System.out.println("i equals k"); <br>} <br>} <br>else if (j==k) { <br>System.out.println("j equals k"); <br>} <br>} <br>}
```

V1 помилка компіляції;

V2 i equals k;

V3 j equals k;

V4 програма скомпілюється, але виникне помилка часу виконання;

V5 правильної відповіді не наведено.

38 Яке ключове слово є специфікатором доступу до методу Java?

V1 friendly;

V2 static;

V3 abstract;

V4 synchronized;

V5 правильної відповіді не наведено.

39 Яке ключове слово є модифікатором методу Java?

V1 protected;

V2 friendly;

V3 abstract;

V4 private;

V5 правильної відповіді не наведено.

40 Який специфікатор доступу до методу визначає його доступним для будь-яких класів?

V1 protected;
V2 friendly;
V3 public;
V4 private protected;
V5 private.

41 Який специфікатор доступу до методу визначає його доступним для класів, які містяться в тому ж пакеті?

V1 protected;
V2 friendly;
V3 public;
V4 private protected;
V5 private.

СПИСОК ЛІТЕРАТУРИ

1 Development of method of definition maximum clique in a non-oriented graph / S. V. Listrovoy, V. M. Butenko, V. O. Bryksin, O. V. Golovko // EasternEuropean Journal of Enterprise Technologies. – 2017. – Vol. 5, № 4 (89). – P. 12 – 17.

2 Formulation of the Problem of Maximum Clique Determination in Non-Oriented Graphs / S. V. Listrovoy, O. V. Golovko, V. M. Butenko, M. V. Ushakov // International Journal of Engineering & Technology Vol 7 No 4.3 (2018): Special Issue 3 PP. 293 – 297. DOI: 10.14419/ijet.v7i4.3.19807.

3 Пат. UA № 116449, МПК⁹ H03K 17/60 (2006.01). Комутаційний пристрій – оптоелектронний аналог електромагнітного реле струму / [Текст] Бутенко В. М., Головка О. В., Зайченко О. Б., Мелешко В. В., Мірошник М. А., Мойсеєнко В. І., Чуб І. М., Чуб С. Г.; заявник і власник Укр. держ. ун-т залізнич. трансп. – № u 2016 11255; заявл. 07.11.2016; опубл. 25.05.2017, Бюл. № 10. – 8 с.

4 Signal flow graph models and alternative gain formula for multiprobe microwave multimeter / O. B. Zaichenko, V. M. Butenko, M. A. Miroshnyk // Інформаційно-керуючі системи на залізничному транспорті. – 2016. – № 12. – С. 12 – 17.

5 Listrovoy S. V., Butenko V. M. Algorithm of Sub Exponential Complexity for the SAT // International Journal of Computer and Information Technology (ISSN: 2279-0764) Volume 02 – Issue 05, September 2013. – P.837 – 842.

6 Математичне моделювання в розподілених інформаційно-керуючих системах залізничного транспорту [Текст] : монографія / С. В. Лістровий, С. В. Панченко, В. І. Мойсеєнко, В. М. Бутенко. – Харків: ФОП Бровін О. В., 2017. – 220 с.

7 Меркулов, В. С. Основи алгоритмізації базових обчислювальних процесів [Текст] : навч. посібник/ В. С. Меркулов, В. М. Бутенко. – Харків : УкрДАЗТ, 2008. – 163 с.

8 Завдання і методичні вказівки до розрахунково-графічної та контрольної робіт з дисциплін «Програмування» та «Інформатика» для студентів факультету АТЗ [Текст] / В. М. Бутенко, О. В. Головка, М. О. Колісник, С. О. Бантюкова. – Харків : УкрДАЗТ, 2016. – 74 с.

9 Нарожний, В. В. Цифрові електронно-обчислювальні машини [Текст] : конспект лекцій/ В. В. Нарожний. – Харків: УкрДАЗТ, 2010. – 105 с.

10 Павленко, Є. П. Інженерія програмного забезпечення. WEB-програмування [Текст] : навч. посібник / Є. П. Павленко, В. М. Бутенко, О. В. Головка. — Харків : УкрДУЗТ, 2019. – 139 с.

11 Оптимізація алгоритму субекспоненціальної складності для вирішення SAT задачі [Текст] / А. Б. Бойнік, В. М. Бутенко, О. В. Головка, М. В. Ушаков //Інформаційно-керуючі системи на залізничному транспорті. – 2018. – № 3 (107). – С. 12 – 17.

12 Мойсеєнко, В. І. Нові процедури обслуговування інформаційно-керуючих систем та контроль фактичного виконання роботи [Текст] / В. І. Мойсеєнко, В. М. Бутенко, О. В. Головка // 31-ша Міжнар. наук.-практ. конф. «Інформаційно-керуючі системи на залізничному транспорті». – 2018. – № 4. – С. 59 – 60.

13 Бутенко, В. М. Методичний посібник до лабораторних робіт з дисципліни «Математичні методи та моделі розрахунку на ЕОМ» [Текст] / В. М. Бутенко, О. Б. Болотов, В. М. Шумєєв. – Харків : УкрДАЗТ, 2006. – Ч. 1. – 28 с.

14 Бутенко, В. М. Конспект лекцій з дисципліни «Інформаційні системи на залізничному транспорті» [Текст] / В. М. Бутенко, С. Є. Бантюков, В. Г. Пчолін. – Харків : УкрДАЗТ, 2008. – Ч. 1. – 28 с.

15 Основи програмування мовами високого рівня [Текст] : навч. посібник / В. М. Бутенко, В. С. Меркулов, О. В. Чаленко, О. В. Казанко. – Харків: УкрДАЗТ, 2009. – 206 с.

16 Математичні методи та моделі в розрахунках на ЕОМ [Текст] : навч. посібник / М. І. Данько, В. С. Меркулов, В. О. Гончаров; за заг. ред. М. І. Данька. — Харків : УкрДАЗТ, 2008. – 172 с.