**Лабораторна робота №2**

**Програмування алгоритмів лінійної структури в Java. Потоки вводу-виводу.**

**Теоретичні відомості**.

1. **Найпростіша програма на Java.**

Мова програмування Java повністю ґрунтується на об’єктно-орієнтованому підході (об’єктно-орієнтованій парадигмі програмування). В об’єктно-орієнтованому програмуванні (ООП) програми утворюються з сукупності об’єктів, що взаємодіють між собою. Об’єкт є іменованою моделлю реальної сутності, що володіє конкретними значеннями своїх властивостей та визначеною поведінкою. Набір властивостей та поведінка об’єкта описуються в класі, на основі якого створюється об’єкт. З іншого боку, клас об’єднує у собі набір даних та методи їх обробки, а об’єкт є представником класу, який володіє конкретними значеннями описаних у класі даних. Навіть найпростіша програма на Java повинна складатися принаймні з одного класу. Наприклад, програма, що виводить на екран текст «Hello, world!», може мати вигляд:

**public class** HelloWorld {

**public static void** main(String[]args) {

System.***out***.println("Hello, world!");

}

}

7/

Ускладнимо клас HelloWorld, змінивши задачу наступним чином:

Написати програму, що запитує в користувача його ім’я, а потім виводить привітання, вставивши в нього введене ім’я. При такій постановці задачі програма повинна реалізувати ввід та запам’ятовування даних, для цього потрібно використовувати змінні. Для опрацювання текстових даних таких як ім’я в Java використовується тип String. Описувати змінну необхідно перед її використанням, але це можна в будь-якому місці програми. Дозволяється, зокрема, описувати змінну безпосередньо перед присвоєнням їй значення. Наприклад, якщо вважати що нашу програму завжди використовуватиме користувач, якого звуть Зеник, код класу HelloWorld, з використанням текстової змінної можна трансформувати так:

**public class** HelloWorld {

**public static void** main(String[]args) {

String userName = "Зеник";

System.***out***.println("Привіт, " + userName + "!");

}

}

Знак «+» в команді виводу використовується для з’єднання (конкатенації) частин тексту.

1. **Потоки вводу – виводу в Java**

Для вводу даних використовується клас Scanner з бібліотеки пакетів Java.

Цей клас потрібно імпортувати в тій програмі, де він буде використовуватися. Це робиться до початку відкриття класу в коді програми. У класі є методи для читання чергового символу заданого типу зі стандартного потоку введення, а також для перевірки існування такого символу.

Для роботи зі потоком введення необхідно створити об’єкт класу Scanner, при створенні вказавши, з яким потоком введення він буде пов’язаний. Стандартний потік введення (клавіатура) в Java представлений об’єктом System.in. А стандартний потік виводу (дисплей) – вже знайомим вам об’єктом System.out. Є ще стандартний потік для виведення помилок System.err.

import java.util.Scanner; // імпортуємо клас

public class Main {

public static void main(String[] args) {

// створюємо об’єкт класу Scanner

Scanner sc = new Scanner(System.in);

System.out.print("Введіть ціле число: ");

if(sc.hasNextInt())

{

/\*

\*повертає істину, якщо з потоку можна зчитати

\*ціле число

\*зчитуємо ціле число з потоку вводу

\*і зберігаємо в змінну

\*/

int i = sc.nextInt();

System.out.println(i\*2);

}

else

{

System.out.println("Введене не ціле число");

}

}

}

Метод hasNextDouble(), що застосовується до об’єкту класу Scanner, перевіряє, чи можна зчитати з потоку вводу дійсне число типу double, а метод nextDouble() — зчитує його. Якщо спробувати зчитати значення без попередньої перевірки, то під час виконання програми можна отримати помилку (дебагер таку помилку не знайде).

Наприклад, розглянемо програму, яка дозволяє опрацьовувати дійсні числа, але видає помилку, безпосередньо в процесі виконання, при введенні символьної величини. У процесі відлагодження помилка виявлена не буде (переконайтеся в цьому).

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

double  i = sc.nextDouble();

// якщо ввести букву, то матимемо помилку

// при виконанні

System.out.println(i/3);

}

}

Існує метод nextLine(), який зчитує цілу послідовність символів, тобто стрічку, тому значення, отримане через цей метод потрібно зберігати в об’єкті класу String. В наступному прикладі створимо два таких об’єкти і виведемо їх конкатенацію:

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        String s1, s2;

        s1 = sc.nextLine();

        s2 = sc.nextLine();

        System.out.println(s1 + s2);

    }

}

Є також метод hasNext(), який перевіряє, чи залишилися в потоці вводу якісь символи.

Для виведення інформації зручно використовувати наступні керуючі Escape- послідовності:

|  |  |
| --- | --- |
| \b | backspace BS - забій |
| \t | tab HT - табуляція |
| \n | linefeed LF - кінець рядка |
| \f | form feed FF - кінець сторінки |
| \r | carriage return CR -повернення каретки |
| \” | double quote " - подвійна лапка |
| \’ | single quote ' - одинарна лапка |
| \\ | backslash \ - зворотна коса риска |

Для виконання математичних обчислень в Java використовують спеціальний клас Math, методами якого є математичні функції та операції. Повний перелік методів класу Math, а також опис їх дії наведено у Додатку 2.

Крім того, Java дозволяє використання традиційних математичних констант:

|  |  |  |
| --- | --- | --- |
| **Modifier and Type** | **Constant Field** | **Value** |
| public static final double | [E](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#E) | 2.718281828459045 |
| public static final double | [PI](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#PI) | 3.141592653589793 |

З більш докладним описом дії методів, обмеженнями на їх використання та роботи з константами можна ознайомитися тут:

<https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html>

**Завдання для роботи**

1. Опрацюйте теоретичні відомості та створіть проект.
2. Створіть, скомпілюйте та запустіть просту Java-програму, що запитує в користувача ім’я, вітається з ним, питає його вік та виводить рік народження.
3. Вважаючи усі вхідні і вихідні дані дійсними числами типу **float**, написати програму для розв’язання задачі згідно варіанту за Додатком 1.
4. Запишіть до звіту коди програм. Додайте скріни отриманих результатів.
5. Дайте відповіді на контрольні запитання.

**Додаток 1**

**Варіанти завдань**

1. Дано два кола із загальним центром і радіусами R1 і R2 (R1> R2). Знайти площі цих кіл S1 та S2, а також площу S3 кільця, зовнішній радіус якого дорівнює R1, а внутрішній радіус дорівнює R2. Як значення π використовувати 3.14.
2. Дано координати двох протилежних вершин прямокутника: (x1, y1), (x2, y2). Сторони прямокутника паралельні осям координат. Знайти периметр і площу даного прямокутника.

3. Дано сторони прямокутника a і b. Знайти його площу S і периметр P.

4. Дано довжини ребер A, B, C прямокутного паралелепіпеда. Знайти його об’єм та площу поверхні.

5. Дано число A. Обчислити A15, використовуючи дві допоміжні змінні і п'ять операцій множення. Для цього послідовно знаходити A2, A3, A5, A10, A15. Вивести всі знайдені степені числа A.

6. Дано значення температури T в градусах Фаренгейта. Визначити значення цієї ж температури в градусах Цельсія. Температура по Цельсію TC і температура за TF Фаренгейтом TF зв'язані наступними співвідношенням: TC = (TF - 32) • 5/9.

7. Дано значення температури T в градусах Цельсія. Визначити значення цієї ж температури в градусах Фаренгейта. Температура за Цельсієм TC і температура за Фаренгейтом TF пов'язані наступним співвідношенням: TC = (TF - 32) • 5/9.

8. Відомо, що X кг шоколадних цукерок коштує A гривень, а Y кг ірисок - B гривень. Визначити, скільки коштує 1 кг шоколадних цукерок, 1 кг ірисок, а також у скільки разів шоколадні цукерки дорожчі ірисок.

9. Дано два ненульових числа. Знайти суму, різницю, добуток і частку їх квадратів.

10. Швидкість першого автомобіля V1 км / год, другого - V2 км / год (V1 > V2), відстань між ними S км. Визначити відстань між ними через T годин, якщо рухаються в одному напрямку.

11. Швидкість першого автомобіля V1 км / год, другого - V2 км / год, відстань між ними S км. Визначити відстань між ними через T годин, якщо автомобілі рухаються назустріч один одному.

1. Пішохід пройшов S1 км за t1 годин і S2 км за t2 годин. Вивести загальний час руху пішохода та його середню швидкість.
2. Змішали m1 грамів розчину, що містить p1 % солі з m2 грамами розчину, що містить p2 % солі. Вивести загальну масу розчину та відсотковий вміст солі в ньому.
3. Човен проходить відстань S за течією річки і назад. Швидкість човна V км/год, швидкість течії U км/год. Знайти час витрачений човном на дорогу та його середню швидкість.
4. Перша труба наповнює резервуар за t1 хвилин, друга – за t2 хвилин. За скільки хвилин обидві труби наповнять резервуар. Яка частина резрвуара при цьому буде заповнена першою трубою.
5. Дано три сторони трикутника: А, В, С. Визначити його периметр та площу.
6. Задано три сторони трикутника: А, В, С. Визначити довжини всіх його медіан.
7. Трикутник задано координатами його вершин: А(x1; y1), B(x2; y2), C(x3; y3). Знайти довжини сторін трикутника та його периметр.
8. Задано радіус кулі. Визначити її об’єм та площу поверхні.
9. Трикутник задано трьома сторонами. Знайти величини кутів трикутника.
10. Куплено m1 метрів штор за ціною с1 гривень та m2 метрів тюлю за ціною с2 гривень. Визначити на скільки більше грошей заплатили за штори, ніж за тюль.
11. Задано два вектори своїми координатами: ( x1, y1, z1) та (x2, y2, z2). Визначити скалярний добуток цих векторів.
12. Задано два вектори своїми координатами: ( x1, y1, z1) та (x2, y2, z2). Визначити векторний добуток цих векторів.
13. Задано три вектори своїми координатами: ( x1, y1, z1), (x2, y2, z2) та (x3, y3, z3). Визначити мішаний добуток цих векторів.
14. Задано кількість годин, хвилин та секунд початку та кінця деякої роботи. Визначити загальну тривалість роботи.
15. В одному магазині придбали m1 кг деякого товару за ціною p1 грн за кілограм, в іншому – ще m2 кг цього ж товару але за ціною p2 грн за кілограм. Знайти загальну вартість покупки. Якою вийшла середня ціна товару.

**Зразок виконання (варіант 26).**

Створюємо новий проект, в ньому клас з методом main, де записуємо відповідний код програми. Орієнтовний код класу:

**import** java.io.BufferedReader; **import** java.io.IOException; **import** java.io.InputStreamReader;

**public class** Lab1 {

**public static void** main(String[]args) **throws** IOException { BufferedReader reader = **new** BufferedReader(

**new** InputStreamReader(System.***in***));

System.***out***.println("Ціна товару в 1-му магазині:");

**float** p1= Float.*parseFloat*(reader.readLine());

System.***out***.println("Маса придбаного товару:");

**float** m1= Float.*parseFloat*(reader.readLine());

System.***out***.println("Ціна товару в 2-му магазині:");

**float** p2= Float.*parseFloat*(reader.readLine());

System.***out***.println("Маса придбаного товару:");

**float** m2= Float.*parseFloat*(reader.readLine());

**float** s=p1\*m1+p2\*m2;

**float** p=s/(m1+m2);

System.***out***.println("Вартість придбаного товару " + s + "грн.");

System.***out***.println("Середня ціна за кілограм - " + p + "грн.");

}

}

**Додаток 2**

|  |  |
| --- | --- |
| **Modifier and Type** | **Method and Description** |
| static double | [**abs**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#abs%28double%29)(double a)  Returns the absolute value of a double value. |
| static float | [**abs**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#abs%28float%29)(float a)  Returns the absolute value of a float value. |
| static int | [**abs**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#abs%28int%29)(int a)  Returns the absolute value of an int value. |
| static long | [**abs**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#abs%28long%29)(long a)  Returns the absolute value of a long value. |
| static double | [**acos**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#acos%28double%29)(double a)  Returns the arc cosine of a value; the returned angle is in the range 0.0 through *pi*. |
| static double | [**asin**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#asin%28double%29)(double a)  Returns the arc sine of a value; the returned angle is in the range -*pi*/2 through *pi*/2. |
| static double | [**atan**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#atan%28double%29)(double a)  Returns the arc tangent of a value; the returned angle is in the range -*pi*/2 through *pi*/2. |
| static double | [**atan2**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#atan2%28double,%20double%29)(double y, double x)  Returns the angle *theta* from the conversion of rectangular coordinates (x, y) to polar coordinates (r, *theta*). |
| static double | [**cbrt**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#cbrt%28double%29)(double a)  Returns the cube root of a double value. |
| static double | [**ceil**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#ceil%28double%29)(double a)  Returns the smallest (closest to negative infinity) double value that is greater than or equal to the argument and is equal to a mathematical integer. |
| static double | [**copySign**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#copySign%28double,%20double%29)(double magnitude, double sign)  Returns the first floating-point argument with the sign of the second floating-point argument. |
| static float | [**copySign**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#copySign%28float,%20float%29)(float magnitude, float sign)  Returns the first floating-point argument with the sign of the second floating-point argument. |
| static double | [**cos**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#cos%28double%29)(double a)  Returns the trigonometric cosine of an angle. |
| static double | [**cosh**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#cosh%28double%29)(double x)  Returns the hyperbolic cosine of a double value. |
| static double | [**exp**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#exp%28double%29)(double a)  Returns Euler's number *e* raised to the power of a double value. |
| static double | [**expm1**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#expm1%28double%29)(double x)  Returns *e*x -1. |
| static double | [**floor**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#floor%28double%29)(double a)  Returns the largest (closest to positive infinity) double value that is less than or equal to the argument and is equal to a mathematical integer. |
| static int | [**getExponent**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#getExponent%28double%29)(double d)  Returns the unbiased exponent used in the representation of a double. |
| static int | [**getExponent**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#getExponent%28float%29)(float f)  Returns the unbiased exponent used in the representation of a float. |
| static double | [**hypot**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#hypot%28double,%20double%29)(double x, double y)  Returns sqrt(*x*2 +*y*2) without intermediate overflow or underflow. |
| static double | [**IEEEremainder**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#IEEEremainder%28double,%20double%29)(double f1, double f2)  Computes the remainder operation on two arguments as prescribed by the IEEE 754 standard. |
| static double | [**log**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#log%28double%29)(double a)  Returns the natural logarithm (base *e*) of a double value. |
| static double | [**log10**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#log10%28double%29)(double a)  Returns the base 10 logarithm of a double value. |
| static double | [**log1p**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#log1p%28double%29)(double x)  Returns the natural logarithm of the sum of the argument and 1. |
| static double | [**max**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#max%28double,%20double%29)(double a, double b)  Returns the greater of two double values. |
| static float | [**max**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#max%28float,%20float%29)(float a, float b)  Returns the greater of two float values. |
| static int | [**max**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#max%28int,%20int%29)(int a, int b)  Returns the greater of two int values. |
| static long | [**max**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#max%28long,%20long%29)(long a, long b)  Returns the greater of two long values. |
| static double | [**min**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#min%28double,%20double%29)(double a, double b)  Returns the smaller of two double values. |
| static float | [**min**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#min%28float,%20float%29)(float a, float b)  Returns the smaller of two float values. |
| static int | [**min**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#min%28int,%20int%29)(int a, int b)  Returns the smaller of two int values. |
| static long | [**min**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#min%28long,%20long%29)(long a, long b)  Returns the smaller of two long values. |
| static double | [**nextAfter**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#nextAfter%28double,%20double%29)(double start, double direction)  Returns the floating-point number adjacent to the first argument in the direction of the second argument. |
| static float | [**nextAfter**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#nextAfter%28float,%20double%29)(float start, double direction)  Returns the floating-point number adjacent to the first argument in the direction of the second argument. |
| static double | [**nextUp**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#nextUp%28double%29)(double d)  Returns the floating-point value adjacent to d in the direction of positive infinity. |
| static float | [**nextUp**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#nextUp%28float%29)(float f)  Returns the floating-point value adjacent to f in the direction of positive infinity. |
| static double | [**pow**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#pow%28double,%20double%29)(double a, double b)  Returns the value of the first argument raised to the power of the second argument. |
| static double | [**random**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#random%28%29)()  Returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0. |
| static double | [**rint**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#rint%28double%29)(double a)  Returns the double value that is closest in value to the argument and is equal to a mathematical integer. |
| static long | [**round**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#round%28double%29)(double a)  Returns the closest long to the argument, with ties rounding up. |
| static int | [**round**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#round%28float%29)(float a)  Returns the closest int to the argument, with ties rounding up. |
| static double | [**scalb**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#scalb%28double,%20int%29)(double d, int scaleFactor)  Return d × 2scaleFactor rounded as if performed by a single correctly rounded floating-point multiply to a member of the double value set. |
| static float | [**scalb**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#scalb%28float,%20int%29)(float f, int scaleFactor)  Return f × 2scaleFactor rounded as if performed by a single correctly rounded floating-point multiply to a member of the float value set. |
| static double | [**signum**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#signum%28double%29)(double d)  Returns the signum function of the argument; zero if the argument is zero, 1.0 if the argument is greater than zero, -1.0 if the argument is less than zero. |
| static float | [**signum**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#signum%28float%29)(float f)  Returns the signum function of the argument; zero if the argument is zero, 1.0f if the argument is greater than zero, -1.0f if the argument is less than zero. |
| static double | [**sin**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#sin%28double%29)(double a)  Returns the trigonometric sine of an angle. |
| static double | [**sinh**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#sinh%28double%29)(double x)  Returns the hyperbolic sine of a double value. |
| static double | [**sqrt**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#sqrt%28double%29)(double a)  Returns the correctly rounded positive square root of a double value. |
| static double | [**tan**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#tan%28double%29)(double a)  Returns the trigonometric tangent of an angle. |
| static double | [**tanh**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#tanh%28double%29)(double x)  Returns the hyperbolic tangent of a double value. |
| static double | [**toDegrees**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#toDegrees%28double%29)(double angrad)  Converts an angle measured in radians to an approximately equivalent angle measured in degrees. |
| static double | [**toRadians**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#toRadians%28double%29)(double angdeg)  Converts an angle measured in degrees to an approximately equivalent angle measured in radians. |
| static double | [**ulp**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#ulp%28double%29)(double d)  Returns the size of an ulp of the argument. |
| static float | [**ulp**](https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#ulp%28float%29)(float f)  Returns the size of an ulp of the argument. |

**Контрольні запитання**:

1. Який клас використовується в Java для вводу даних?
2. Яким об’єктом представлено стандартний потік введення даних?
3. Яким об’єктом представлено стандартний потік виведення даних?
4. Яким об’єктом представлено стандартний потік виведення помилок?
5. Опишіть метод hasNextDouble().
6. Опишіть метод nextDouble().
7. Опишіть метод nextLine().
8. Опишіть метод hasNext().
9. Які керуючі Escape – послідовності в Java ви знаєте? Запишіть їх, поясніть дію.
10. Які математичні константи можна використовувати в Java. Якого вони типу?
11. Опишіть наступні методи класу Math: abs(), acos(), asin(), atan(), cbrt(), ceil(), cos(), exp(), floor(), hypot(), log(), log10(), max(), min(), pow(), random(), round(), signum(), sin(), sqrt(), tan(), toDegrees(), toRadians().