

# **Лабораторна робота № 1**

## **з дисципліни «Програмування алгоритмічних структур»**

### **Тема «Програмування лінійних алгоритмів»**

**Мета роботи:** набуття практичних навичок використання типів даних, математичних операцій та форматowanego виводу значень змінних під час створення програмних проектів на мові Java.

Час виконання – 4 год.

#### **Завдання на лабораторну роботу.**

1. Написати програму мовою Java для обчислення значень змінних, що зазначені у таблиці 1, за заданими розрахунковими формулами і наборами вхідних даних. Вивести на екран дату і час в заданому форматі. Варіант обрати за списком групи.

2. Розробити програмний проект в середовищі IntelliJ IDEA (або в текстовому редакторі Sublime Text) для реалізації написаної програми. Перевірити результати обчислень альтернативними розрахунками (наприклад, Excel, калькулятор, тощо).

3. Зробити Звіт з лабораторної роботи та вчасно надіслати викладачу на перевірку.

#### **Вимоги до розв'язання завдання (табл. 1).**

Завдання запрограмувати в одному класі, який має назву Lr1. В даному класі організувати такі методи: main, calcR, calcS, outputDate, де

calcR – метод для обчислення змінної R;

calcS – метод для обчислення змінної S;

outputDate – метод для виведення дати та часу в заданому форматі.

В методі main організувати:

- введення початкових даних для завдання 1 (табл. 1). Введення даних зробити за допомогою присвоєння значень змінним;
- виклик методів calcR, calcS;
- отримання із методів значень змінних R та S;
- виведення вхідних даних та результатів розрахунку змінних R та S. Виведення організувати за допомогою метода System.out.printf форматowanego виводу значень. Для дробових чисел – не більше 4 знаків після коми. Вивід супроводжувати найменуваннями виведених змінних;
- ввести початкові дані за допомогою Scanner (з клавіатури) та ще раз викликати методи calcR, calcS. Отримати значення R та S. Вивести вхідні дані та значення R та S в консоль за допомогою метода System.out.printf форматowanego виводу значень. Для дробових чисел – не більше 4 знаків після коми. Вивід супроводжувати найменуваннями виведених змінних.

- виклик методу `outputDate`, в якому організувати виведення поточної дати і часу у вказаному форматі.

Методи `calcR`, `calcS` мають приймати вхідні дані, обчислювати змінні `R` та `S`, та повертати до методу `main` значення `R` та `S`.

### **Вимоги до оформлення Звіту з Лабораторної роботи.**

Номер варіанта слід обрати за номером у списку групи. Перший за списком групи студент виконує варіант 1, другий – варіант 2, і так далі. Так як варіантів в даній роботі всього 10, то студент із номером 11 має виконувати варіант 1, а студент із номером 26, має виконувати варіант 6.

Звіт має друкуватись в MS Word та включати в себе такі складові: титульний лист (додаток 2), Завдання на ЛР-1, Короткий опис рішення завдання разом із скріншотом варіанту завдання із табл.1; Текст програми (копія коду на білому фоні); Результати рішення програми (скріншот); Висновки.

**Звіт необхідно надіслати викладачу до 23.59 понеділка 4 жовтня.**

Увага! Назва файлу Звіту має бути чітко за зразком:

[ЛР-1\\_ТР-12\\_Петренко\\_ПАС.doc](#).

Відправляти Звіти треба на поштову скриньку [vpm11@ukr.net](mailto:vpm11@ukr.net)

Тема листа має бути такою: [ЛР-1\\_ТР-12\\_Петренко\\_ПАС](#)

### **Вимоги щодо захисту Звіту з Лабораторної роботи**

Лабораторні роботи, які вчасно здано (надіслано) викладачу, підлягають перевірці та захисту. Захист полягає у відповіді на три запитання, кожне з яких оцінюється певною сумою балів. Викладач оцінює правильність програми, відповідність вимогам, якість підготовки Звіту та правильність відповідей на питання. Питання на захисті ставляться виключно в рамках даної лабораторної роботи.

Таблиця 1

## Варіанти завдань

Варіант	Розрахункові формули	Значення вхідних даних	Формат дати і часу
1	$R = x^2(x+1)/b - \sin^2(x+a); s = \sqrt{\frac{xb}{a}} + \cos^2(x+b)^3$	$a=0.7$ $b=0.05$ $x=0.5$	Дата у форматі рр-мм-дд
2	$f = \sqrt[3]{mtgt +  csint }; z = m\cos(btsint) + c$	$m=2; c=-1$ $t=1.2$ $b=0.7$	Дата і час з точністю до мілісекунд
3	$y = btg^2 x - \frac{a}{\sin^2(x/a)}; d = ae^{-\sqrt{a}} \cos(bx/a)$	$a=3.2$ $b=17.5$ $x=-4.8$	Місяць, день рік та день тижня
4	$s = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24}; f = x(\sin x^3 + \cos^2 y)$	$x=0.335$ $y=0.025$	Час у форматі гг:хх:сс
5	$s = x^3 tg^2(x+b)^2 + \frac{a}{\sqrt{x+b}}; Q = \frac{bx^2 - a}{e^{ax} - 1}$	$a=16.5$ $b=3.4$ $x=0.61$	Дата у форматі дд міс рррр
6	$y = e^{-bt} \sin(at+b) - \sqrt{bt+a}; s = b \sin(at^2 \cos 2t) - 1$	$a=-0.5$ $b=1.7$ $t=0.44$	Дата у форматі дд місяць рррр
7	$y = \sin^3(x^2+a)^2 - \sqrt{\frac{x}{b}}; z = \frac{x^2}{a} + \cos(x+b)^3$	$a=1.1$ $b=0.004$ $x=0.2$	День тижня число і місяць
8	$a = \frac{2\cos(x-\pi/6)}{1/2 + \sin^2 y}; b = 1 + \frac{z^2}{3+z^2/5}$	$x=1.426$ $y=-1.220$ $z=3.5$	Дата у форматі дд-мм-рр
9	$w = \sqrt{x^2+b} - b^2 \sin^3(x+a)/x; y = \cos^2 x^3 - \frac{x}{\sqrt{a^2+b^2}}$	$a=1.5$ $b=15.5$ $x=-2.8$	День тижня та час
10	$c = \left  x^{y/x} - \sqrt[3]{y/x} \right ; f = (y-x) \frac{y-z/(y-x)}{1+(y-x)^2}$	$x=1.825$ $y=18.225$ $z=-3.298$	Час у форматі гг:хх та дата дд-мм-рр

## Короткі теоретичні відомості

### Огляд структури Java-програми

Всі Java-програми містять в собі 4 основні різновиди будівельних блоків: класи (classes), методи (methods), змінні (variables) і пакети (packages). На якій би мові Ви не програмували раніше, Ви скоріш за все вже добре знайомі з методами, які є не що інше, ніж функції чи підпрограми, та зі змінними, в яких зберігаються дані. З іншого боку, класи представляють собою фундамент об'єктно-орієнтованих властивостей мови. Поки що, для простоти, можна вважати клас деяким цілим, що містить у собі змінні і методи. Нарешті, пакети містять в собі класи і допомагають компілятору знайти ті класи, що потрібні йому для компіляції прикладної програми.

Java-програма може містити в собі будь-яку кількість класів, але один з них завжди має особливий статус, і безпосередньо взаємодіє з оболонкою часу виконання. Цей клас називають первинним класом (primary class).

Коли програма запускається з командного рядка, системі потрібен тільки один спеціальний метод, що повинен бути присутнім у первинному класі, - метод main. Коли ми будемо розглядати програмування аплетів, ми побачимо, що первинний клас аплету повинен містити вже декілька таких спеціальних методів. Розглянемо приклад програми мовою Java:

```
import java.util.Date;
// імпортування класу Date зі стандартного пакету java.util

public class OurPrimaryClass {
    public static void main(String[] S) {
        System.out.println("Hello, Java!");
        Date d = new Date();
        System.out.println("Date: "+d.toString());
    }
}
```

Наведена програма виводить на екран повідомлення "Hello, Java!" та поточну системну дату.

### Стандартні типи даних Java

Всі змінні та вирази у мові програмування Java можуть бути віднесені до однієї з двох великих груп типів: примітивних типів (primitive types), або посилальних типів (reference types), що містять у собі типи, визначені користувачем, і типи масивів. До примітивних типів відносяться стандартні,

вбудовані в мову типи для представлення чисельних значень, одиночних символів і логічних значень. Навпаки, усі посилальні типи є динамічними типами. Головні розбіжності між двома згаданими групами типів перелічені у наступній таблиці:

Таблиця 1.1 Порівняння примітивних і посилальних типів

Характеристика	Примітивні типи	Посилальні типи
Чи визначені в самій мові Java?	Так	Ні
Чи мають визначений розмір?	Так	Ні
Чи повинна для змінних цих типів виділятися пам'ять під час роботи програми?	Ні	Так

На практиці найважливішим розходженням між примітивними і посилальними типами є те, про що свідчить останній рядок цієї таблиці, а саме - що пам'ять для змінних посилального типу повинна виділятися під час виконання програми. Використовуючи змінні посилальних типів, ми повинні явно вимагати необхідну кількість пам'яті для кожної змінної перш, ніж ми зможемо зберегти в цій змінній деяке значення. Причина цього проста: оболонка часу виконання сама по собі не знає, яка кількість пам'яті потрібна для того чи іншого посилального типу.

Усього в мові Java визначено вісім примітивних типів, що перелічені в таблиці 1.2.

Таблиця 1.2. Примітивні типи мови Java

Тип	Розмір	Діапазон	Приклад
<b>byte</b>	1 байт	від -128 до 127	125
<b>short</b>	2 байти	від -32768 до 32767	-23
<b>int</b>	4 байти	від -2147483648 до 2147483647	2002300
<b>long</b>	8 байт	від -922372036854775808 до 922372036854775807	1243565
<b>float</b>	4 байти	Залежить від розрядності числа	1.2f
<b>double</b>	8 байт	Залежить від розрядності числа	123.4
<b>boolean</b>	1 біт	false, true	true

<b>char</b>	2 байти	Усі символи стандарту Unicode	'z'
-------------	---------	-------------------------------	-----

## Стандартні математичні функції

Оскільки мова Java є об'єктно-орієнтованою, то математичні функції повинні належати до деякого класу. Фактично існують два класи, що визначають математичні операції: Math та StrictMath, Останній призначений для виконання обчислень із "підвищеною точністю", але через поширення вбудованих у процесори математичних модулів, "звичайна" і "підвищена" точність у сучасній Java не розрізняються. Тому найчастіше використовується саме клас Math.

Усі стандартні математичні функції в мові Java є статичними методами класу Math, який визначений з модифікатором final, тобто не припускає спадкування. Крім того, клас Math має декілька визначених констант, наведемо дві з них:

Таблиця 1.3 Основні константи класу Math

Константа	Значення
Math.PI	Число $\pi = 3.14159...$
Math.E	Число $e = 2.71828...$

Основні статичні методи класу Math наведені у наступній таблиці:

Таблиця 1.4 Основні методи класу Math

Функція – метод	Пояснення
Math.abs(x)	Модуль числа x
Math.acos(x)	Арккосинус x
Math.asin(x)	Арксинус x
Math.atan(x)	Арктангенс x
Math.cbrt(x)	Кубічний корінь з x
Math.ceil(x)	Найближче число до x, що не містить дробової частини і більше за x
Math.cos(x)	Косинус x
Math.exp(x)	Експонента від x
Math.floor(x)	Найближче число до x, що не містить дробової частини і менше за x

Math.hypot(x,y)	Гіпотенуза прямокутного трикутника зі сторонами x, y
Math.log(x)	Натуральний логарифм x
Math.max(x,y)	Більше з двох чисел
Math.min(x,y)	Менше з двох чисел
Math.pow(x,y)	X в степені Y
Math.random()	Випадкове число з проміжку [0;1)
Math rint(x)	Найближче число до x, що не містить дробової частини
Math.round(x)	Найближче до x ціле число
Math.sin(x)	Синус x
Math.sqrt(x)	Квадратний корінь з x
Math.tan(x)	Тангенс x
Math.toDegrees(x)	Переведення кута з радіанів у градуси
Math.toRadians(x)	Переведення кута з градусів у радіани

Примітка. Починаючи з версії j2sdk 5.0 (30.09.2004) у мові Java з'явилась можливість імпорту статичних змінних та методів класу за допомогою директиви `import static` на початку програми. Наприклад:

```
import static java.lang.Math.*;

// імпортування статичних змінних і методів класу Math

public class OurPrimaryClass {

    public static void main(String[] S) { double x;

        x = sin(PI/6);

        // без статичного імпорту треба писати
        x=Math.sin(Math.PI/6);

        System.out.println(x);
    }
}
```

## Клас Date

Для роботи з датами і часом у стандартній бібліотеці мови програмування Java є декілька класів. Одним з них є клас `Date`. Цей клас знаходиться у пакеті `java.util`. Тому, для роботи з ним, на початку програми треба додати директиву `import java.util.Date;` Приклад його використання наведений на початку теоретичних відомостей до цієї лабораторної роботи.

## Виведення даних у консолі Java-програм

Для виведення інформації на консоль використовуються методи стандартного класу `PrintStream`:

- `print`
- `println`
- `printf`
- `format` (точна копія `printf`)

Кожна програма на мові Java містить стандартний об'єкт типу `PrintStream`

– `System.out`. Таким чином, виведення інформації на екран буде записуватися як `System.out.print(...)`, `System.out.println(...)`, або `System.out.printf(...)`.

Методи `print` та `println` повинні завжди мати один параметр – вираз будь-якого типу, що може бути автоматично приведений до рядкового типу.

Наприклад,

```
System.out.println("2+2="+ (2+2)); // буде виведено 2+2=4
System.out.println("Значення суми="+s);
```

```
// буде виведено Значення суми=xxx , де xxx – значення змінної S
```

Методи `printf` та `format` можуть мати список параметрів, що розділяються комами. Перший параметр – рядок, що містить текст для виведення і форматні шаблони для виведення значень інших параметрів.

Наприклад, якщо `a=2`, `b=3`

```
System.out.printf("Значення %d + %d = %d", a, b, a+b);
```

```
// буде виведено Значення 2 + 3 = 5
```

Форматні шаблони для виведення звичайних, символьних та числових типів мають наступний синтаксис:

`%[індекс_аргумента$][опції][ширина][.точність]перетворення`

Необов'язковий параметр `індекс_аргумента` є цілим числом, що вказує позицію в списку аргументів. Посилання на перший аргумент буде записане як `"1$"`, на другий – `"2$"`, і т.д.



Необов'язковий параметр опції – це набір символів, що змінюють формат виведення. Набір припустимих опцій залежить від типу перетворення.

Необов'язковий параметр ширина – це невід'ємне ціле число, що показує мінімальну кількість символів, що їх треба вивести.

Необов'язковий параметр точність – це невід'ємне ціле число, що зазвичай використовується для обмеження кількості символів, що будуть виведені. Його дія залежить від параметру перетворення.

Обов'язковий параметр перетворення – це один символ, що вказує як аргумент буде відформатований. Набір припустимих перетворень для вказаного аргументу залежить від типу даних аргументу.

Форматні шаблони для виведення типів, що означають дату і час мають такий синтаксис:

`%[індекс_аргумента$][опції][ширина]перетворення`

Індекс\_аргумента, опції, ширина – описані вище, а перетворення – два символи, де перший – 't', або 'T', а другий – описує тип перетворення. Інформацію про всі типи перетворень можна знайти на сайті Oracle, у розділі присвяченому мові програмування Java[1]. У таблиці 1.5 наведено основні з них.

Для виведення дати/часу недостатньо вказати лише символ форматування "t". Додатково треба вказати, яка саме частина дати/часу буде виводитись (день, місяць, рік, година, хвилина тощо). Для цього використовуються додаткові (уточнюючі) символи перетворень. Основні з них показані у таблиці 1.6.

Приклади використання `System.out.printf` для виведення на екран

<code>System.out.printf("Hello, World!");</code>	Hello, World!
<code>System.out.printf("Hello, World!\n");</code> або <code>System.out.printf("Hello, World!\n");</code>	Hello, World!
<code>System.out.printf("Sum %d + %d = %d", a, b, a+b);</code>	Sum 15 + 2 = 17
<code>System.out.printf("Const of Pi = %5.2f", Math.PI);</code>	Const of Pi = 3,14
<code>Date d = new Date();</code> <code>System.out.printf("Сьогодні %1\$te %1\$tB %1\$tY\n", d);</code>	Сьогодні 8 вересня 2015

Таблиця 1.5. Основні типи – символи перетворень

Перетворення	Категорія	Описання
'b', 'B'	boolean	Якщо аргумент <code>arg</code> є <code>null</code> , тоді результатом буде <code>"false"</code> . Якщо <code>arg</code> належить до типу <code>boolean</code> або <code>Boolean</code> , то результатом буде рядок – <code>"true"</code> або <code>"false"</code> в залежності від значення <code>arg</code> . У всіх інших випадках результатом буде <code>"true"</code> .
's', 'S'	general	Якщо аргумент <code>arg</code> є <code>null</code> , тоді результатом буде <code>"null"</code> . Якщо <code>arg</code> має метод <code>formatTo</code> , то він буде викликаний. Інакше, результат буде отриманий через виклик <code>arg.toString()</code> .
'c', 'C'	character	Результатом буде символ Unicode
'd'	integral	Результат буде відформатований, як ціле десяткове число
'e', 'E'	floating point	Результат буде відформатований, як число з плаваючою точкою у "науковому" форматі
'f'	floating point	Результат буде відформатований, як десяткове число
'g', 'G'	floating point	Результат буде відформатований, як число у "науковому" форматі, залежно від точності та значення після округлення.
't', 'T'	date/time	Префікс для символу перетворень дати і часу.
'%'	percent	Результатом буде символ <code>'%'</code> ( <code>"\u0025"</code> )
'n'	line separator	Результатом буде символ, що відокремлює рядки в залежності від платформи.

Таблиця 1.6 Основні символи перетворення для дати і часу

'H'	Година поточного дня, з нулем попереду, якщо потрібно, тобто 00 - 23.
'k'	Година поточного дня, без нуля попереду, тобто 0 - 23.
'M'	Хвилина поточної години, з нулем попереду, якщо потрібно, тобто 00 - 59.
'S'	Секунда поточної хвилини, з нулем попереду, якщо потрібно, тобто 00 - 60
'L'	Мілісекунди поточної секунди, з нулями попереду, якщо потрібно, тобто 000 - 999.
'N'	Наносекунди поточної секунди, тобто 000000000 - 999999999.
'B'	Повна назва місяця, відповідно мовних налаштувань, наприклад "вересня", "січня".
'b'	Скорочена назва місяця, відповідно мовних налаштувань, наприклад "вер", "січ".
'A'	Повна назва дня тижня, відповідно мовних налаштувань, наприклад "неділя", "понеділок"
'a'	Скорочена назва дня тижня, відповідно мовних налаштувань, наприклад "нд", "пн"
'C'	Чотиризначне число року поділене на 100, як два знаки з нулем попереду, якщо потрібно, тобто 00 - 99
'Y'	Рік, у вигляді чотиризначного числа
'y'	Останні дві цифри року, з нулем попереду, якщо потрібно, тобто 00 - 99.
'j'	День року, відформатований як три знаки, з нулями попереду, якщо потрібно, тобто 001 — 366 для Григоріанського календаря.
'm'	Місяць, відформатований як два знаки, з нулем попереду, якщо потрібно, тобто 01 - 13.
'd'	День місяця, у форматі двох знаків, з нулем попереду, якщо потрібно, тобто 01 - 31.

'e'	День місяця, у форматі двох знаків, тобто 1 - 31.
'R'	Час у форматі 24-годин як "%tH:%tM"
'T'	Час у форматі 24-годин як "%tH:%tM:%tS".
'D'	Дата у форматі як "%tm/%td/%ty".
'c'	Дата і час форматовані як "%ta %tb %td %tT %tZ %tY", наприклад "нд вер 06 12:13:21 EEST 2015".

## Введення даних з консолі

Для введення даних у мові програмування java можна скористатися різними засобами. Один з них використовує спеціальний об'єкт, що належить до класу Scanner. Цей клас містить методи для введення найрізноманітніших типів даних. Приклад його використання наведений нижче:

```
import java.io.*;
import java.util.*;

public class InOutExample {
    public static void main(String[] s) {
        Scanner s = new Scanner(System.in);
        // Читання цілого числа з рядка
        int i = s.nextInt();
        // Читання дійсного числа з рядку
        double x = s.nextDouble();
    }
}
```

## Створення і виконання Java-програм у середовищі NetBeans

1. Створіть новий проект, для цього:
2. Після запуску NetBeans у головному меню програми оберіть File -> New Project...
3. У вікні, що відкриється, оберіть категорію Java та вид проекту Java Application, та натисніть кнопку Next
4. У наступному вікні введіть ім'я проекту (Project Name). Ім'я проекту оберіть так, щоб було зрозуміло його призначення (наприклад First).
5. Оберіть місце розміщення файлів проекту (Project Location) та ім'я головного класу проекту (ім'я може містити ім'я пакету), наприклад, first.Main. Залиште відмітку у обох CheckBox'ах.
6. Натисніть кнопку Finish.

7. Впишіть код вашої програми у вікно редактора коду NetBeans.

8. Для запуску програми натисніть кнопку "Run" (на ній зображено зелений трикутник).

Примітка 1. Інші інструменти середовища NetBeans будуть розглянуті у наступних лабораторних роботах.

Примітка 2. Для запуску автономної програми на мові Java можна перейти в каталог, де розміщено скомпільований код програми – файли з розширенням class (у нашому випадку каталог classes проекту) та виконати таку команду (у режимі командного рядка):

```
javac <ім'я_класу_з_розширенням>  
java <ім'я_класу_без_розширення>
```

Наприклад,  
javac firstMain.java  
java firstMain

Приклад програми, що створена у середовищі NetBeans

```
package first;  
public class Main {  
    /**@param args the command line arguments */  
  
    public static void main(String[] args) {  
        Main prog = new Main();  
        prog.run();  
    }  
  
    private int calcSquare(int x) {  
        return x*x;  
    }  
  
    private void print(int x, int y) {  
        System.out.println("x="+x);  
        System.out.println("x^2="+y);  
    }  
  
    private void run() {  
        int x = 5;  
        int y = calcSquare(x);  
        print(x,y);  
    }  
}
```

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

## **ЗВІТ**

### **з лабораторної роботи № 1**

**з дисципліни «Програмування алгоритмічних структур»**

**Тема «Програмування лінійних алгоритмів»**

Варіант № \_\_\_\_

Виконав:

Студент групи ТР-12

Петренко П.П.

Дата здачі \_\_\_\_\_

Київ – 2021