Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра ЕОМ

Звіт

Лабораторна робота № 1

З дисципліни «Кросплатформні засоби програмування»

На тему: «Дослідження базових конструкцій мови Java»

Виконала: ст. гр. КІ-303

Кілик І.Р.

Перевірив:

доцент кафедри ЕОМ

Іванов Ю.С.

Львів 2025

ЗМІСТ

Розділ 1. Мета роботи.

Ознайомитися з базовими конструкціями мови Java та оволодіти навиками написання й автоматичного документування простих консольних програм мовою Java.

Розділ 2. Теоретичні відомості.

Автоматичне документування. При автоматичній генерації документації використовується утиліта javadoc, яка аналізує вміст між /** і */ та на його базі генерує документацію у форматі *.html. Коментарі між /** і */ прийнято починати з описового тексту, за яким слідують дескриптори. Використання дескрипторів полегшує як автоматичну генерацію документації, так і розуміння коду, до якого відноситься коментар. Дескриптор, на відміну від решти коментарів, починається з символу @ за яким слідує ім'я дескриптора. Оскільки документація генерується у форматі *.html, то між /** і */ допускається розташування html-тегів, включаючи рисунки.

Масив – структура даних, що зберігає набір значень однакового типу. Пам'ять під масив виділяється у керованій купі. При завершенні життєвого циклу масиву пам'ять, яку він займав, вивільняється збирачем сміття. Доступ до елементів масиву здійснюється за допомогою індексів. Індексація масивів у Java починається з 0. Для створення масиву у Java необхідно оголосити змінну-масив та ініціалізувати її. При створенні за допомогою оператора new масиву чисел всі його елементи ініціалізуються нулями (масиви типу boolean ініціалізуються значеннями false, масиви об'єктів ініціалізуються значеннями null). Після створення масиву змінити його розмір неможливо.

Багатовимірний масив – це масив, який складається з множини масивів. У Java немає багатовимірних масивів в принципі, а багатовимірні масиви реалізуються як множина одновимірних. Кількість вимірів масиву задається парами закриваючих і відкриваючих прямокутних дужок. Як і одновимірні, багатовимірні масиви перед використанням необхідно оголосити і ініціалізувати.

Зубчаті масиви. Завдяки тому, що багатовимірні масиви у Java реалізуються як множина одновимірних масивів, стає можливим реалізувати багатовимірні масиви з різною кількістю елементів у межах виміру. Синтаксис оголошення зубчатого масиву нічим не відрізняється від синтаксису оголошення звичайного багатовимірного масиву. Різниця є лише у способі ініціалізації, де використовується виділення пам'яті під різну кількість елементів у межах виміру.

Оператор циклу for з синтаксисом foreach дозволяє послідовно перебирати всі елементи набору даних без застосування лічильника. Таким набором даних може бути будь-який клас, що реалізує інтерфейс Iterable, або масив. Оператор циклу for з синтаксисом foreach має наступний вигляд:

for (змінна : набір даних)

оператори

При опрацюванні циклу змінній послідовно присвоюється кожен елемент набору даних (наприклад, елемент масиву) після чого виконується оператор.

Оператори переривання потоку виконання. До операторів переривання потоку виконання відносяться оператори break і continue. Вони призначені для переривання послідовності виконання операцій в циклах. У циклах дані оператори можуть використовуватися з мітками і без них.

Ввід з консолі. Для введення інформації з консолі необхідно створити об’єкт класу Scanner і зв’язати його з стандартним потоком вводу System.in.

Вивід на консоль. Популярним механізмом виводу на консоль є використання методу print об’єкту out з пакету System, який виводить переданий через параметр текстовий рядок на екран.

Ввід з текстового файлу. Для введення інформації з файлу необхідно підключити пакет java.io та створити об’єкт класу Scanner з об’єкту File.

Вивід у текстовий файл. Для виведення інформації у текстовому вигляді у файл треба підключити пакет java.io та створити об’єкт класу PrintWriter в конструкторі якого необхідно вказати назву файлу, що відкривається на запис.

Розділ 3. Завдання (варіант № 10).

1. Написати та налагодити програму на мові Java згідно варіанту. Програма має задовольняти наступним вимогам:

• програма має розміщуватися в загальнодоступному класі Lab2ПрізвищеГрупа;

• програма має генерувати зубчатий масив, який міститиме лише заштриховані області квадратної матриці згідно варіанту (рис. 3.1);

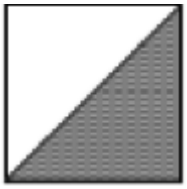


Рис. 3.1. Заштрихована область квадратної матриці.

• розмір квадратної матриці і символ-заповнювач масиву вводяться з клавіатури;

• при не введені або введенні кількох символів-заповнювачів відбувається коректне переривання роботи програми;

• сформований масив вивести на екран і у текстовий файл;

• програма має володіти коментарями, які дозволять автоматично згенерувати документацію до розробленої програми.

2. Автоматично згенерувати документацію до розробленої програми.

3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.

4. Дати відповіді на контрольні запитання.

Розділ 4. Хід роботи

1. Запустити на виконання середовище Eclipse IDE.

2. Дослідити тестову програму, що наведена в методичних вказівках.

3. Почати створення власного проекту. Для цього слід викликати підпункут меню File->New->Project… У вікні, що відкриється слід вибрати Java Project (рис. 4.1) та натиснути кнопку ”Next>”..

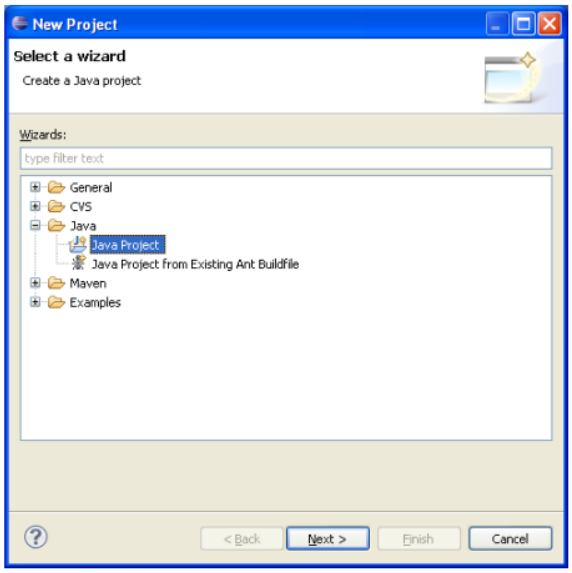


Рис. 4.1. Діалогове вікно вибору типу проекту в Eclipse IDE.

4. У вікні, що відкрилося, необхідно вказати назву нового проекту, місце розташування проекту, версію JRE на яке орієнтована програма, топологію проекту (Project layout) та натиснути кнопку ”Next>” (рис. 4.2).

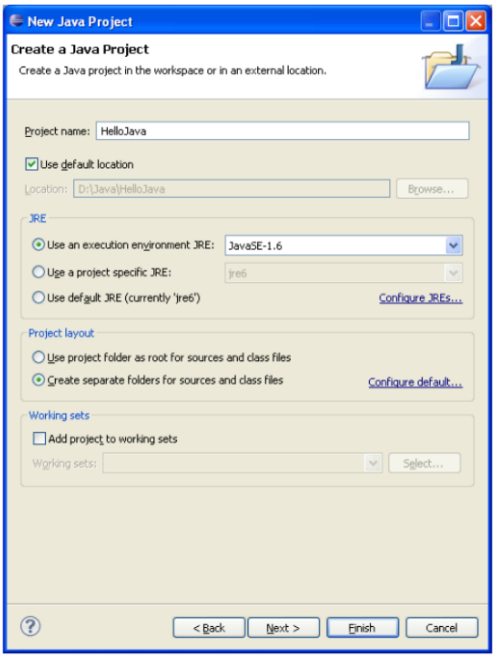


Рис. 4.2. Діалогове вікно створення нового проекту в Eclipse IDE.

5. У вікні, що відкрилося, натиснути кнопку ”Finish” (рис. 4.3).

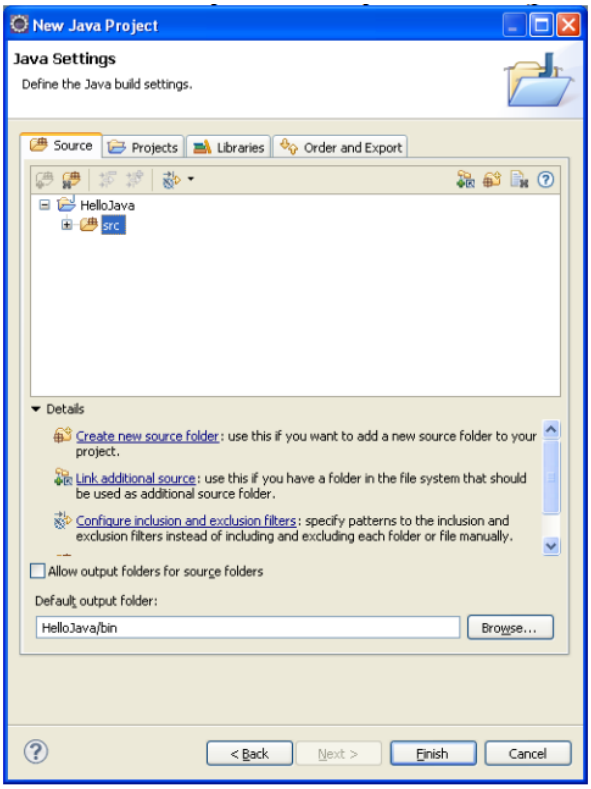


Рис. 4.3. Діалогове вікно налаштування властивостей побудови нового проекту в Eclipse IDE.

6. У вікні, що з’явилося (рис. 4.4), викликати підпункт меню File->New->File. У вікні, що відкрилося слід задати каталог, де розташовуватиметься новий файл з кодом, назву файлу з розширенням .java та натиснути кнопку ”Finish” (рис. 4.5).

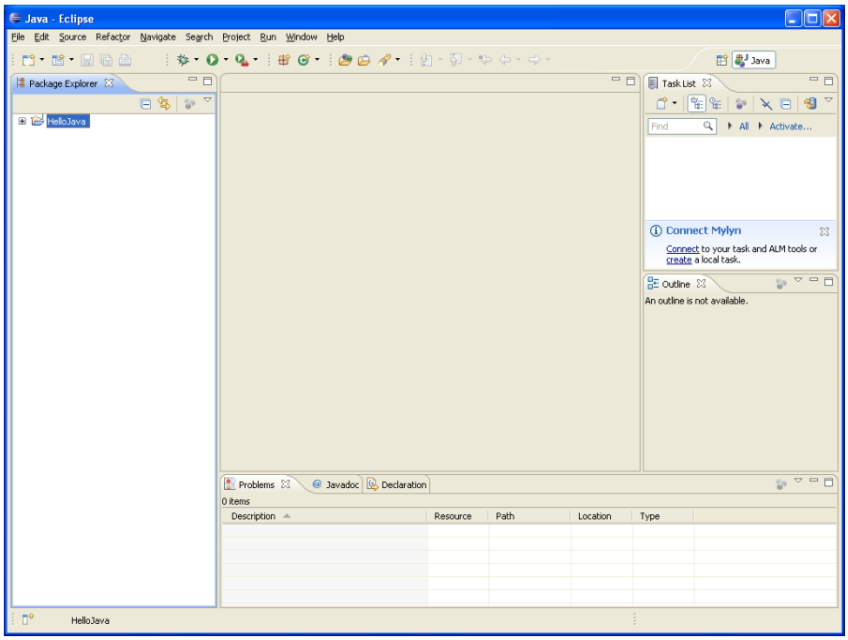


Рис. 4.4. Середовище Eclipse IDE з відкритим проектом Java.

7. У створеному файлі написати програму згідно завдання. Програма має обов'язково містити public клас, назва якого співпадає з назвою файлу.

8. Запустити програму на виконання. Для цього слід вибрати підпункт меню Run->Run.

9. Встановити точки переривання, запустити налагоджувач (Run->Debug) та покроково дослідити процес виконання програми (рис. 4.6).

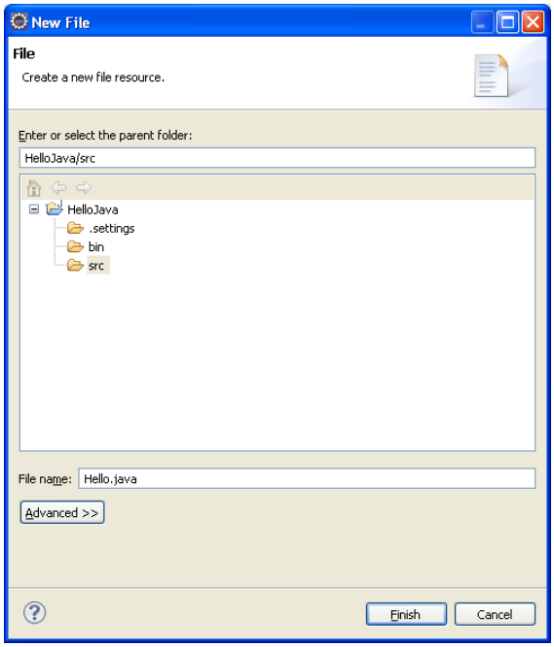


Рис. 4.5. Створення нового файлу.

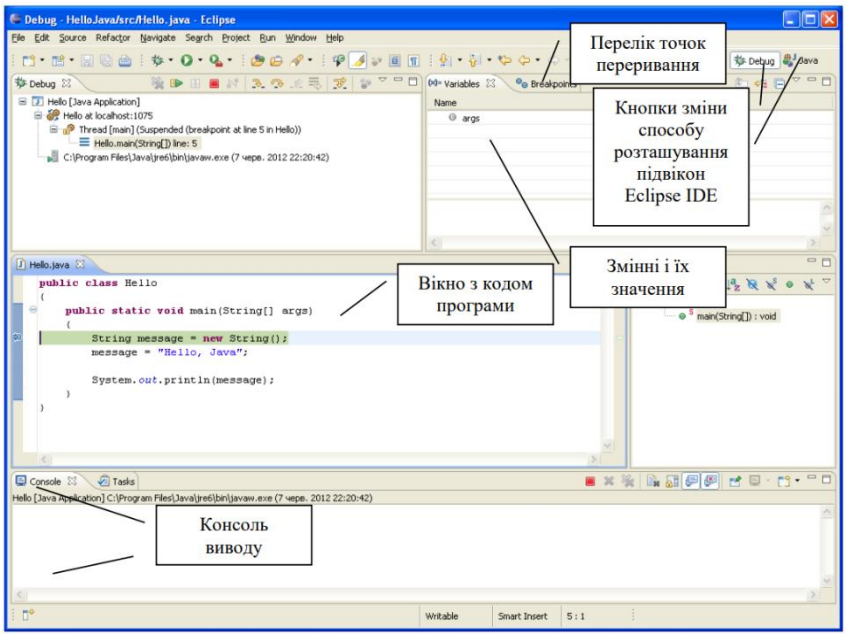


Рис. 4.6. Процес налагодження програми.

10. Автоматично згенерувати документацію у каталог \doc вашого проекту. Проаналізувати автоматично згенеровану документацію.

Висновки

У результаті виконання лабораторної роботи створено програму, яка генерує зубчастий масив згідно індивідуального завдання. Вихідні дані для формування зубчастого масиву вводяться з клавіатури, а сформований виводиться на консоль та у файл. Розроблена програма також містить коментарі, які дозоляють автоматично згенерувати документацію.

Список використаних джерел

1. *Schildt H. Java: The Complete Reference, 12th Edition. / Herbert Schildt. – McGraw Hill, 2021. – 1280 p.*
2. *Java SE Documentation at a Glance [електронний ресурс]. – Режим доступу до документації: http://www.oracle.com/technetwork/java/javase/documentation/index.html*

Додатки

Додаток А. Вихідний код програми

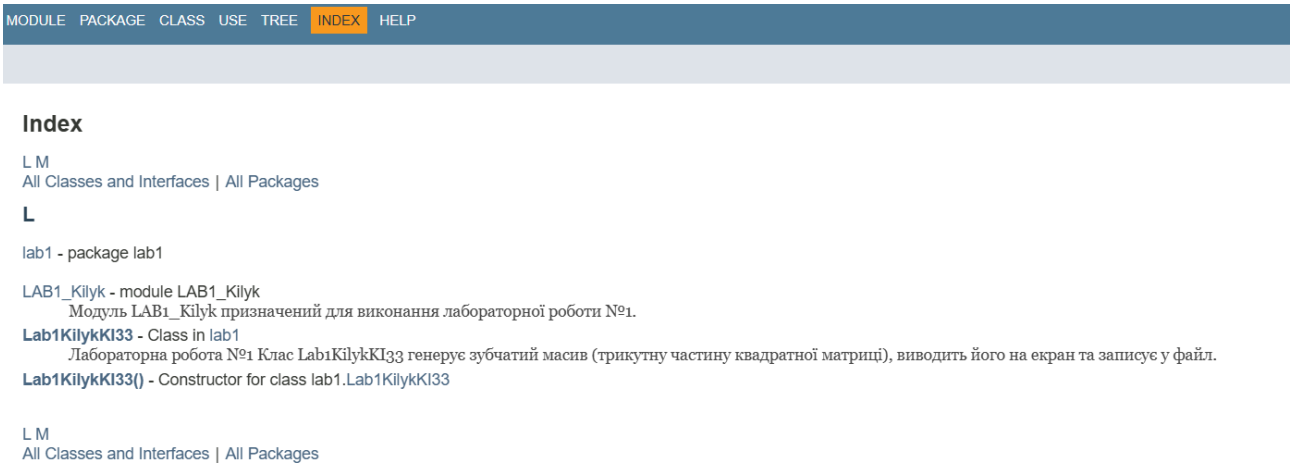
```
package lab1;
import java.io.IOException;
import java.io.PrintStream;
import java.nio.charset.StandardCharsets;
import java.util.Scanner;
/**
* Лабораторна робота №1
* Клас Lab1KilykKI33 генерує зубчатий масив (трикутну частину квадратної матриці),
* виводить його на екран та записує у файл.
* <p>Виконано згідно індивідуального варіанту</p>
*
* @author Kilyk
* @version 1.0
* @since 1.0
*/
public class Lab1KilykKI33 {
/**
* Точка входу у програму
* @param args аргументи командного рядка
* @throws IOException якщо не вдасться створити файл для виводу
*/
    public static void main(String[] args) throws IOException {

        PrintStream out = new PrintStream(System.out, true, StandardCharsets.UTF_8);
        Scanner in = new Scanner(System.in);
        out.print("Введіть розмір трикутника: ");
        int nRows = in.nextInt();
```

```
        in.nextLine();
        out.print("Введіть символ-заповнювач: ");
        String filler = in.nextLine();

        if (filler.length() != 1) {
            out.println("Помилка");
            return;
        }
        char fillChar = filler.charAt(0);

        try (PrintStream fileOut = new PrintStream("output.txt",
StandardCharsets.UTF_8)) {
            for (int i = 0; i < nRows; i++) {
                for (int k = 0; k < nRows - i - 1; k++) {
                    out.print("\t");
                    fileOut.print("\t");
                }

                for (int j = 0; j <= i; j++) {
                    out.print(fillChar + "\t");
                    fileOut.print(fillChar + "\t");
                }
                out.println();
                fileOut.println();
            }

            out.println("Малюнок збережено ");
        }
        in.close();
```

Додаток Б. Результат виконання програми

```
Введіть розмір трикутника: 4
Введіть символ-заповнювач: *
                        *
                    *       *
                *       *       *
            *       *       *       *
Малюнок збережено
```

Додаток В. Фрагмент згенерованої документації.



MODULE PACKAGE CLASS USE TREE INDEX HELP

Index

L M
All Classes and Interfaces | All Packages

L

lab1 - package lab1

LAB1_Kilyk - module LAB1_Kilyk
Модуль LAB1_Kilyk призначений для виконання лабораторної роботи №1.
Lab1KilykKI33 - Class in lab1
Лабораторна робота №1 Клас Lab1KilykKI33 генерує зубчатий масив (трикутну частину квадратної матриці), виводить його на екран та записує у файл.
Lab1KilykKI33() - Constructor for class lab1.Lab1KilykKI33

L M
All Classes and Interfaces | All Packages