



Звіт

до лабораторної роботи № 1

З дисципліни: «Кросплатформенні засоби програмування»

На тему: «ДОСЛІДЖЕННЯ БАЗОВИХ КОНСТРУКЦІЙ МОВИ JAVA»

Варіант 16

Виконав: ст.гр. КІ-301

Онисько М.М.

Прийняв: доц.

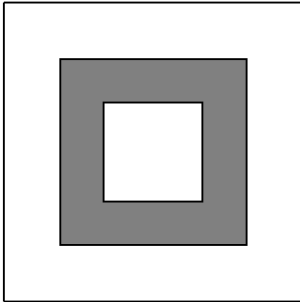
Майдан М.В.

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

Завдання:

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;
 - розмір квадратної матриці і символ-заповнювач масиву вводяться з клавіатури;
 - при не введенні 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 відповідь на контрольні запитання.

Варіант завдання:

№	
16	

Виконання роботи:

Код програми:

```
package lab2kzp;

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

/**
 * Клас Lab20nysko301 реалізує приклад програми до лабораторної роботи №1
 */
```

```

* @author Mykola Onysko
* @version 1.2
* @since version 1.1
*
**/

public class Lab2Onysko301 {
    /**
     * Статичний метод main є точкою входу в програму
     *
     * @param args
     * @throws FileNotFoundException
     */
    public static void main(String[] args) throws FileNotFoundException{

        int nRows;

        Scanner scan = new Scanner(System.in);

        File dataFile = new File("MyFile.txt");
        PrintWriter fout = new PrintWriter(dataFile);

        System.out.print("Enter the size of the square matrix: ");
        nRows = scan.nextInt();
        scan.nextLine();
        System.out.print("\nEnter the filler character: ");
        String filler;
        filler = scan.nextLine();

        char[][] arr = new char[nRows][];
        // Розмір порожніх рядків вгорі та знизу (20%)
        int emptyRowCount = (int) Math.round(nRows * 0.2);

        // Розмір рядків з коробкою (60%)
        int boxRowCount = (int) Math.round(nRows * 0.6);

        for (int i = 0; i < nRows; i++) {
            if (i < emptyRowCount || i >= nRows - emptyRowCount) {
                // Порожні рядки
                arr[i] = new char[0];
            } else {
                // Рядки з коробкою
                arr[i] = new char[(int) (nRows*0.8)];
            }
        }

        // Розрахунок розмірів поля та рамки
        int boxSize = (int) Math.round(nRows * 0.2);
        int borderSize = (int) Math.round(nRows * 0.2);
        exit:

        if(filler.length() == 1){

            // створення коробки
            for (int i = borderSize; i < nRows - borderSize; i++) {
                for (int j = borderSize; j < nRows - borderSize; j++) {
                    if (i < borderSize + boxSize || i >= nRows - borderSize - boxSize
||
                    j < borderSize + boxSize || j >= nRows - borderSize -
boxSize) {

```

```

arr[i][j] = (char) filler.codePointAt(0); // заповнення
    }
}

// вивід зубчастого масиву в консоль та запис у файл
for (int i = 0; i < nRows; i++) {
    for (int j = 0; j < arr[i].length; j++) {
        System.out.print(arr[i][j] + " ");
        fout.print(arr[i][j] + " ");
    }
    System.out.println();
    fout.println();
}

else if (filler.length()==0){
    System.out.print("\nНе введено символ заповнювач");

    break exit;

}

else{
    System.out.print("\nЗабагато символів заповнювачів");
    break exit;

}

// Закриваємо файл та виводимо дані у файл
fout.flush();
fout.close();
}
}

```

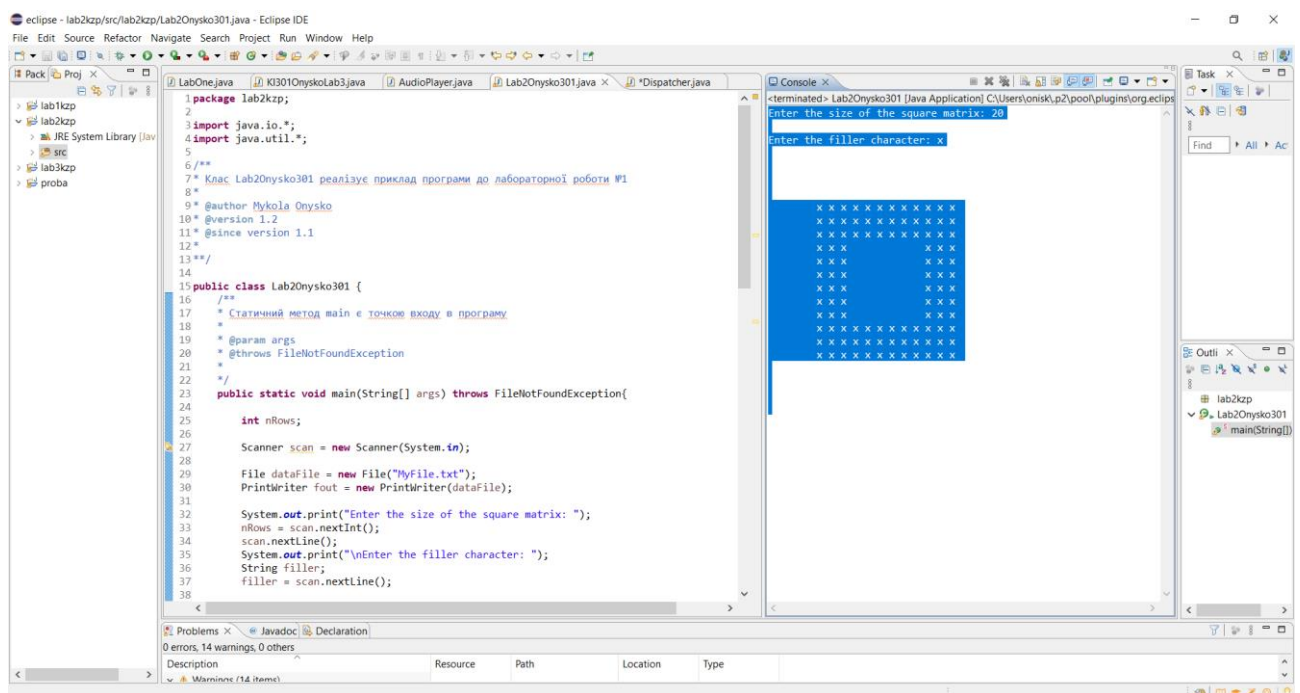


Рис.1 Код та вивід програми.

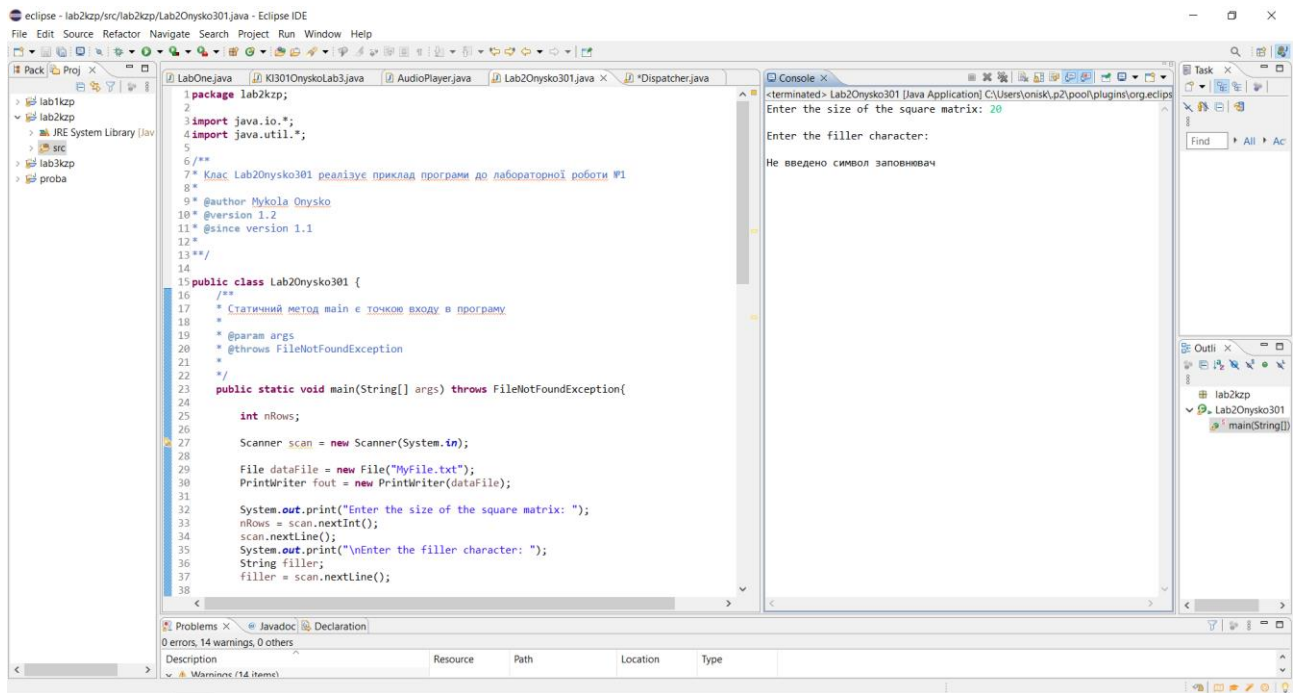


Рис.2 Код та вивід програми.

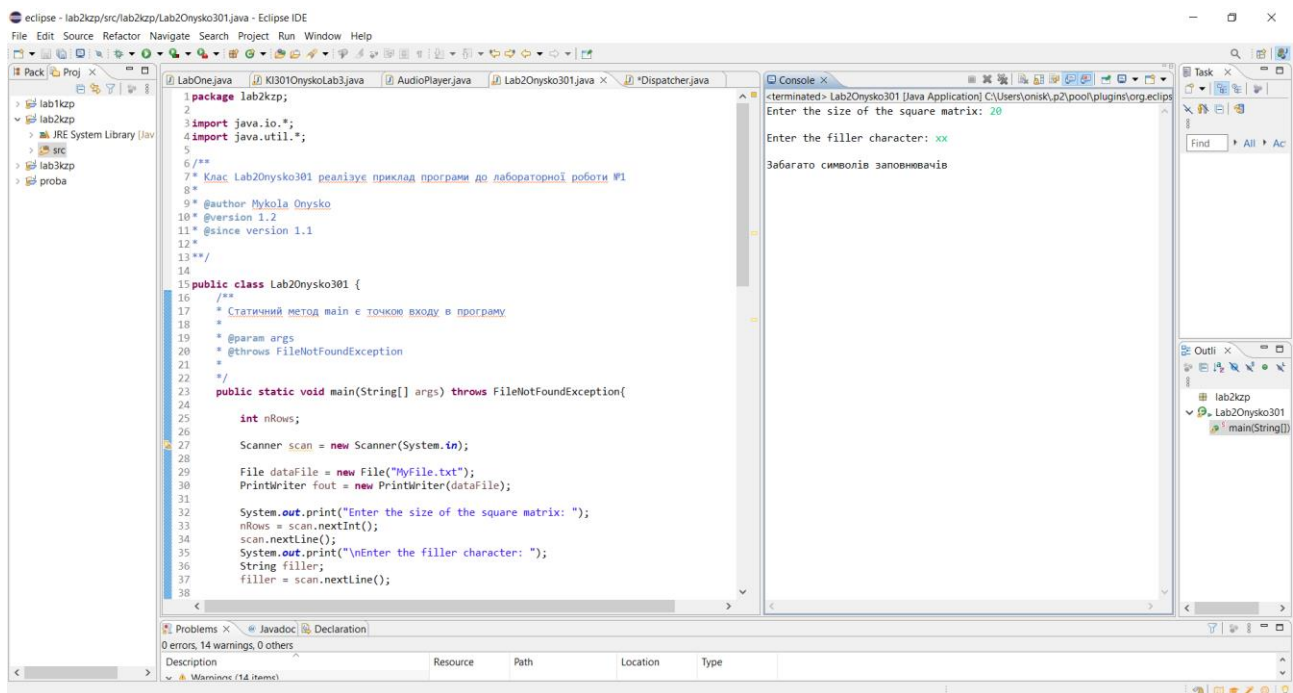


Рис.3 Код та вивід програми.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.3324]
(c) Корпорація Майкрософт. Усі права захищені.

C:\eclipse\lab2kzp\src\lab2kzp>javac Lab20nysko301.java

C:\eclipse\lab2kzp\src\lab2kzp>cd..

C:\eclipse\lab2kzp\src>javadoc lab2kzp\Lab20nysko301.java
Loading source file lab2kzp\Lab20nysko301.java...
Constructing Javadoc information...
Standard Doclet version 11.0.16.1
Building tree for all the packages and classes...
Generating .\lab2kzp\Lab20nysko301.html...
Generating .\lab2kzp\package-summary.html...
Generating .\lab2kzp\package-tree.html...
Generating .\constant-values.html...
Building index for all the packages and classes...
Generating .\overview-tree.html...
Generating .\index-all.html...
Building index for all classes...
Generating .\allclasses-index.html...
Generating .\allpackages-index.html...
Generating .\deprecated-list.html...
Building index for all classes...
Generating .\allclasses.html...
Generating .\allclasses.html...
Generating .\index.html...
Generating .\help-doc.html...

C:\eclipse\lab2kzp\src>
```

Рис.4 Проміжний етап генерування автоматичної документації.

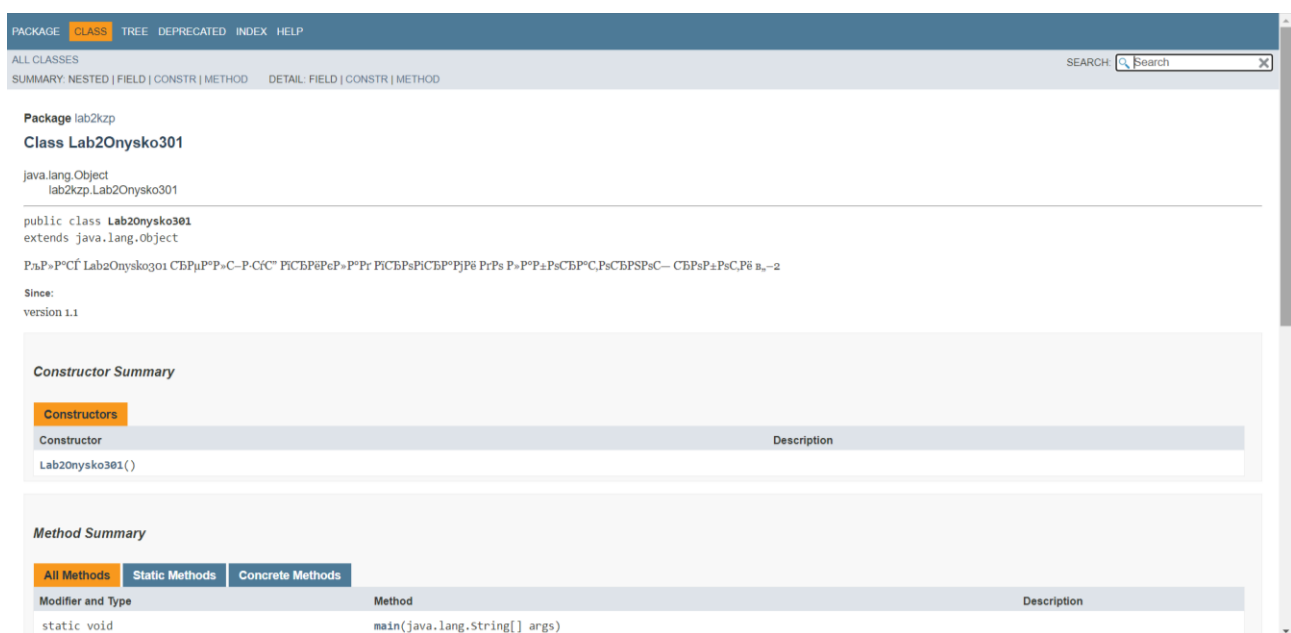


Рис.5 Автоматично згенерована документація до розробленої програми.

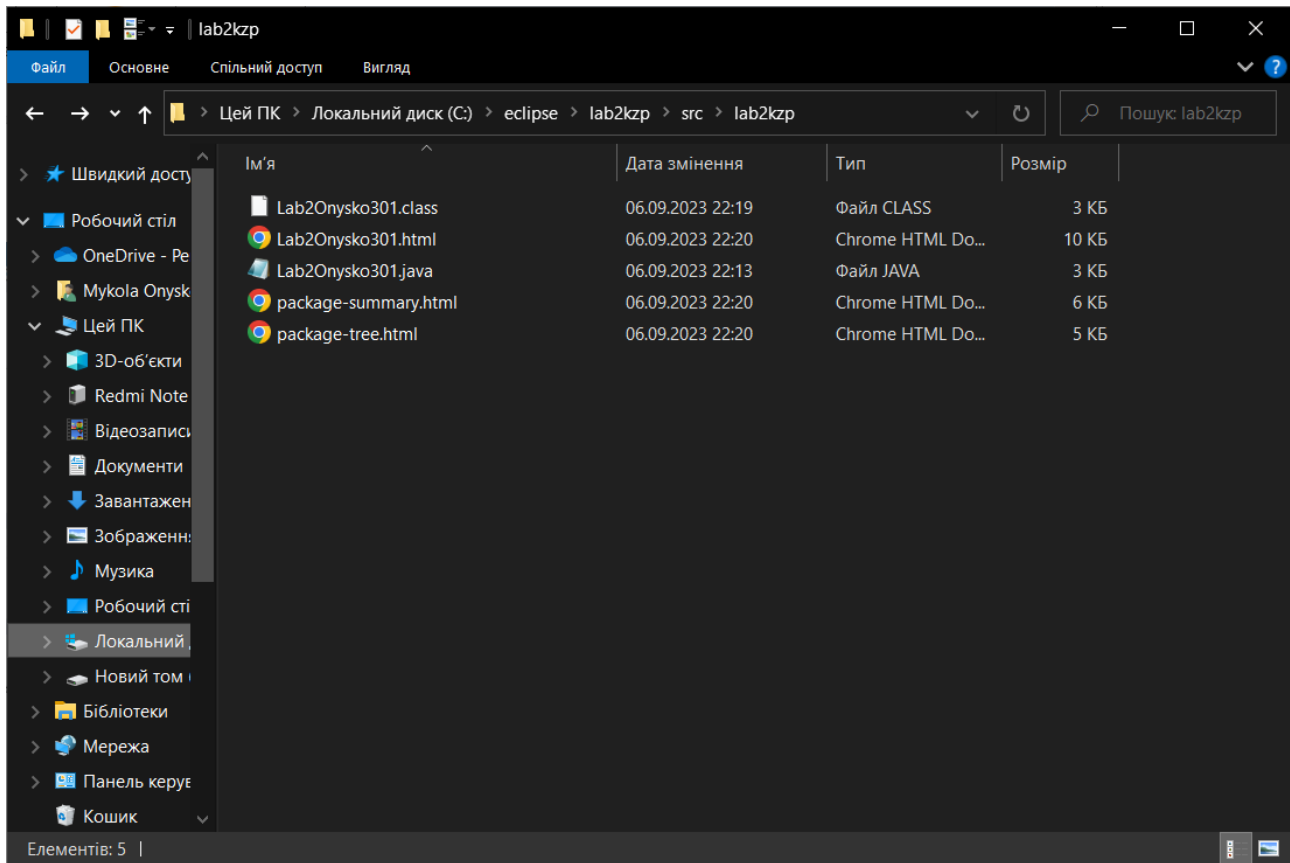


Рис.6 Документація та програма, які розміщені у папці.

Висновок:

У результаті роботи реалізована програма, яка створює зубчатий масив з вказаними параметрами та здійснює вивід цього масиву на екран і в текстовий файл. Освоєно базові конструкціями мови Java та оволодіти навиками написання й автоматичного документування простих консольних програм мовою Java.

Контрольні питання:

1. Які дескриптори використовуються при коментуванні класів?

В Java, для коментування класів використовуються JavaDoc коментарі. Вони починаються з `/**` і можуть містити опис класу, його поля та методи, анотації і багато іншої інформації.

2. Які дескриптори використовуються при коментуванні методів?

Для коментування методів також використовуються JavaDoc коментарі. Вони розміщуються перед оголошенням методу і містять опис методу, його параметри, тип повернення і можуть включати приклади використання.

3. Як автоматично згенерувати документацію?

Для автоматичного генерування документації в Java використовують інструмент JavaDoc. Ви можете запустити JavaDoc з командного рядку та вказати шлях до вашого Java-коду. Він створить HTML-файли з документацією на основі JavaDoc коментарів.

4. Які прості типи даних підтримує Java?

Java підтримує наступні прості типи даних:

- byte: 8-бітове ціле число
- short: 16-бітове ціле число
- int: 32-бітове ціле число
- long: 64-бітове ціле число
- float: 32-бітове число з плаваючою точкою
- double: 64-бітове число з плаваючою точкою
- char: 16-бітовий символ
- boolean: значення true або false

5. Як оголосити змінну-масив?

Змінну-масив можна оголосити наступним чином:

```
int[] myArray; // Оголошення масиву цілих чисел
String[] names; // Оголошення масиву рядків
```

6. Які керуючі конструкції підтримує Java?

Java підтримує різні керуючі конструкції, включаючи:

- if: умовний оператор
- for: цикл для ітерації через послідовність
- while: цикл з умовою на виконання
- do-while: цикл, який завжди виконується хоча б один раз
- switch: перемикач для обробки багатьох можливих значень
- break і continue: оператори для керування виконанням циклів
- return: для повернення значення з методу

7. В чому різниця між різними варіантами оператора for?

В Java існують кілька варіантів оператора for:

for (ініціалізація; умова; інкремент) { ... }: Стандартний for-цикл з ініціалізацією, умовою та інкрементом.

for (тип змінної з ініціалізацією : колекція) { ... }: Розширений for-цикл для ітерації через елементи колекції або масиву.

8. Як здійснити ввід з консолі?

Для вводу даних з консолі в Java можна використовувати клас Scanner. Ось приклад:

```
Scanner scanner = new Scanner(System.in);
System.out.print("Введіть число: ");
int number = scanner.nextInt();
```

9. Як здійснити ввід з текстового файлу?

Для вводу з текстового файлу в Java використовують класи FileReader та BufferedReader. Ось приклад:

```
try (BufferedReader br = new BufferedReader(new FileReader("файл.txt")))
{
    String line;
    while ((line = br.readLine()) != null) {
        System.out.println(line);
    }
} catch (IOException e) {
    e.printStackTrace();
}
```

10. Як здійснити запис у текстовий файл?

Для запису у текстовий файл в Java використовують класи `FileWriter` та `BufferedWriter`. Ось приклад:

```
try (BufferedWriter bw = new BufferedWriter(new FileWriter("файл.txt")))
{
    String text = "Це текст, який буде записано в файл.";
    bw.write(text);
} catch (IOException e) {
    e.printStackTrace();
}
```