

**Національний університет «Львівська політехніка»**



**Звіт**

до лабораторної роботи № 2

з дисципліни: «Кросплатформенні засоби програмування»

на тему: «ДОСЛІДЖЕННЯ БАЗОВИХ КОНСТРУКЦІЙ МОВИ  
JAVA»

Варіант №14

Виконала:  
ст.гр. КІ-34  
Олексій М. В.  
Прийняв:  
Іванов Ю.С.

**Львів 2022**

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

## ТЕОРЕТИЧНІ ВІДОМОСТІ

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 слідує дескриптор. 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.

Для автоматичної генерації документації між `/** і */` можна розмістити:

- коментарі до класу;
- коментарі до методів;
- коментарі до полів;
- загальні коментарі.

Коментарі до класу мають бути розміщені після директив `import` безпосередньо перед визначенням класу. Найчастіше цей коментар має вигляд одного або кількох коротких речень:

```
/**
```

```
    Об'єкт класу Person описує особу.
```

```
    Особа має властивості: ім'я, прізвище та стать.
```

```
*/
```

Коментарі до методів розташовуються безпосередньо перед методами, які вони описують. Крім дескрипторів загального призначення для коментування методів використовуються дескриптори:



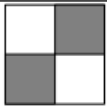
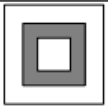


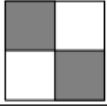
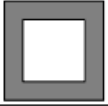


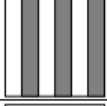
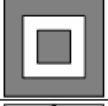
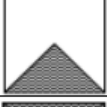
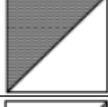

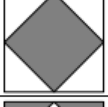




- `@param` змінна опис Цей дескриптор додає в опис методу розділ “parameters”. Опис цього елемента може складатися з кількох рядків та містити `html`-теги. Всі дескриптори `@param`, що відносяться до одного методу слід групувати разом.
- `@return` опис Цей дескриптор додає в опис методу розділ “returns”. Опис цього елемента може складатися з кількох рядків та містити `html`-теги.
- `@throws` опис\_класу

Цей дескриптор додає в опис методу інформацію про класи об'єкти яких можуть генеруватися при виключних ситуаціях. Відомості про кожен клас слід описувати в окремому дескрипторі `@throws`.

## 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.

## ВАРІАНТИ ЗАВДАНЬ

№		№		№		№	
1		6		11		16	
2		7		12		17	
3		8		13		18	
4		9		14		19	
5		10		15		20	

## Виконання:

## Код програми:

```
package lab2;

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

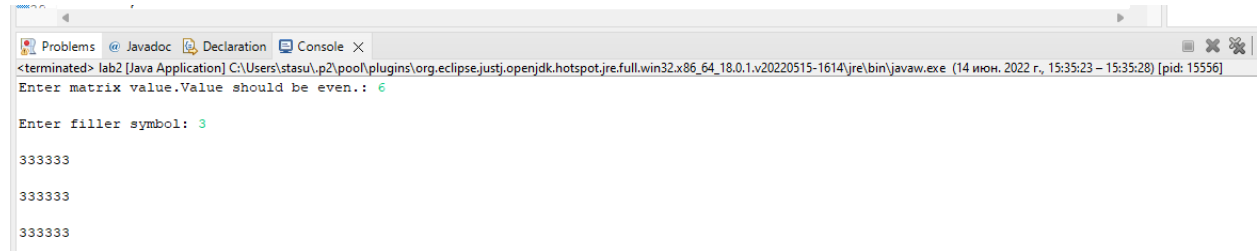
public class lab2 {
    public static void main(String[] args) throws FileNotFoundException {
        int nRows;
        String filler;
        Scanner in = new Scanner(System.in);
        File dataFile = new File("MyFile.txt");
        PrintWriter fout = new PrintWriter(dataFile);

        System.out.print("Enter matrix value.Value should be
even.: ");
        nRows = in.nextInt();
        in.nextLine();
        if (nRows % 2 == 1)
        {
            System.out.print ("Odd matrix value!");
            fout.close();
            return;
        }
        char[][] arr = new char[nRows][nRows];
        for (int i = 0; i < nRows; i++) {
            for (int j = 0; j < nRows; j++) {
                arr[i][j] = ' ';
            }
        }
        System.out.print("\nEnter filler symbol: ");
        filler = in.nextLine();
        if (filler.length() > 1)
        {
            System.out.print ("Too much symbols!");
            fout.close();
            return;
        }
        else if (filler.length() == 0)
        {
            System.out.print ("Small number of symbols!");
            fout.close();
            return;
        }

        int temp = 0;
        for (int i = temp; i < nRows; i++) {
            i++;
            System.out.print("\n");
            fout.print("\n");
            for (int j = 0; j < nRows; j++) {
                arr[i][j] = filler.charAt(0);
                System.out.print(filler);
                fout.print(filler);
            }
            System.out.print("\n");
            fout.print("\n");
        }
        fout.flush();
    }
}
```

```
        fout.close();  
    }  
  
}
```

## Виконання програми:



The screenshot shows a Java console window titled "lab2 [Java Application]". The console output is as follows:

```
<terminated> lab2 [Java Application] C:\Users\stasu\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_18.0.1.v20220515-1614\jre\bin\javaw.exe (14 июн. 2022 г., 15:35:23 – 15:35:28) [pid: 15556]  
Enter matrix value.Value should be even.: 6  
  
Enter filler symbol: 3  
  
333333  
  
333333  
  
333333
```

**Висновок:** ознайомилась з базовими конструкціями мови Java та оволоділа навиками написання й автоматичного документування простих консольних програм мовою Java.