JOBSHEET 11

LOOPING 2

DAVA RIZKY H

254107020246

1. Objective

- Students are able to understand the concept of nested loop
- Students are able to explain nested loop writing format
- Students are able to implement nested loop flowchart using Java programming language

2. Laboratory

2.1 Experiment 1: Loop Review

1. This experiment is aimed at reviewing the loop that had been studied in the previous week. In experiment 1, a program will be made to make a view * N times sideways.
2. Create a new class with the name **Star** and save it in the **Star.java** file
3. Write the basic structure of the Java programming language which contains the **main()** function
4. Because the program requires input from the keyboard, it is necessary to import the **Scanner** class, so add the import syntax in the top line of the program.
5. In the **main()** function that has been created, declare a **Scanner** object with the name **sc**
6. On the next line, write the instructions for entering the value to be stored in variable **N**

```
System.out.print("Enter the value of N: ");
int N = sc.nextInt();
```

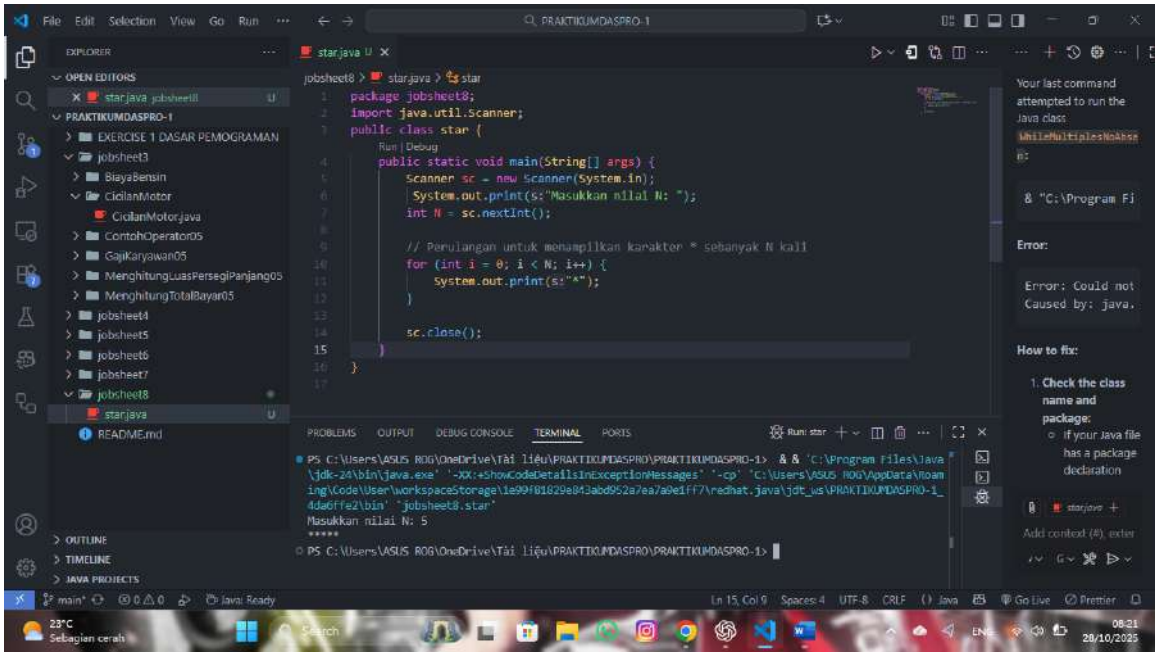
7. On the next line, create a loop syntax using **for**

```
for (int i = 1; i <= N; i++) {
    System.out.print("*");
}
```

Note: *please note that the print command is used, not println because we want to display without any new lines*

8. Compile and run the program.
9. Observe the results, match the results of the running programs that you have created according to the following display

```
Enter the value of N: 5
*****
```



Questions!

1. If in **for** loop, the initialization **i = 1** is changed to **i = 0**, what is the result? How can It be like that?
2. If in **for** loop, condition **i <= N** is changed to **i > N**, what is the result? How can It be like that?
3. If in **for** loop, the condition for step **i++** is changed to **i--** what is the result? How can It be like that?

ANSWER

1. When i starts from **0** instead of **1**, the loop will run from **0 to N**, not from **1 to N**. That means the loop executes **N + 1 times**, because **0** is included as a valid iteration.
2. The condition i > N means the loop will only execute if **i is greater than N**. However, at the start, i is usually **less than or equal to N**, so the condition is **false** from the beginning, and the loop is **skipped**.
3. i++ means the variable increases, while i-- decreases it. If the condition is i <= N, then i will keep getting smaller and **never reach a value greater than N**, so the condition will **always be true** — the loop never stops.

2.2 Experiment 2: Square Star

1. In Experiment 2, an experiment was carried out on nested loops. The case that will be solved is to create a square display *, with side lengths of N. Suppose N is entered as 5, then the results are as follows:

```
*****
*****
*****
*****
*****
```

2. If you look closely, it's actually like the case of Experiment 1, isn't it? In Experiment 1, for example, the input N has a value of 5, then what will be produced is ***** (we can think of this as an **inner loop** that displays 5 stars *****), then for the case of Experiment 2, this is not the result of experiment 1 does it just need to be repeated N times? (by adding an **outer loop** to repeat the **inner loop** process N times)
3. Create a new class with the name **Square** and save it in the **Square.java** file
4. Because the program requires input from the keyboard, it is necessary to import the **Scanner** class, so add the import syntax in the top line of the program.
5. Create a **main()** function, and add the same program code as the contents of the **main()** function in Experiment 1

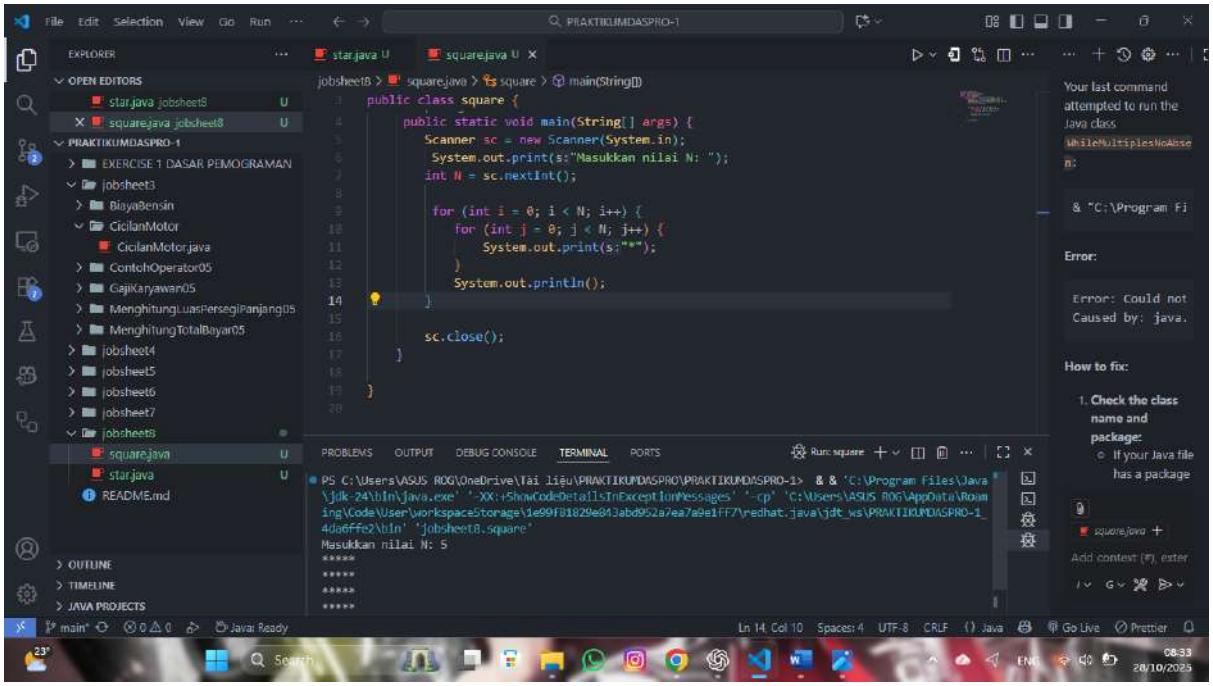
```
Scanner sc = new Scanner(System.in);
System.out.print("Enter the value of N: ");
int N = sc.nextInt();
for (int i = 1; i <= N; i++) {
    System.out.print("*");
}
```

6. Compile and run the program. Make sure the results given are the same as in Experiment 1
7. Pay attention to the iterative syntax used to print * N times sideways. In step 5, we make **for** loop structure (red box) as an **inner loop**
8. Furthermore, the inner loop needs to be repeated N times to display the * symbol to generate output like step 1. Thus, it is necessary to add an **outer loop**

```
for (int iOuter = 1; iOuter <= N; iOuter++) {
    for (int i = 1; i <= N; i++) {
        System.out.print("*");
    }
    System.out.println("");
}
```

9. Compile and run the program.
10. Observe the results, match the results of the running programs that you have created according to the following display

```
Enter the value of N: 5
*****
*****
*****
*****
*****
```



Questions!

1. Pay attention to outer loop. If in **for** syntax, the initialization **iOuter = 1** is changed to **iOuter = 0**, what is the result? How can it be like that?
2. Return the program to normal with initialization **iOuter = 1**. Then pay attention to the inner loop. If in **for** syntax, the initialization **i = 1** is changed to **i = 0**, what is the result?

   How can it be like that?
3. What is the difference between outer loop and inner loop?
4. Why is it necessary to add the syntax **System.out.println();** under inner loop? What will happen if the syntax is omitted?
5. Commit and push the changes to GitHub

ANSWER

1. **iOuter** controls **how many rows** will be printed (the outer loop). If it starts from **0**, the loop runs from **0 to N**, not from **1 to N**. That means the outer loop executes **N + 1 times**, so the output will have **one additional line**.
2. The **inner loop** controls **how many stars (*) are printed per row**. If it starts from **0**, it runs from **0 to N**, which means **N + 1 iterations**. As a result, each row prints **one extra star**.

3. • **Outer loop:**
   Controls **how many rows** or how many times the inner loop will be repeated.
   • **Inner loop:**
   Controls **what is printed inside each row**, such as how many stars or numbers appear in one line.
4. `System.out.println();` is used to move to a new line after the inner loop finishes printing one row.
   If it is removed, then all the stars will appear on the same line without line breaks.
5. https://github.com/rizkydava1233-ops/PRAKTIKUMDASPRO/blob/main/jobsheet8/square.java

2.3 Experiment 3: Triangle Star

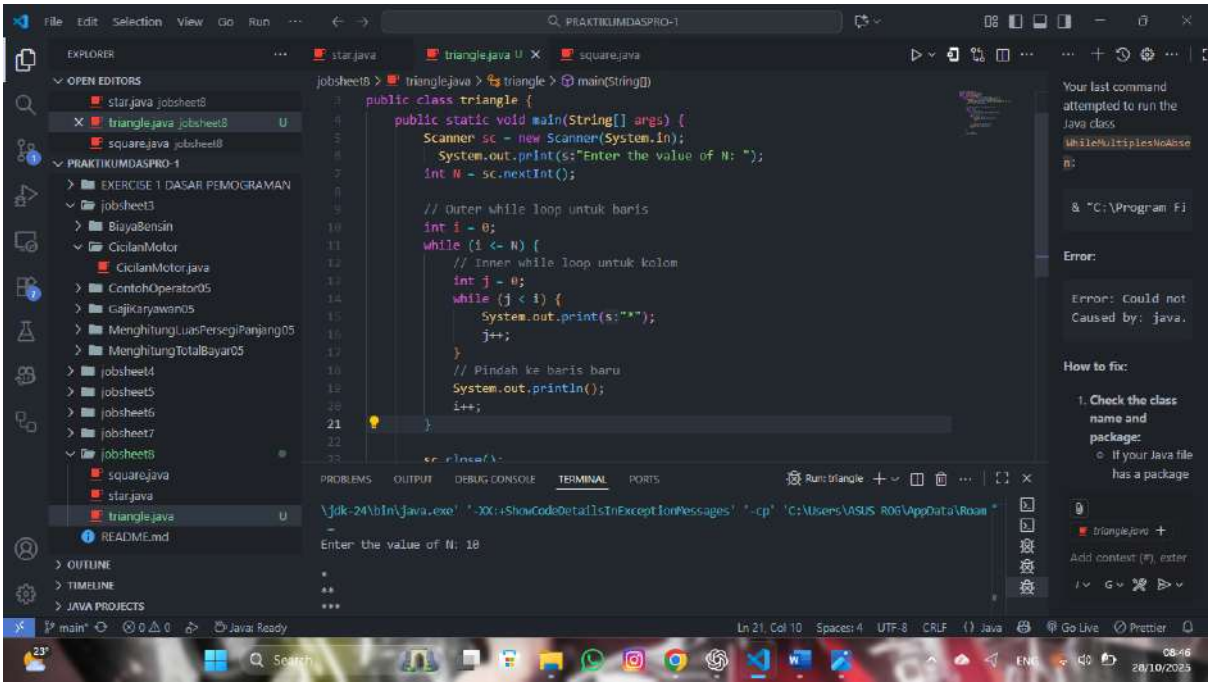
1. In Experiment 3, a right-angled triangle with a height of N was carried out. Suppose N is entered as 5, then the results are as follows:

```
*
**
***
****
*****
```

2. Create a new class with the name **Triangle** and save it in the **Triangle.java** file
3. Because the program requires input from the keyboard, it is necessary to import the **Scanner** class, so add the import syntax in the top line of the program.
4. Create a **main()** function, and fill in the following program code into the **main()** function

```
System.out.print("Enter the value of N: ");
int N = sc.nextInt();
int i = 0;
while (i <= N) {
    int j = 0;
    while (j < i) {
        System.out.print("*");
        j++;
    }
    i++;
}
```

5. Compile and run the program. Observe the results!



Questions!

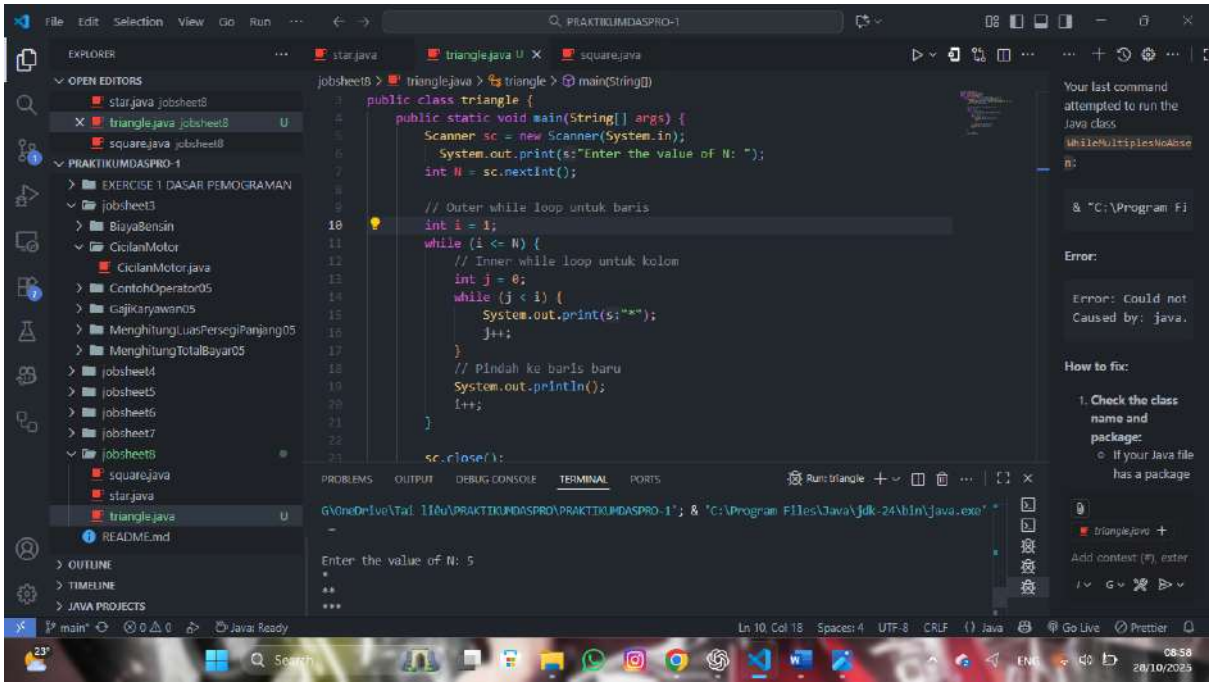
1. Look at the results, does the output produced with a value of N = 5 match the following display?

```
*
**
***
****
*****
```

IT IS NOT IN ACCORDANCE WITH! There is a blank line at the beginning that shouldn't be there.

2. If not, which parts should be improved or added? Describe any parts that need to be improved or added!

Perbaikan yang dilakukan: Mengubah inisialisasi `i = 0` menjadi `i = 1`



2.4 Experiment 4: Guess the Number Quiz

1. Create a new class with the name **Quiz** and save it in the **Quiz.java** file 2. Add Scanner and Random libraries in the top line of the program

```
import java.util.Scanner;
import java.util.Random;
```

3. Create a **main()** function

4. In the **main()** function, make a **Scanner** declaration with the name **input** and **Random** declaration with the name **rand**. In this case, Random is used to randomize the numbers

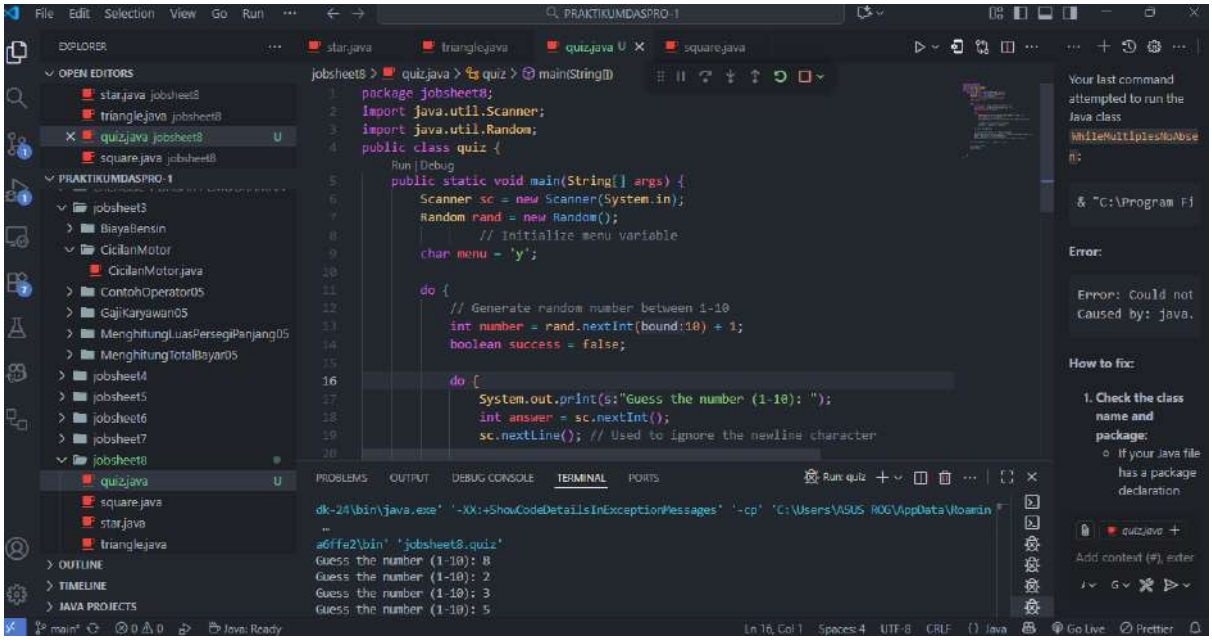
```
Scanner input = new Scanner(System.in);
Random rand = new Random();
```

5. Then on the next line, add the following syntax

```
char menu = 'y';
do {
    int number = rand.nextInt(10) + 1;
    boolean success = false;
    do {
        System.out.print("Guess the number (1-10): ");
        int answer = input.nextInt();
        input.nextLine();
        success = (answer == number);
    } while (!success);
    System.out.print("Do you want to repeat the game (Y/N)");
    menu = input.next().charAt(0);
    input.nextLine();
} while (menu == 'Y' || menu == 'y');
```

Note: the **input.nextLine()** syntax in that snippet is used to ignore the new line character

6. Compile and run the program. Observe the results!



Questions!

1. Explain the program flow in Experiment 4!

   import java.util.Scanner;

   import java.util.Random;

- **Scanner** is used to read input from the keyboard.
- **Random** is used to generate random numbers.

```
public class Quiz {
  public static void main(String[] args) {
```

- `input` is used to receive user input.
- `rand` is used to generate random numbers between 1 and 10.

```
char menu = 'y';
```

The variable menu is used to store the user's choice — whether they want to play again or not.

```
do {
  ...
} while (menu == 'Y' || menu == 'y');
```

- This loop makes the game **repeat** as long as the user enters **Y or y**.
- If the user enters **N or n**, the game stops.

```
int number = rand.nextInt(10) + 1;
```

This line generates a random number between **1 and 10**.
(`nextInt(10)` generates numbers 0–9, so adding 1 shifts the range to 1–10.)

```
boolean success = false;
```

This variable is used to mark whether the user's guess is correct **or** not**.**

```
do {
System.out.print("Guess the number (1-10): ");
int answer = input.nextInt();
input.nextLine(); // ignore newline character


success = (answer == number);
} while (!success);
```

- The program asks the user to enter a guess (`answer`).

- It checks if the guess matches the random number (`number`).
- If it's **incorrect**, the loop repeats.
- If it's **correct**, the inner loop ends.

```
System.out.print("Do you want to repeat the game (Y/N)? ");
menu = input.next().charAt(0);
```

- After guessing correctly, the program asks the user if they want to play again.
- If the user enters **Y/y**, the game restarts.
- If the user enters **N/n**, the game ends.

```
input.close();
```

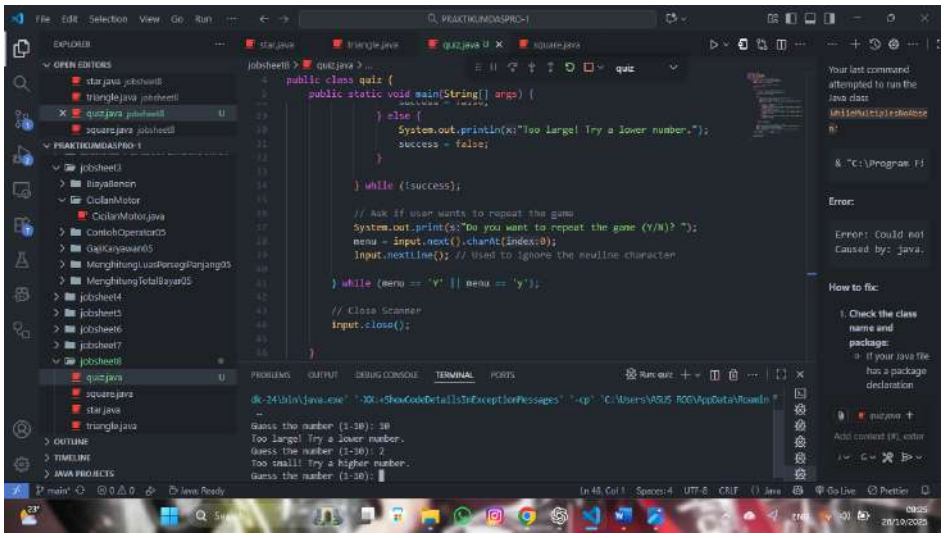
This closes the `Scanner` object to free up system resources.

2. What must be done to discontinue (not repeat) the game?

   Type "N" or "n" when asked "Do you want to repeat the game (Y/N)?", or

   Remove the outer loop so the program runs only once.

3. Modify the program above, so that it can display information about: input the guess value entered by the user, whether it is smaller or greater than the answer (number) randomly determined by the computer!



4. Commit and push the changes to GitHub

   https://github.com/rizkydava1233-ops/PRAKTIKUMDASPRO/blob/main/jobsheet8/square.java

2.5 Experiment 5: Filling and Displaying Arrays

1. Create a new class with the name **NestedLoopStudentID**
2. Create a **main()** function
3. In the **main()** function, add a declaration for Scanner named **scanner** and a 2dimensional array declaration named **temps** with 5 rows and 7 columns of type **double**
4. Add the following program code

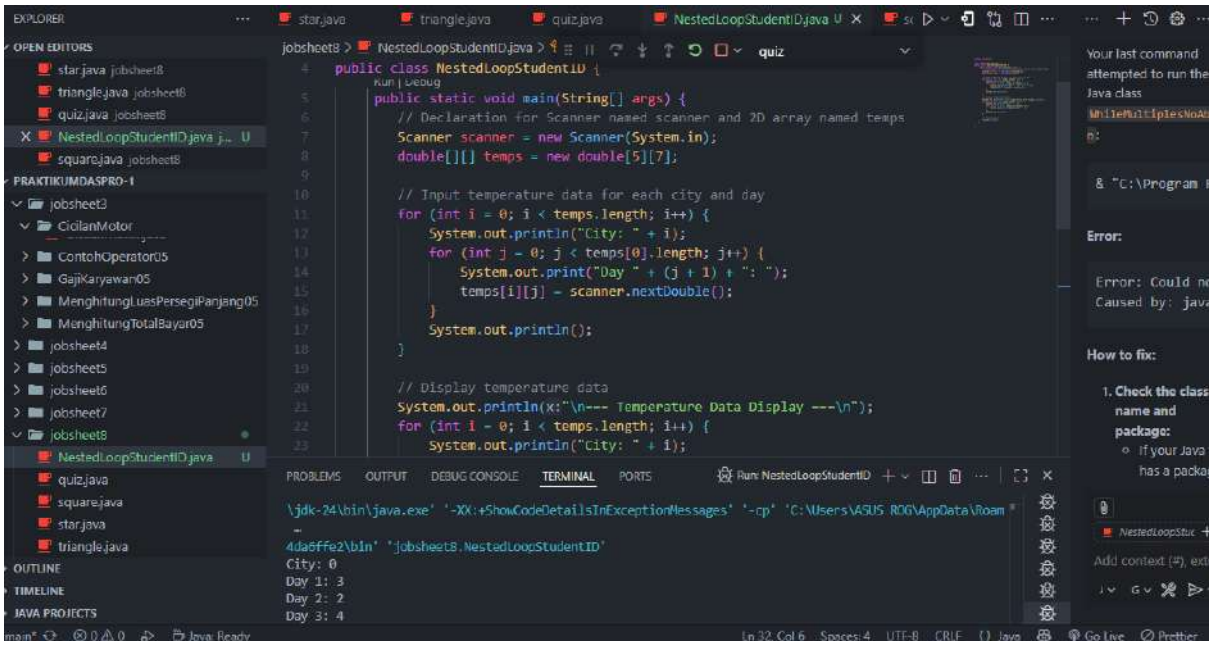
```
for (int i = 0; i < temps.length; i++) {
    System.out.println("City: " + i);
    for (int j = 0; j < temps[0].length; j++) {
        System.out.print("Day " + (j + 1) + ": ");
        temps[i][j] = scanner.nextDouble();
    }
    System.out.println();
}
```

5. Then, also add the following several lines of program code

```
for (int i = 0; i < temps.length; i++) {
    System.out.println("City: " + i);
    for (int j = 0; j < temps[0].length; j++) {
        System.out.print(temps[i][j] + " ");
    }
    System.out.println();
}
```

6. Compile and run the program. Observe the results!

Note: *This program example is a case example for storing temperature data in a city. Check back to last week's material slides.*



Questions!

1. Explain the program flow in Experiment 5!

double[][] temps = new double[5][7];

- The program creates a 2D array named `temps` with **5 rows (cities)** and **7 columns (days)**.
- Each element stores temperature data in **double** format.

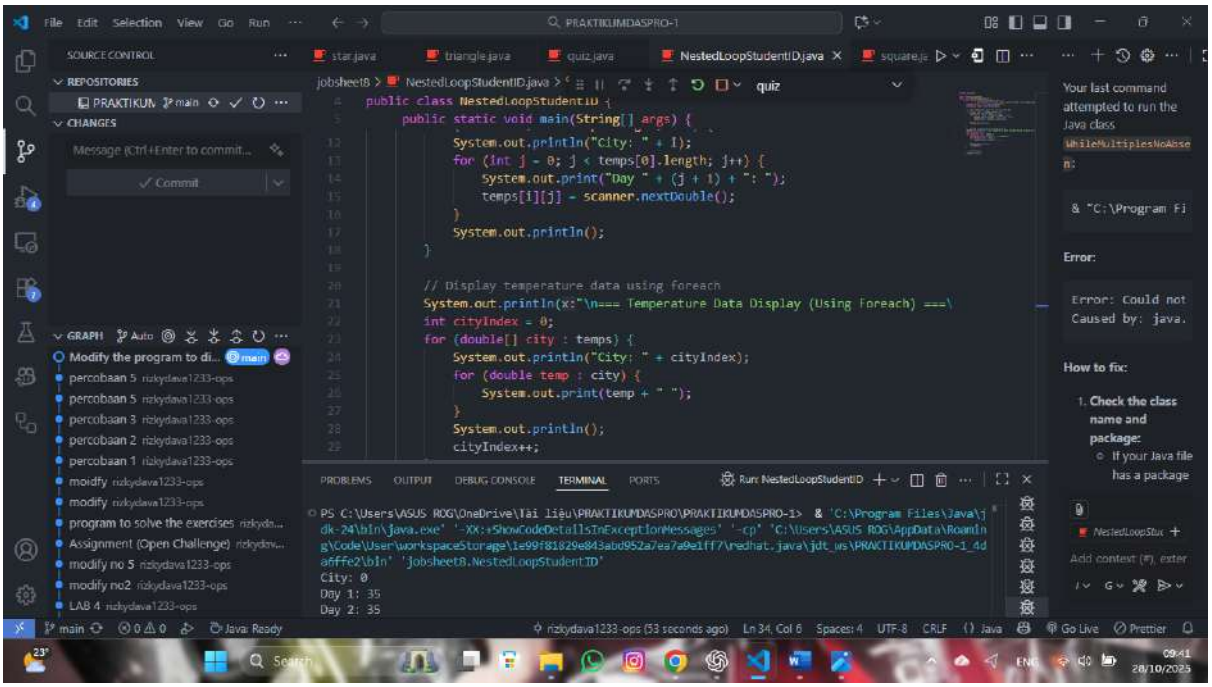
for (int i = 0; i < temps.length; i++) { ... }

- The **outer loop (i)** represents each **city**.
- The **inner loop (j)** represents each **day** for that city.
- The user is asked to enter temperature values one by one.

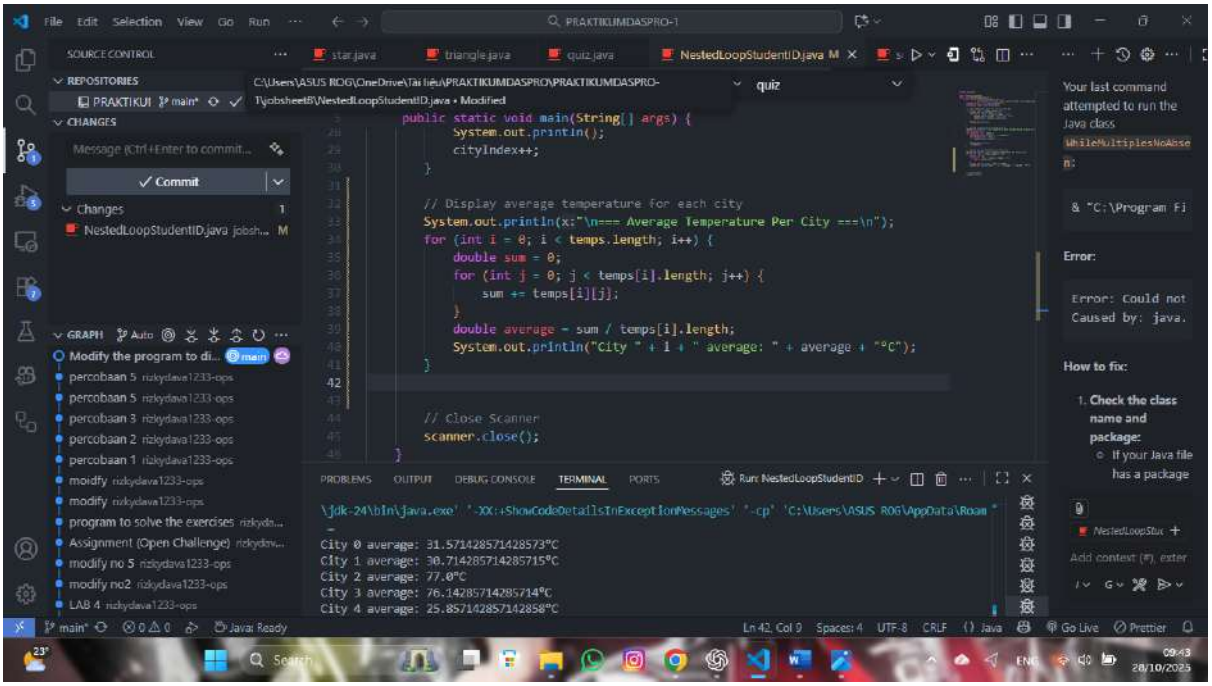
for (int i = 0; i < temps.length; i++) { ... }

- The outer loop displays the **city name/number**.
- The inner loop prints all **temperature values for each day** in that city.
- After finishing one city, it moves to a new line.

2. Modify the program to display an array using foreach!



3. Modify the program so that it can display the average value for each city!



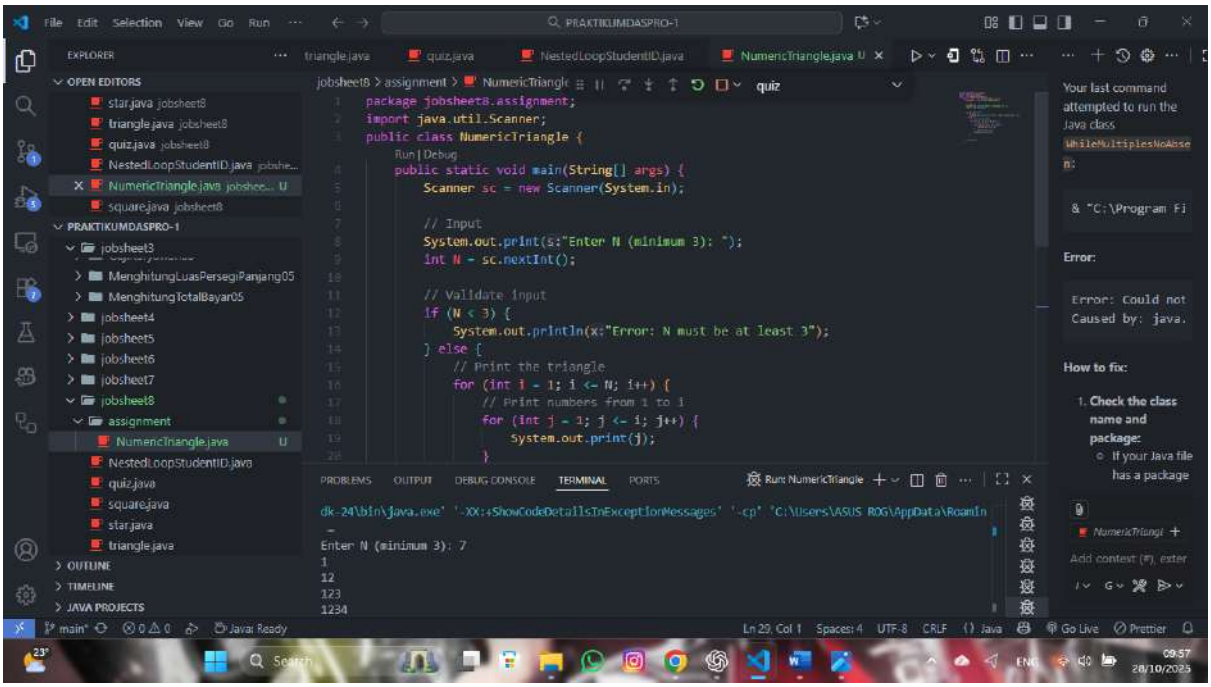
4. Commit and push the changes to GitHub

https://github.com/rizkydava1233-ops/PRAKTIKUMDASPRO/blob/main/jobsheet8/NestedLoopStudentID.java

3. Assignment

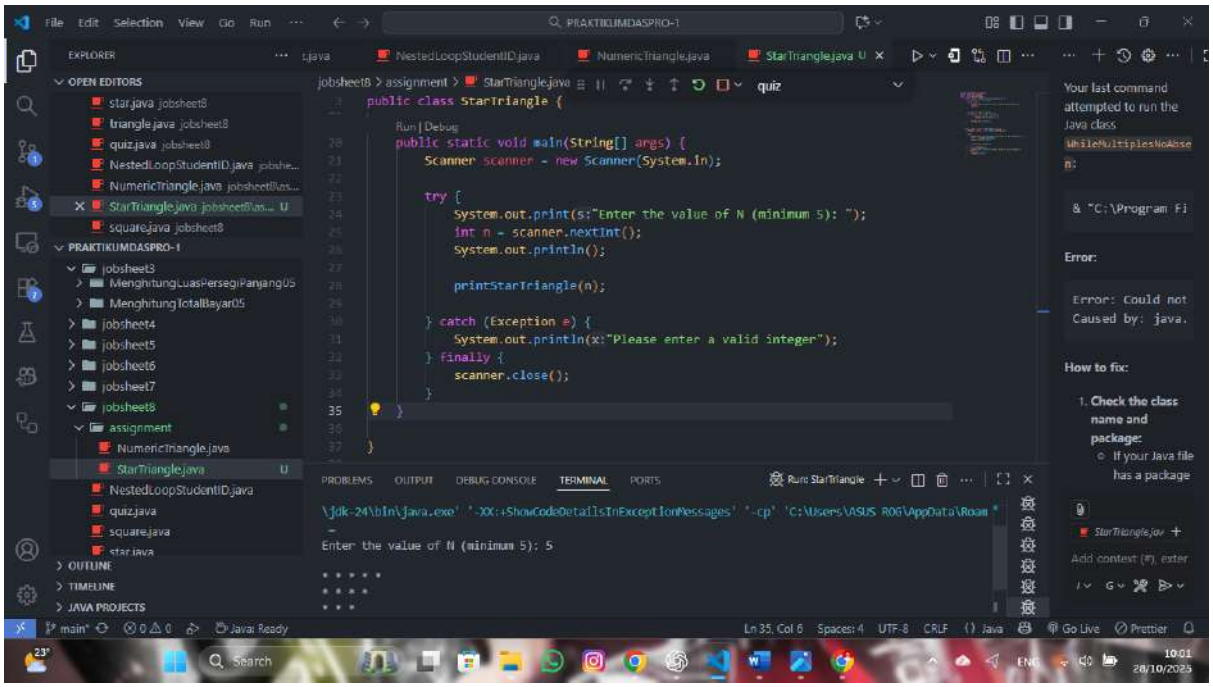
1. Create a program to print a numeric triangle display as below based on the N input (minimum N value is 3). Example N = 5

```
    1
   12
  123
 1234
12345
```



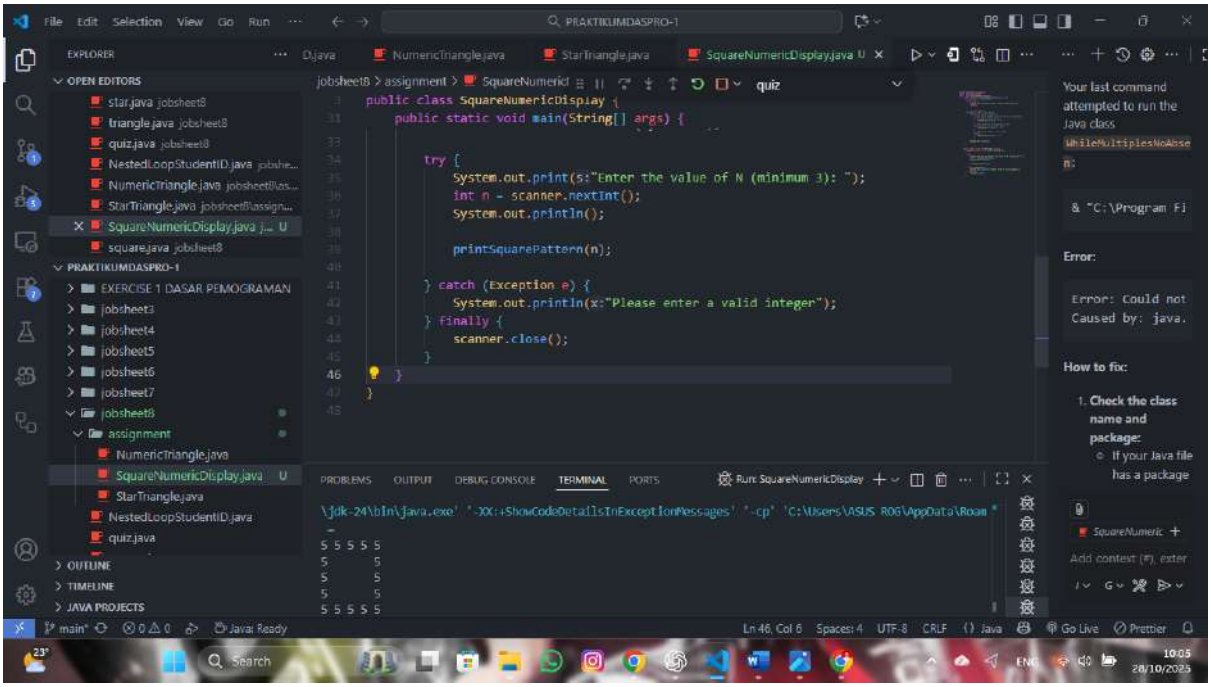
2. Create a program to print the star triangle view shown below based on the N input (minimum N value is 5). Example N = 7

```
*******
******
*****
****
***
**
*
```

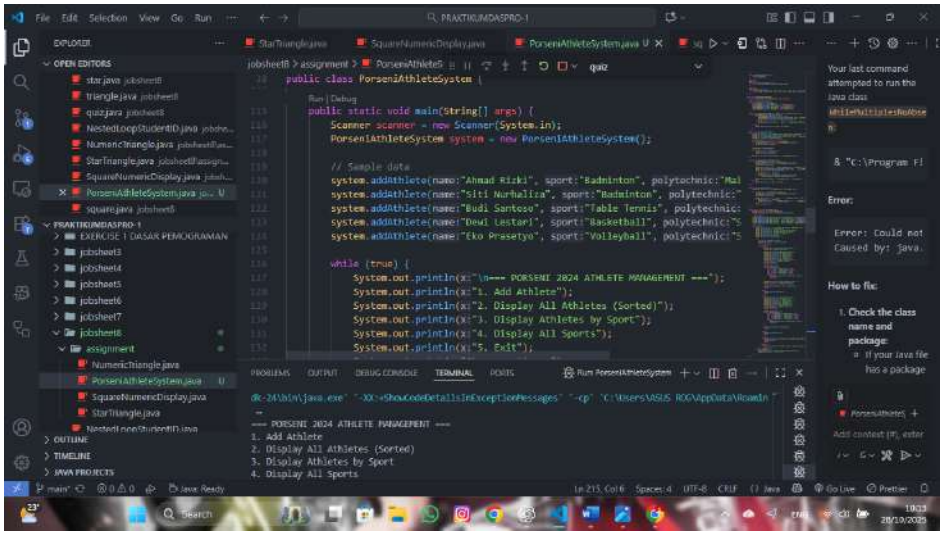


3. Create a program to print a square numeric display like the one below based on N input (minimum N value is 3). Example N = 3 and N = 5

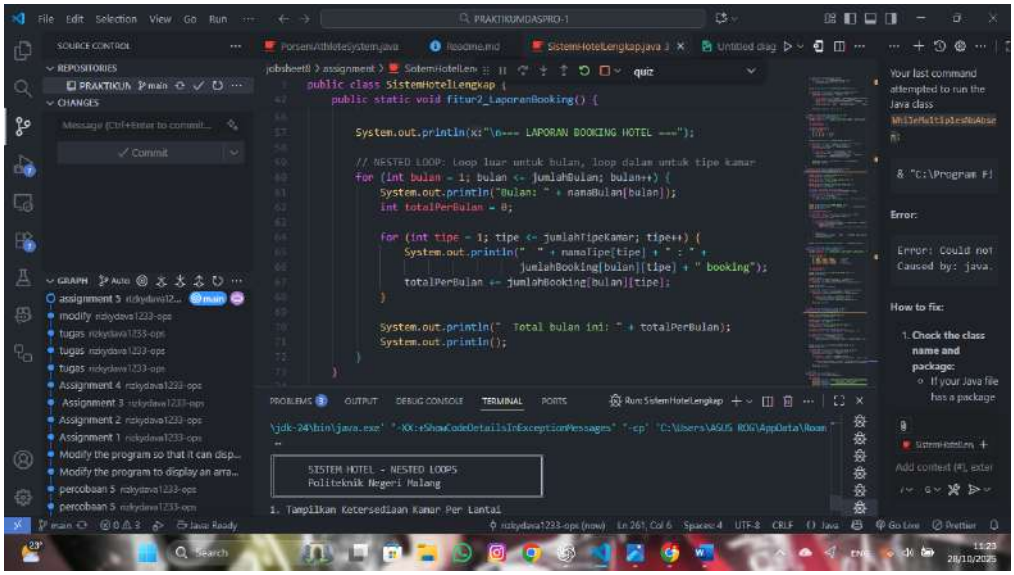
```
                5 5 5 5 5
                5       5
3 3 3           5       5
3   3           5       5
3 3 3           5 5 5 5 5
```



4. In 2024, Malang State Polytechnic will host the Porseni national event. There are several sports that are competed in, such as **badminton**, **table tennis**, **basketball**, and **volleyball**. Each sport sends its **5 best athletes** from all polytechnics throughout Indonesia to take part in this biannual event. Create a data storage to display information on the **names of athletes** from the various branches mentioned in **ascending** order.



5. Implement the flowchart of the features you created in the previous theory assignment about nested loops!



6. Don't forget, all program code must be pushed to your repository.

   https://github.com/rizkydava1233-ops/PRAKTIKUMDASPRO/blob/main/jobsheet8/assignment/SistemHotelLengkap.javass


POLITEKNIK NEGERI MALANG