OODP workshop 4

1. What are different types of loops in java? Explain each of them with proper syntax.
2. Write a program to ask the user to enter a number and print count down from that number to 0.
3. Write a program to ask user to enter the temperature of seven days and then find out the average temperature and print the results.
4. Write a program to ask user a value and make sure that value is between 3 and 6 inclusive. Keep asking the user until user enters a valid value and display a success message and print error when user enter invalid value.

1. Ans:

Different types of loops in Java with explanations and proper syntax:

1. for loop: The for loop is the most commonly used loop in Java. It is used to iterate over a block of code a specified number of times. It has three parts: initialization, condition, and iteration.

Syntax:

```
for (initialization; condition; iteration) {
  // code to be executed
}
```

Example:

```
for (int i = 0; i < 5; i++) {
  System.out.println("Value of i: " + i);
}
```

2. while loop: The while loop is used to execute a block of code repeatedly as long as a given condition is true.

Syntax:

```
while (condition) {
  // code to be executed
}
```

Example:

```
int count = 0;
while (count < 5) {
  System.out.println("Count: " + count);
  count++;
}
```

3. do-while loop: The do-while loop is similar to the while loop, but it guarantees that the code block will execute at least once before checking the condition.

Syntax:

```
do {
  // code to be executed
} while (condition);
```

Example:

```
int num = 10;
do {
  System.out.println("Number: " + num);
  num++;
} while (num < 5);
```

4. for-each loop (Enhanced for loop): The for-each loop is used to iterate over the elements of an array or a collection (like ArrayList) in a compact and readable way.

Syntax:

```
for (dataType element : array/collection) {
   // code to be executed
}
```

Example:

```
int[] numbers = {1, 2, 3, 4, 5};
for (int num : numbers) {
   System.out.println("Number: " + num);
}
```

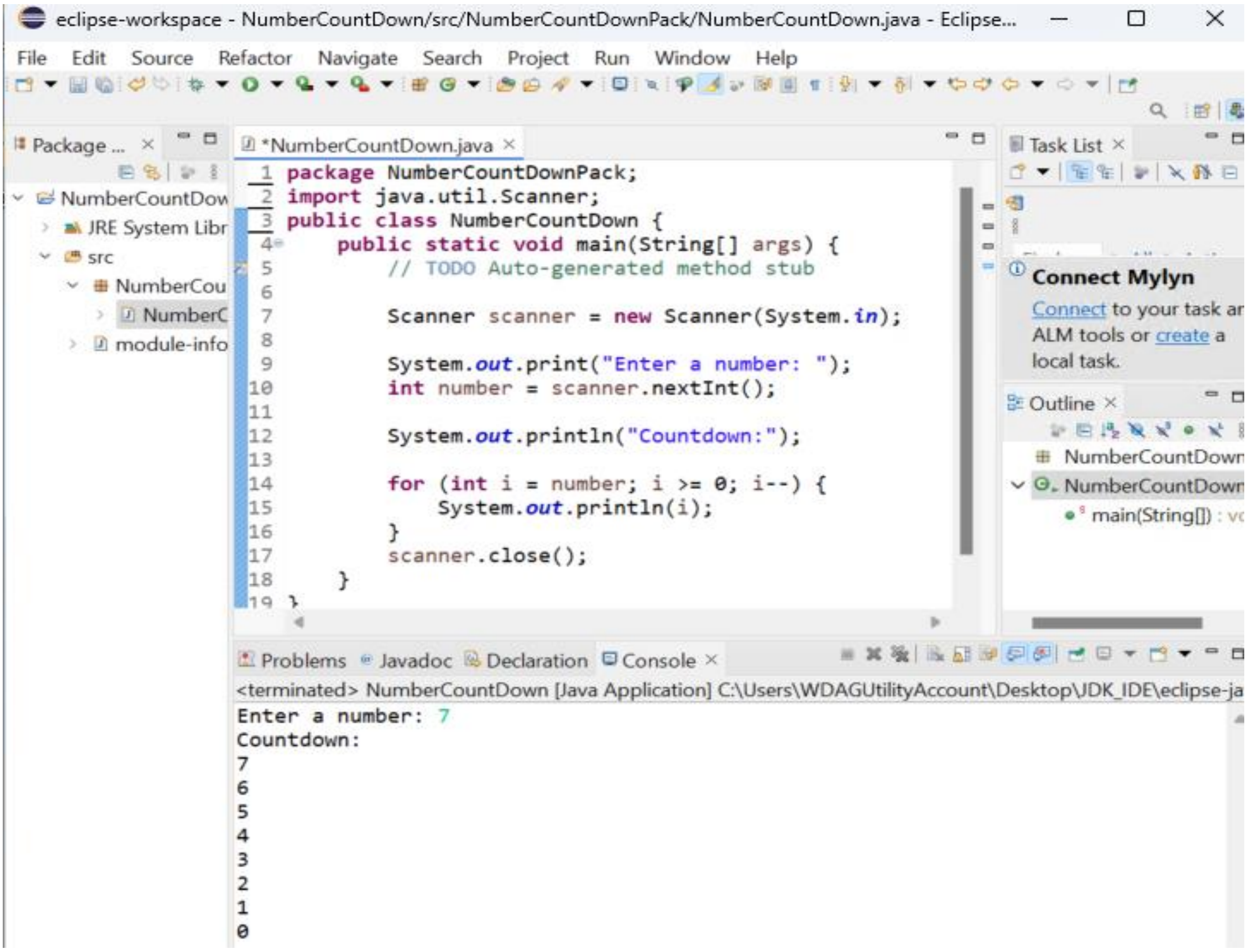
5. Nested loops: Loops can be nested within other loops in Java. This means that one loop can be placed inside the body of another loop. Nested loops are useful when you need to iterate over two or more collections or perform operations on multi-dimensional arrays.

Example:

```
for (int i = 1; i <= 3; i++) {
   for (int j = 1; j <= 3; j++) {
      System.out.println("i: " + i + ", j: " + j);
   }
}
```

These are the main types of loops in Java.

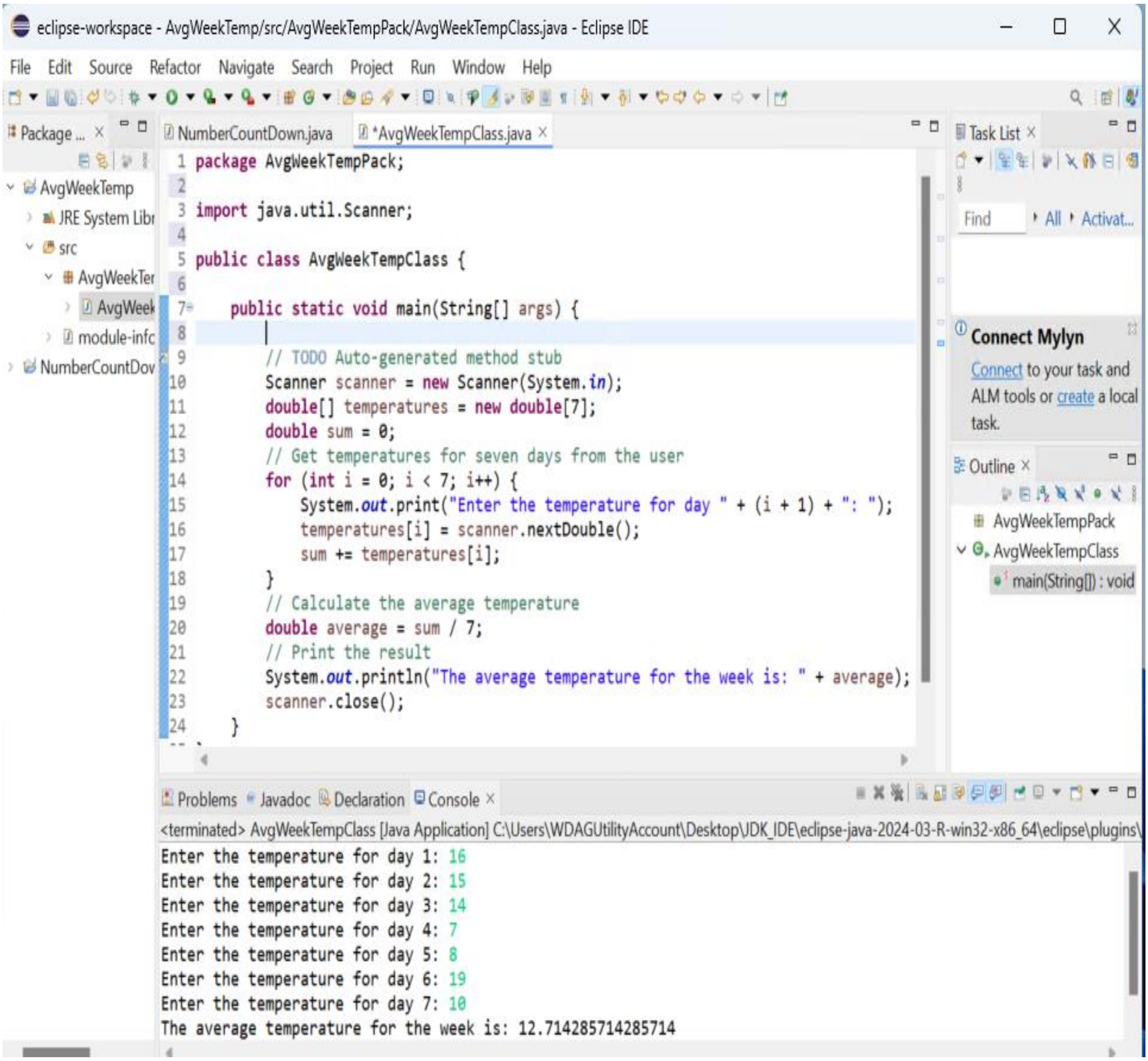
2. Ans: Count-Down Program


eclipse-workspace - NumberCountDown/src/NumberCountDownPack/NumberCountDown.java - Eclipse...
File Edit Source Refactor Navigate Search Project Run Window Help
Package ...
NumberCountDow
JRE System Libr
src
NumberCou
NumberC
module-info
*NumberCountDown.java
Task List
Connect Mylyn
Connect to your task and ALM tools or create a local task.
Outline
NumberCountDown
NumberCountDown
main(String[]) : vo
Problems Javadoc Declaration Console

```
package NumberCountDownPack;
import java.util.Scanner;
public class NumberCountDown {
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int number = scanner.nextInt();

        System.out.println("Countdown:");

        for (int i = number; i >= 0; i--) {
            System.out.println(i);
        }
        scanner.close();
    }
}
```

```
<terminated> NumberCountDown [Java Application] C:\Users\WDAGUtilityAccount\Desktop\JDK_IDE\eclipse-ja
Enter a number: 7
Countdown:
7
6
5
4
3
2
1
0
```

3.Ans:

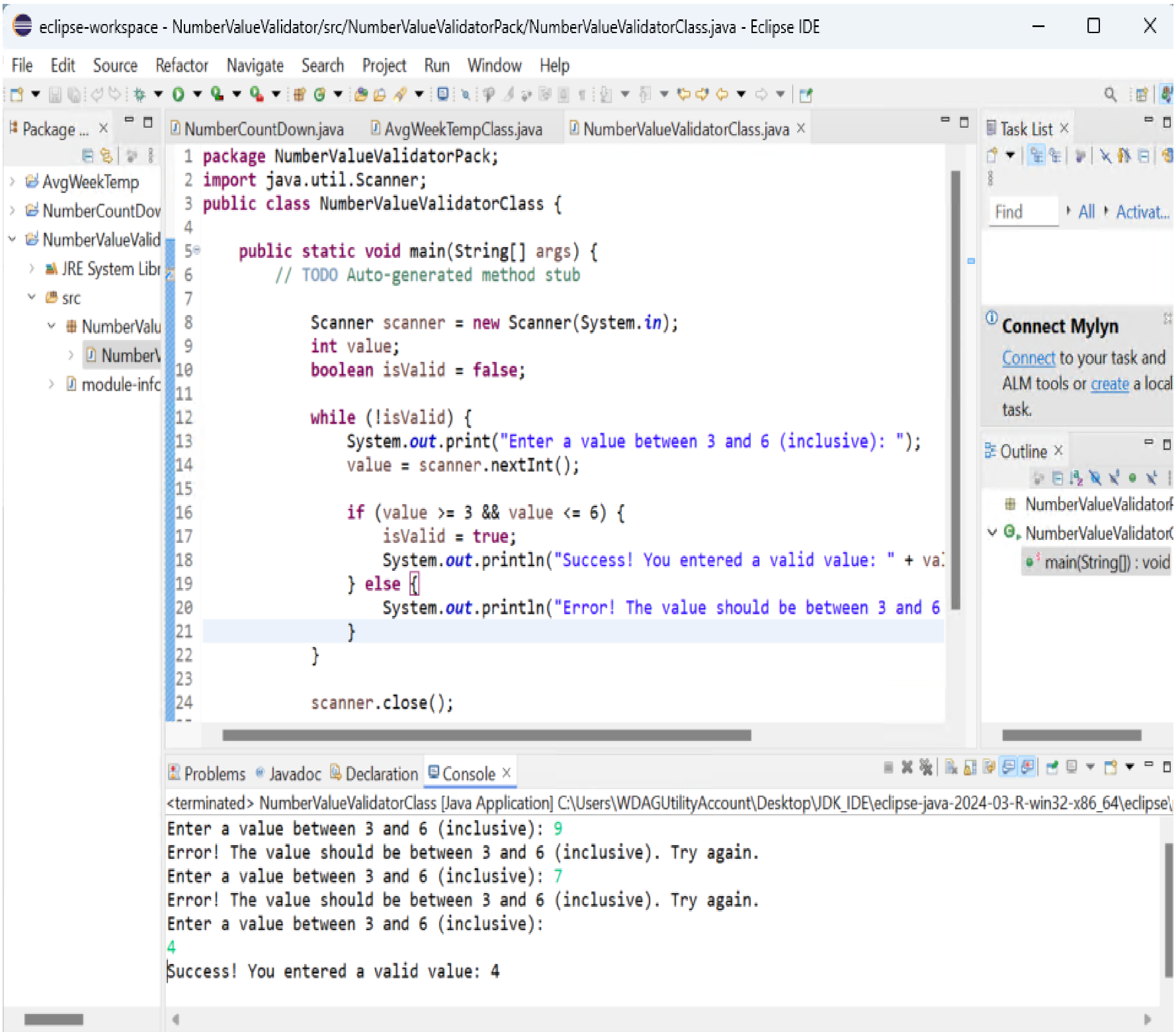


```java
package AvgWeekTempPack;

import java.util.Scanner;

public class AvgWeekTempClass {

    public static void main(String[] args) {

        // TODO Auto-generated method stub
        Scanner scanner = new Scanner(System.in);
        double[] temperatures = new double[7];
        double sum = 0;
        // Get temperatures for seven days from the user
        for (int i = 0; i < 7; i++) {
            System.out.print("Enter the temperature for day " + (i + 1) + ": ");
            temperatures[i] = scanner.nextDouble();
            sum += temperatures[i];
        }
        // Calculate the average temperature
        double average = sum / 7;
        // Print the result
        System.out.println("The average temperature for the week is: " + average);
        scanner.close();
    }
```

```
<terminated> AvgWeekTempClass [Java Application] C:\Users\WDAGUtilityAccount\Desktop\JDK_IDE\eclipse-java-2024-03-R-win32-x86_64\eclipse\plugins\
Enter the temperature for day 1: 16
Enter the temperature for day 2: 15
Enter the temperature for day 3: 14
Enter the temperature for day 4: 7
Enter the temperature for day 5: 8
Enter the temperature for day 6: 19
Enter the temperature for day 7: 10
The average temperature for the week is: 12.714285714285714
```

(Continue to page 4)

4.Ans



```
package NumberValueValidatorPack;
import java.util.Scanner;
public class NumberValueValidatorClass {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

            Scanner scanner = new Scanner(System.in);
            int value;
            boolean isValid = false;

            while (!isValid) {
                System.out.print("Enter a value between 3 and 6 (inclusive): ");
                value = scanner.nextInt();

                if (value >= 3 && value <= 6) {
                    isValid = true;
                    System.out.println("Success! You entered a valid value: " + va
                } else {
                    System.out.println("Error! The value should be between 3 and 6
                }
            }

            scanner.close();
```

```
<terminated> NumberValueValidatorClass [Java Application] C:\Users\WDAGUtilityAccount\Desktop\JDK_IDE\eclipse-java-2024-03-R-win32-x86_64\eclipse\
Enter a value between 3 and 6 (inclusive): 9
Error! The value should be between 3 and 6 (inclusive). Try again.
Enter a value between 3 and 6 (inclusive): 7
Error! The value should be between 3 and 6 (inclusive). Try again.
Enter a value between 3 and 6 (inclusive):
4
Success! You entered a valid value: 4
```