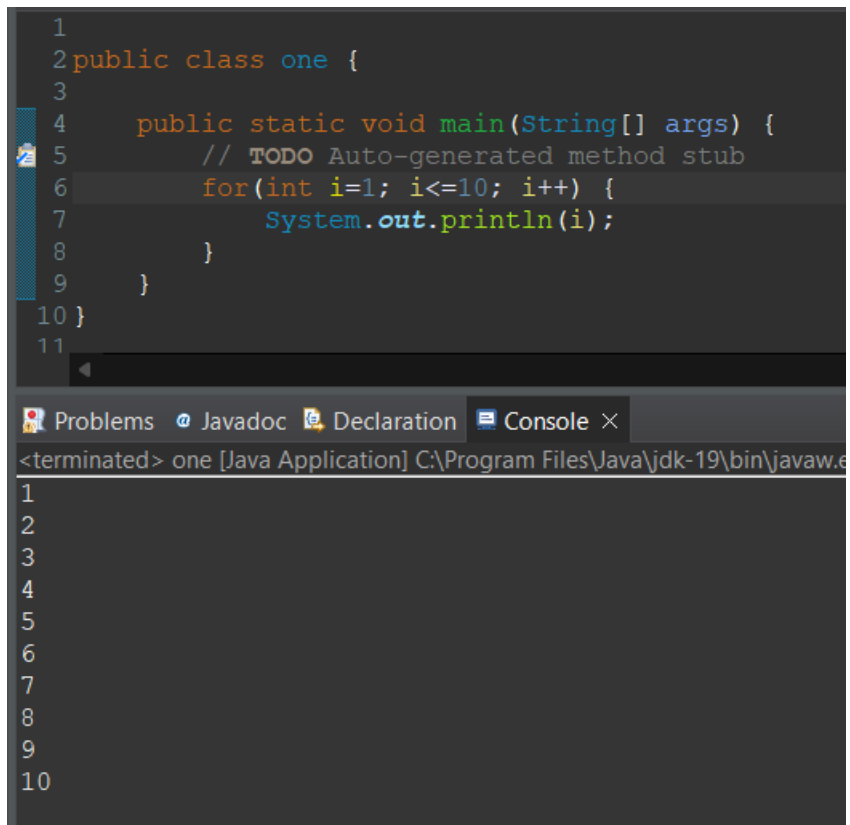


Looping:

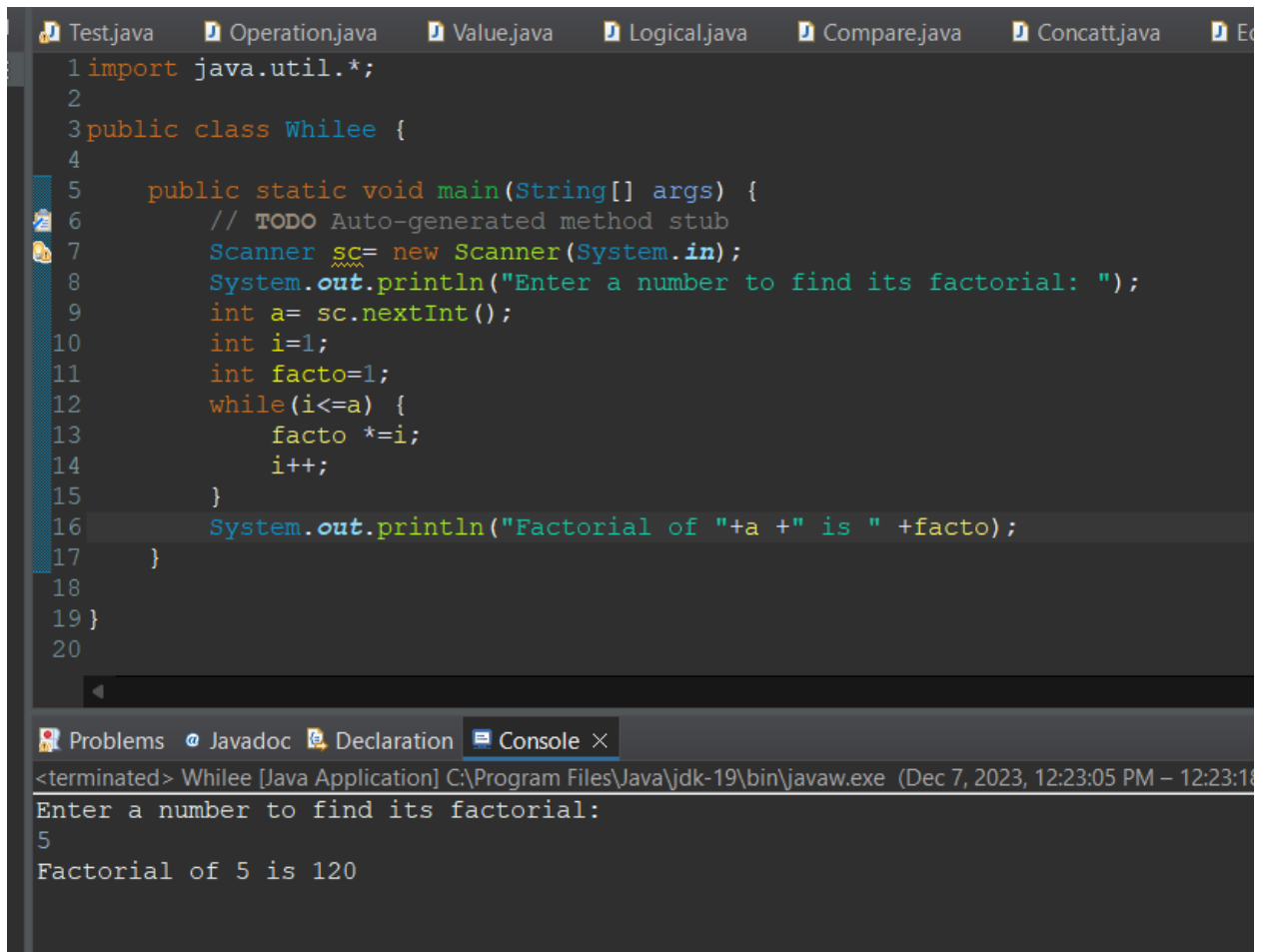
1. Write a Java program that uses a "for" loop to print the numbers from 1 to 10.



```
1
2 public class one {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         for(int i=1; i<=10; i++) {
7             System.out.println(i);
8         }
9     }
10 }
11
```

The screenshot shows an IDE with a Java file named 'one'. The code uses a 'for' loop to print numbers from 1 to 10. The console output at the bottom shows the numbers 1 through 10, each on a new line.

2. Implement a Java program that utilizes a "while" loop to find the factorial of a given number.



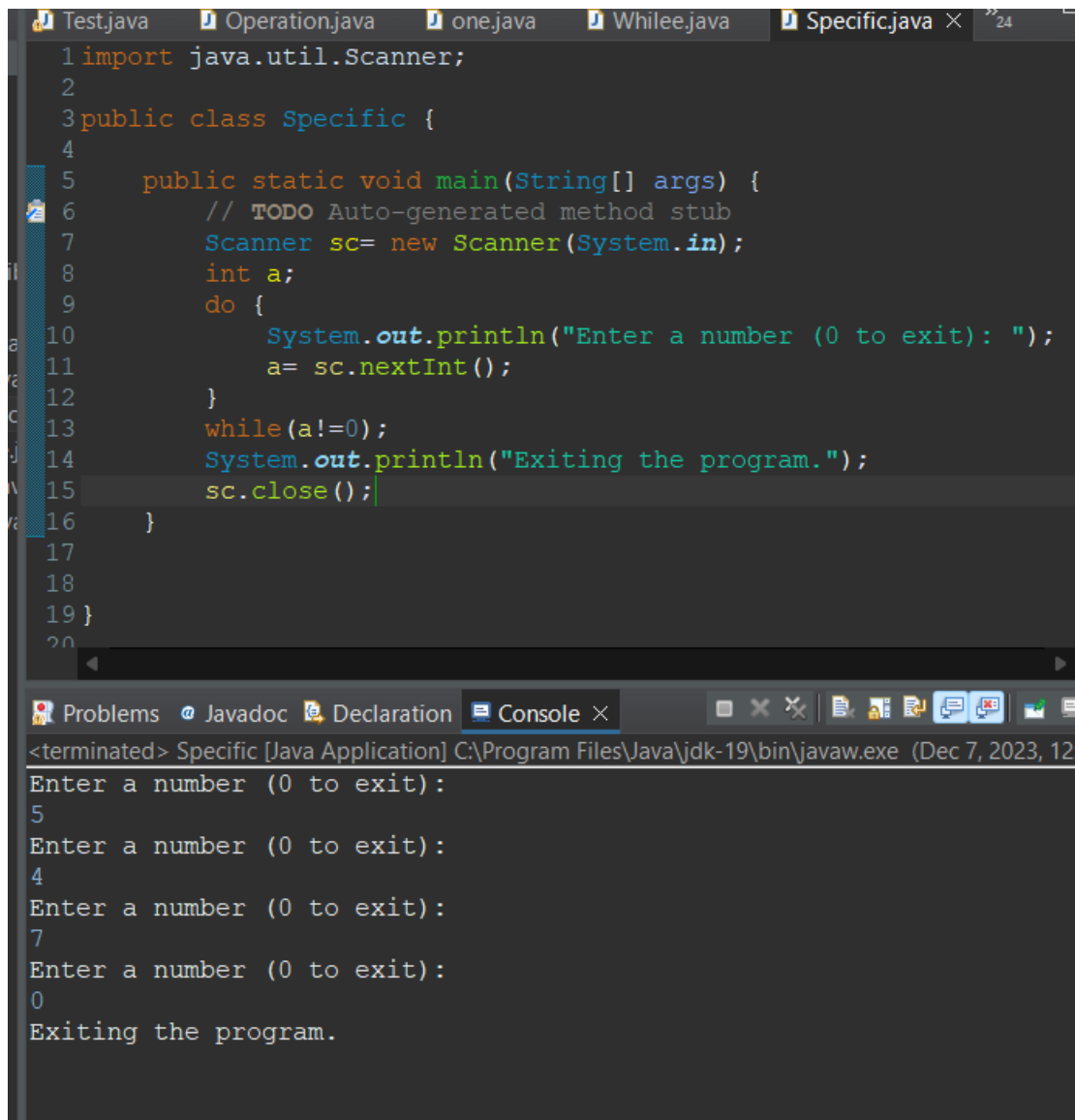
```
Test.java  Operation.java  Value.java  Logical.java  Compare.java  Concat.java  Ed
1 import java.util.*;
2
3 public class Whilee {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         Scanner sc= new Scanner(System.in);
8         System.out.println("Enter a number to find its factorial: ");
9         int a= sc.nextInt();
10        int i=1;
11        int facto=1;
12        while(i<=a) {
13            facto *=i;
14            i++;
15        }
16        System.out.println("Factorial of "+a +" is " +facto);
17    }
18
19 }
20
```

Problems Javadoc Declaration Console X

<terminated> Whilee [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 7, 2023, 12:23:05 PM – 12:23:14)

Enter a number to find its factorial:
5
Factorial of 5 is 120

3. Create a Java program using a "do-while" loop to repeatedly ask the user for input until they enter a specific value (e.g., 0).



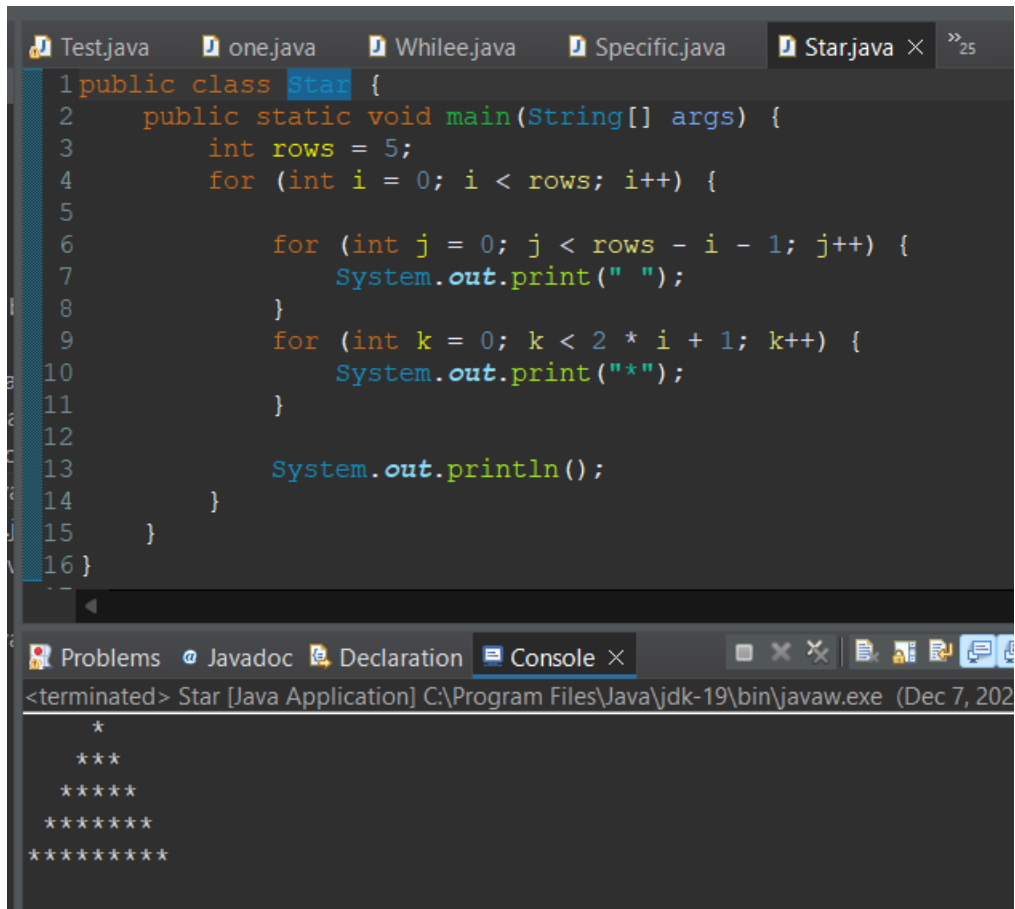
The screenshot shows an IDE with several tabs: Test.java, Operation.java, one.java, Whilee.java, and Specific.java. The 'Specific.java' tab is active, displaying the following code:

```
1 import java.util.Scanner;
2
3 public class Specific {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         Scanner sc= new Scanner(System.in);
8         int a;
9         do {
10             System.out.println("Enter a number (0 to exit): ");
11             a= sc.nextInt();
12         }
13         while(a!=0);
14         System.out.println("Exiting the program.");
15         sc.close();
16     }
17
18
19 }
20
```

Below the code editor is a console window with the following output:

```
<terminated> Specific [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 7, 2023, 12:
Enter a number (0 to exit):
5
Enter a number (0 to exit):
4
Enter a number (0 to exit):
7
Enter a number (0 to exit):
0
Exiting the program.
```

4. Write a Java program that demonstrates the use of nested loops to print a pattern, such as a pyramid of stars.



The screenshot shows an IDE with a Java file named `Star.java`. The code defines a `public class Star` with a `main` method. Inside `main`, it sets `rows = 5` and uses nested loops to print a pattern of stars. The first loop iterates over rows, and the second loop iterates over columns, printing spaces followed by stars. The output in the console shows a pattern of 5 rows of stars, with the number of stars increasing from 1 to 5 in each row.

```
1 public class Star {
2     public static void main(String[] args) {
3         int rows = 5;
4         for (int i = 0; i < rows; i++) {
5
6             for (int j = 0; j < rows - i - 1; j++) {
7                 System.out.print(" ");
8             }
9             for (int k = 0; k < 2 * i + 1; k++) {
10                 System.out.print("*");
11             }
12
13             System.out.println();
14         }
15     }
16 }
```

Console output:

```
<terminated> Star [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 7, 202
*
***
*****
*****
*****
```

Arrays:

5. Develop a Java program that declares and initializes an array of integers. Print the elements of the array in reverse order.

```
1
2 public class Reverse {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         int arr[] = {1,2,3,4,5,6};
7         int l= arr.length;
8
9         System.out.println("Array in reverse order: ");
10        for(int i=l-1 ;i>=0; i--) {
11            System.out.print(arr[i]+" ");
12        }
13
14    }
15
16 }
```

Problems Javadoc Declaration Console ×

<terminated> Reverse [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 7

Array in reverse order:
6 5 4 3 2 1

6. Implement a Java program that finds the sum and average of elements in an array of floating-point numbers.

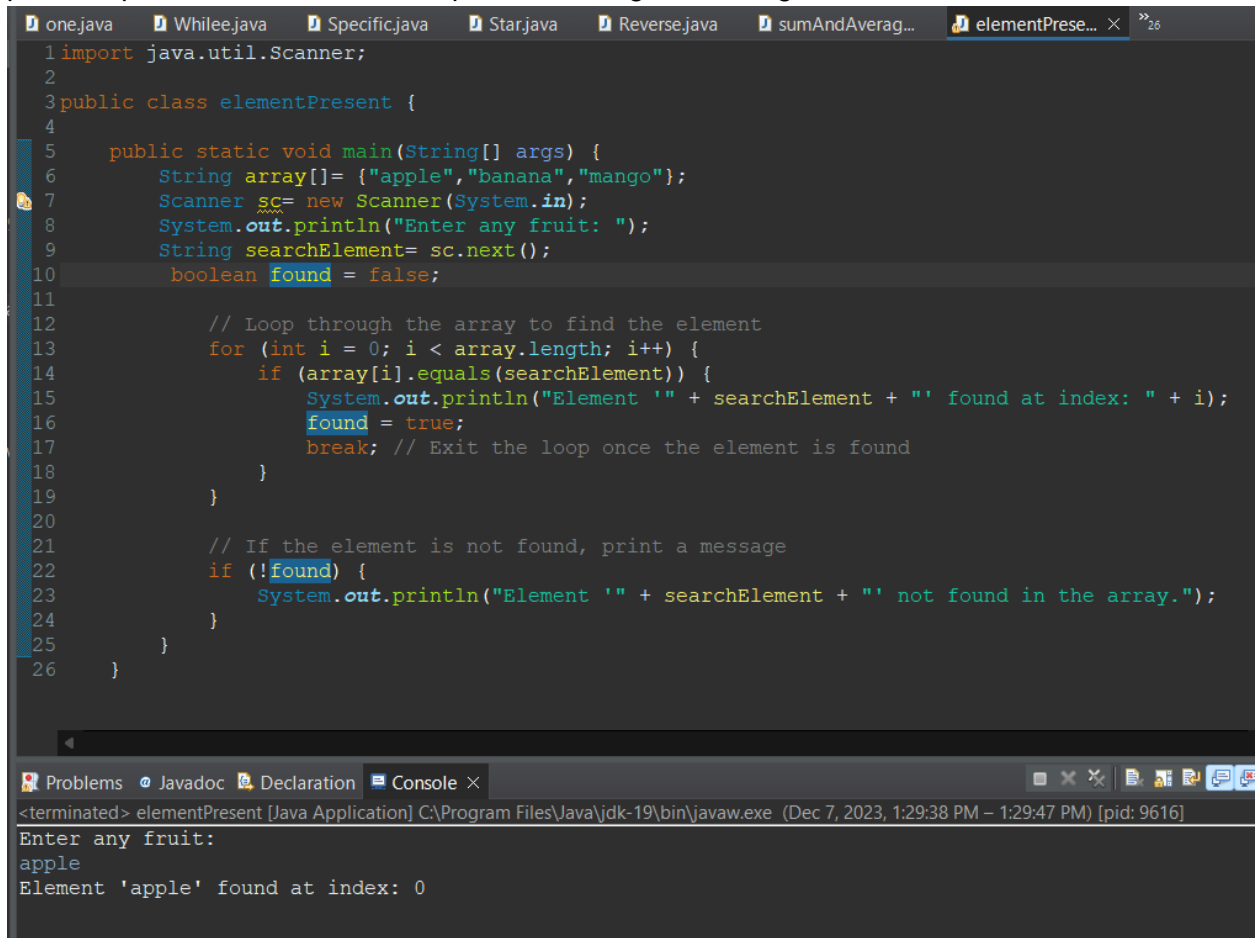
```
one.java Star.java Reverse.java sumAndAverag... × 28
1
2 public class sumAndAverage {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         float arr[] = {1,2,3,4,5};
7         float sum=0;
8         for(int i=0; i<=arr.length; i++) {
9             sum= sum+ arr[i];
10        }
11        System.out.println("Sum: "+sum);
12
13        double avg= sum/arr.length;
14        System.out.println("Average: "+avg);
15    }
16 }
```

Problems Javadoc Declaration Console ×

<terminated> sumAndAverage [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe

Sum: 12.0
Average: 2.4000000953674316

7. Write a Java program that checks if a given element is present in an array of strings. If present, print its index; otherwise, print a message indicating its absence.



The screenshot shows an IDE with a Java file named 'elementPresent.java'. The code defines a class 'elementPresent' with a 'main' method. It imports 'java.util.Scanner', creates an array of strings ['apple', 'banana', 'mango'], and uses a 'Scanner' to take user input. A loop checks each element in the array against the input. If found, it prints the index and sets a 'found' flag to true. If not found after the loop, it prints a message. The console output shows the user entered 'apple' and the program printed 'Element 'apple' found at index: 0'.

```
1 import java.util.Scanner;
2
3 public class elementPresent {
4
5     public static void main(String[] args) {
6         String array[] = {"apple", "banana", "mango"};
7         Scanner sc = new Scanner(System.in);
8         System.out.println("Enter any fruit: ");
9         String searchElement = sc.next();
10        boolean found = false;
11
12        // Loop through the array to find the element
13        for (int i = 0; i < array.length; i++) {
14            if (array[i].equals(searchElement)) {
15                System.out.println("Element '" + searchElement + "' found at index: " + i);
16                found = true;
17                break; // Exit the loop once the element is found
18            }
19        }
20
21        // If the element is not found, print a message
22        if (!found) {
23            System.out.println("Element '" + searchElement + "' not found in the array.");
24        }
25    }
26 }
```

Problems Javadoc Declaration Console X

<terminated> elementPresent [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 7, 2023, 1:29:38 PM – 1:29:47 PM) [pid: 9616]

Enter any fruit:
apple
Element 'apple' found at index: 0

8. Create a Java program that sorts an array of integers in ascending order using the bubble sort algorithm.

```
2
3 public class Bubblesort {
4
5     public static void bubbleSortAlgorithm(int[] arr) {
6         int size = arr.length;
7
8         for (int i = 0; i < size - 1; i++) {
9             for (int j = 0; j < size - i - 1; j++) {
10                 if (arr[j] > arr[j + 1]) {
11                     int temp = arr[j];
12                     arr[j] = arr[j + 1];
13                     arr[j + 1] = temp;
14                 }
15             }
16         }
17     }
18
19     public static void main(String[] args) {
20         int[] array = {100, 87, 56, 12, 10};
21
22         // Call the bubbleSortAlgorithm method to sort the array
23         bubbleSortAlgorithm(array);
24
25         // Print the sorted array
26         System.out.print("The array after performing the bubble sort algorithm is: ");
27         System.out.println(Arrays.toString(array));
28     }
29 }
```

Problems Javadoc Declaration Console X

<terminated> Bubblesort [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 7, 2023, 1:35:15 PM – 1:35:15 PM) [pid: 17184]
The array after performing the bubble sort algorithm is: [10, 12, 56, 87, 100]

Enhanced For Loop:

9. Develop a Java program that uses the enhanced for loop to iterate through an array of characters and count the number of vowels.

```
1 public class Characters {
2     public static void main(String[] args) {
3         char[] characters = {'a', 'e', 'q', 't', 'i', 'o', 'p'};
4         int count = 0;
5         for (char c : characters) {
6             if ('a' == c || 'e' == c || 'i' == c || 'o' == c || 'u' == c ||
7                 'A' == c || 'E' == c || 'I' == c || 'O' == c || 'U' == c) {
8                 count++;
9             }
10        }
11        System.out.println("The number of vowels in the array is: " + count);
12    }
13 }
14
```

Problems Javadoc Declaration Console ×

<terminated> Characters [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 7, 2023, 1:46:48 PM – 1:46:48 PM)

The number of vowels in the array is: 4

10. Write a Java program that uses the enhanced for loop to find the maximum value in an array of doubles.

```
sumAndAverag... elementPrese... Bubblesort.java Characters.java maxinarr.java ×
```

```
1 public class maxinarr {
2     public static void main(String[] args) {
3         double[] numbers = {3.5, 2.1, 8.9, 4.7, 1.2};
4         double max = numbers[0];
5         for (double num : numbers) {
6             if (num > max) {
7                 max = num;
8             }
9         }
10        System.out.println("The maximum value in the array is: " + max);
11    }
12 }
13
```

Problems Javadoc Declaration Console ×

<terminated> maxinarr [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 7, 2023, 1:56:31 PM – 1:56:31 PM)

The maximum value in the array is: 8.9

11. Implement a Java program that initializes a 2D array and uses an enhanced for loop to calculate the sum of all elements.

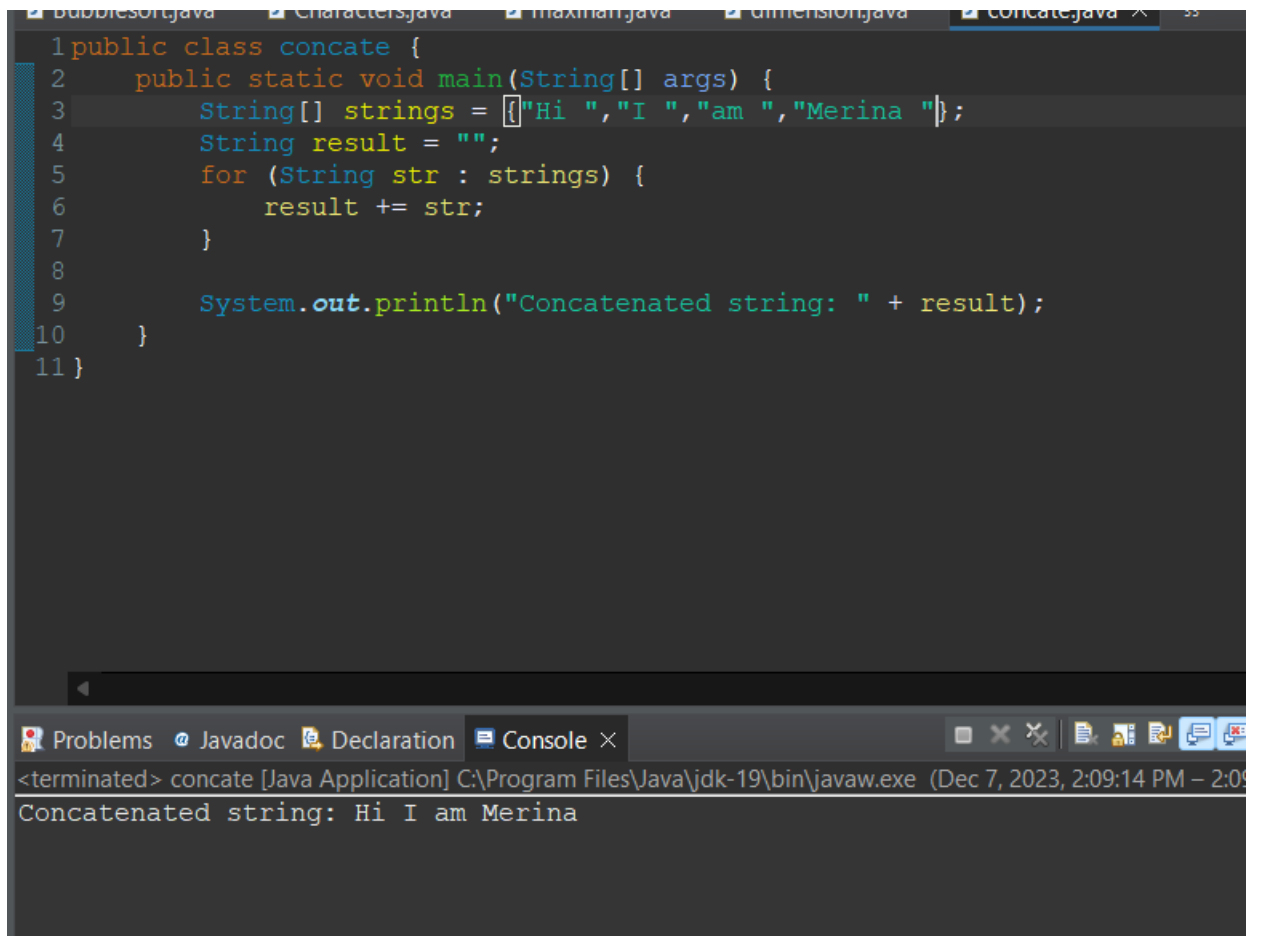

```
1
2 public class dimension {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         int arr[][] = {{2,3},{4,5}};
7         int sum = 0;
8
9         for (int[] row : arr) {
10             for (int element : row) {
11                 sum += element;
12             }
13         }
14         System.out.println("The sum of all elements in the 2D array is: " + sum);
15     }
16 }
17
18
```

Problems Javadoc Declaration Console ×

<terminated> dimension [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 7, 2023, 2:07:21 PM – 2:07:21 PM)
The sum of all elements in the 2D array is: 14

12. Create a Java program that utilizes the enhanced for loop to concatenate all strings in an array and print the result.

Object Oriented Programming and Design
Workshop 3



The screenshot shows an IDE with a Java file named `Concatate.java` open. The code defines a `concatate` class with a `main` method that concatenates the strings "Hi ", "I ", "am ", and "Merina " into a single string. The IDE's interface includes tabs for `Bubblesort.java`, `Characters.java`, `Maximian.java`, `Dimension.java`, and `Concatate.java`. Below the code editor, there is a `Console` tab showing the program's output.

```
1 public class concatate {  
2     public static void main(String[] args) {  
3         String[] strings = {"Hi ", "I ", "am ", "Merina "};  
4         String result = "";  
5         for (String str : strings) {  
6             result += str;  
7         }  
8  
9         System.out.println("Concatenated string: " + result);  
10    }  
11 }
```

Problems Javadoc Declaration Console ×

<terminated> concatate [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 7, 2023, 2:09:14 PM – 2:09:14 PM)
Concatenated string: Hi I am Merina