PROGRAMMING IN JAVA LAB

ASSIGNMENT 1

YASHITA UGHADE

22070126135

AIML B3

CALCULATOR CLASS

```
import java.util.Arrays;
import java.util.HashMap;
import java.util.Map;
public class Calculator {
// Function to perform addition
 public double add(double num1, double num2) {
 return num1 + num2;
 }
 // Function to perform subtraction
 public double subtract(double num1, double num2) {
 return num1 - num2;
 }
 // Function to perform multiplication
 public double multiply(double num1, double num2) {
```

```
return num1 * num2;
}
// Function to perform division
public double divide(double num1, double num2) {
return num1 / num2;
}
// Function to calculate the sum of an array
public double sumArray(double[] array) {
return Arrays.stream(array).sum();
}
// Function to calculate the mean of an array
public double meanArray(double[] array) {
return sumArray(array) / array.length;
}
// Function to calculate the mode of an array
public double modeArray(double[] array) {
if (array.length == 0) {
throw new IllegalArgumentException("Array
cannot be empty");
}
```

```
 // Create a frequency map to store the occurrences
of each element
 Map<Double, Integer> frequencyMap = new
HashMap<>();
 // Populate the frequency map
 for (double num : array) {
 frequencyMap.put(num,
frequencyMap.getOrDefault(num, 0) + 1);
 }
 // Find the mode(s)
 double mode = 0;
 int maxFrequency = 0;
 for (Map.Entry<Double, Integer> entry :
frequencyMap.entrySet()) {
 if (entry.getValue() > maxFrequency) {
 maxFrequency = entry.getValue();
 mode = entry.getKey();
 }
 }
 // Check if there is more than one mode
```

```
 for (Map.Entry<Double, Integer> entry :
frequencyMap.entrySet()) {
 if (entry.getValue() == maxFrequency &&
entry.getKey() != mode) {
 throw new IllegalStateException("Array has
multiple modes");
 }
 }
 return mode;
 }
 // Function to calculate the median of an array
 public double medianArray(double[] array) {
 Arrays.sort(array);
 int middle = array.length / 2;
 return (array.length % 2 == 0) ? (array[middle - 1] +
array[middle]) / 2.0 : array[middle];
 }
 // Function to calculate the variance of an array
 public double varianceArray(double[] array) {
 double mean = meanArray(array);
```

```
 double sumSquaredDifferences =
Arrays.stream(array).map(x -> Math.pow(x - mean, 2)).sum();
 return sumSquaredDifferences / array.length;
 }
 // Function to calculate the standard deviation of an array
 public double standardDeviationArray(double[] array) {
 return Math.sqrt(varianceArray(array));
 }
}
MAIN CLASS
public class Main {
public static void main(String[] args){
// Create an instance of UserInput
class
 UserInput userInput = new UserInput();
 // Get two numbers from the user
 double num1 =
userInput.getDoubleInput("Enter the first number: ");
 double num2 =
userInput.getDoubleInput("Enter the second number: ");
```

```
 // Create an instance of Calculator class
 Calculator calculator = new Calculator();
 // Perform basic operations
 System.out.println("Addition: " +
calculator.add(num1, num2));
 System.out.println("Subtraction: " +
calculator.subtract(num1, num2));
 System.out.println("Multiplication: " +
calculator.multiply(num1, num2));
 // Check for division by zero
 if (num2 != 0) {
 System.out.println("Division: " +
calculator.divide(num1, num2));
 } else {
 System.out.println("Error! Division by
zero is not allowed.");
 }
 // Get an array from the user
 double[] numbersArray =
userInput.getArrayInput();
```

```
 // Perform array operations
 System.out.println("Sum of Array: " +
calculator.sumArray(numbersArray));
 System.out.println("Mean of Array: " +
calculator.meanArray(numbersArray));
 System.out.println("Mode of Array: " +
calculator.modeArray(numbersArray));
 System.out.println("Median of Array: " +
calculator.medianArray(numbersArray));
 System.out.println("Variance of Array: " +
calculator.varianceArray(numbersArray));
 System.out.println("Standard Deviation of
Array: " + calculator.standardDeviationArray(numbersArray));
 }
}
```

INPUT CLASS

```
import java.util.Scanner;
import java.util.Arrays;
public class UserInput {
```

```
private Scanner scanner;
public UserInput() {
scanner = new Scanner(System.in);
}
//Function to get input from user
public double getDoubleInput(String prompt) {
System.out.println(prompt);
return scanner.nextDouble();
}
//Function to get an array of doubles from user
public double[] getArrayInput() {
System.out.print("Enter the size of the
array: ");
int size = scanner.nextInt();
double[] array = new double[size];
for (int i = 0; i < size; i++) {
array[i] = getDoubleInput("Enter the
elements " + (i + 1) + " :");
}
return array;
```

```
}
}
```

```
Enter the first number:
12
Enter the second number:
45
Addition: 57.0
Subtraction: -33.0
Multiplication: 540.0
Division: 0.26666666666666666
Enter the size of the array: 5
Enter the elements 1 :
1
Enter the elements 2 :
2
Enter the elements 3 :
2
Enter the elements 4 :
2
Enter the elements 5 :
67
Sum of Array: 74.0
Mean of Array: 14.8
Mode of Array: 2.0
Median of Array: 2.0
Variance of Array: 681.36
Standard Deviation of Array: 26.102873405048726
```

FIBONACCI SERIES

```
import java.io.*;
public class Fibonacci
{
public static void main(String args[])
{
//int number = Integer.parseInt(args[0]);
//BufferedReader
BufferedReader reader = new
BufferedReader(new InputStreamReader(System.in));
```

```
try
{
String input = reader.readLine();
 int number = Integer.parseInt(input);
for(int i = 0; i < number; i++)
{
System.out.println(fibonacci(i) +
" ");
}
}
catch(IOException e)
{
System.out.println("Invalid input.
Please enter a valid number");
}
}
public static int fibonacci(int n)
{
if(n <= 1)
{
```

```
return n;
}
else
{
return fibonacci(n - 1) + fibonacci(n
- 2);
}
}
}
```