Calculator.java

```java
// Name: Rupankar Das
// PRN: 23070126111
// Batch: Class of 2027 | B.Tech AIML | B2

import java.util.HashMap;
import java.util.Map;

public class Calculator {

    // Method to add two numbers
    public double add(double a, double b) {
        return a + b; // Return the sum of a and b
    }

    // Method to subtract second number from first number
    public double subtract(double a, double b) {
        return a - b; // Return the difference of a and b
    }

    // Method to multiply two numbers
    public double multiply(double a, double b) {
        return a * b; // Return the product of a and b
    }

    // Method to divide first number by second number
    public double divide(double a, double b) {
        if (b == 0) { // Check if divisor is zero
            throw new ArithmeticException("Cannot divide by zero"); // Throw exception if divisor is zero
        }
        return a / b; // Return the quotient of a and b
    }

    // Method to calculate the nth Fibonacci number
    public int fibonacci(int n) {
        if (n <= 1) return n; // Base case: return n if n is 0 or 1
        return fibonacci(n - 1) + fibonacci(n - 2); // Recursive case: return sum of previous two Fibonacci numbers
    }

    // Method to calculate the mean of an array of numbers
    public double mean(double[] array) {
        double sum = 0; // Initialize sum to 0
        for (double num : array) { // Iterate through each number in the array
            sum += num; // Add each number to sum
        }
        return sum / array.length; // Return the mean (sum divided by number of elements)
    }

    // Method to find the mode of an array of integers
    public int mode(int[] array) {
```

```
50          Map<Integer, Integer> frequencyMap = new HashMap<>(); // Create a map to
    store frequency of each number
51          for (int num : array) { // Iterate through each number in the array
52              frequencyMap.put(num, frequencyMap.getOrDefault(num, 0) + 1); // Update
    the frequency of each number
53          }
54
55          int mode = array[0]; // Initialize mode to the first element of the array
56          int maxCount = 0; // Initialize maxCount to 0
57          for (Map.Entry<Integer, Integer> entry : frequencyMap.entrySet()) { //
    Iterate through the frequency map
58              if (entry.getValue() > maxCount) { // Check if current frequency is
    greater than maxCount
59                  maxCount = entry.getValue(); // Update maxCount
60                  mode = entry.getKey(); // Update mode
61              }
62          }
63          return mode; // Return the mode
64      }
65  }
66
```