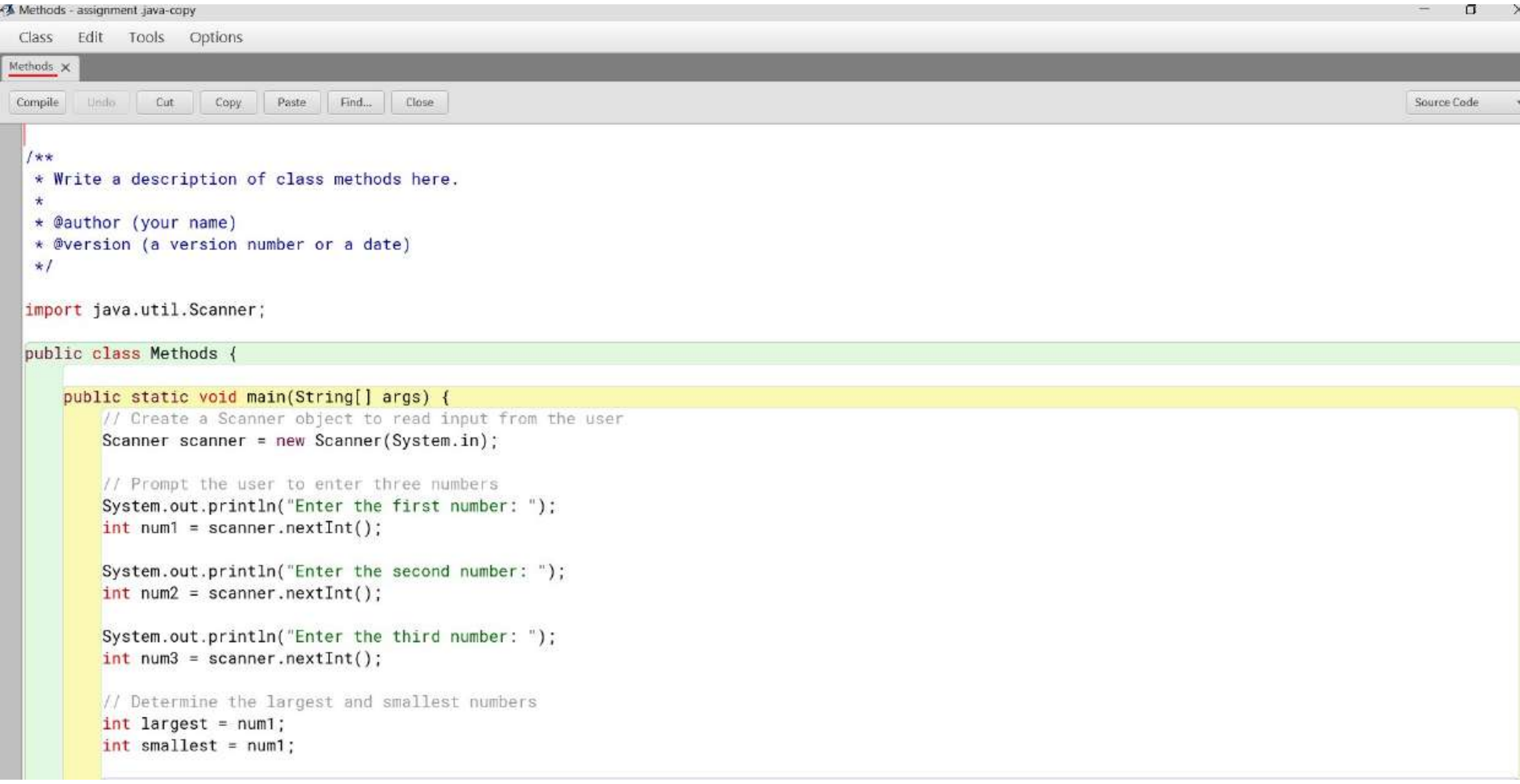
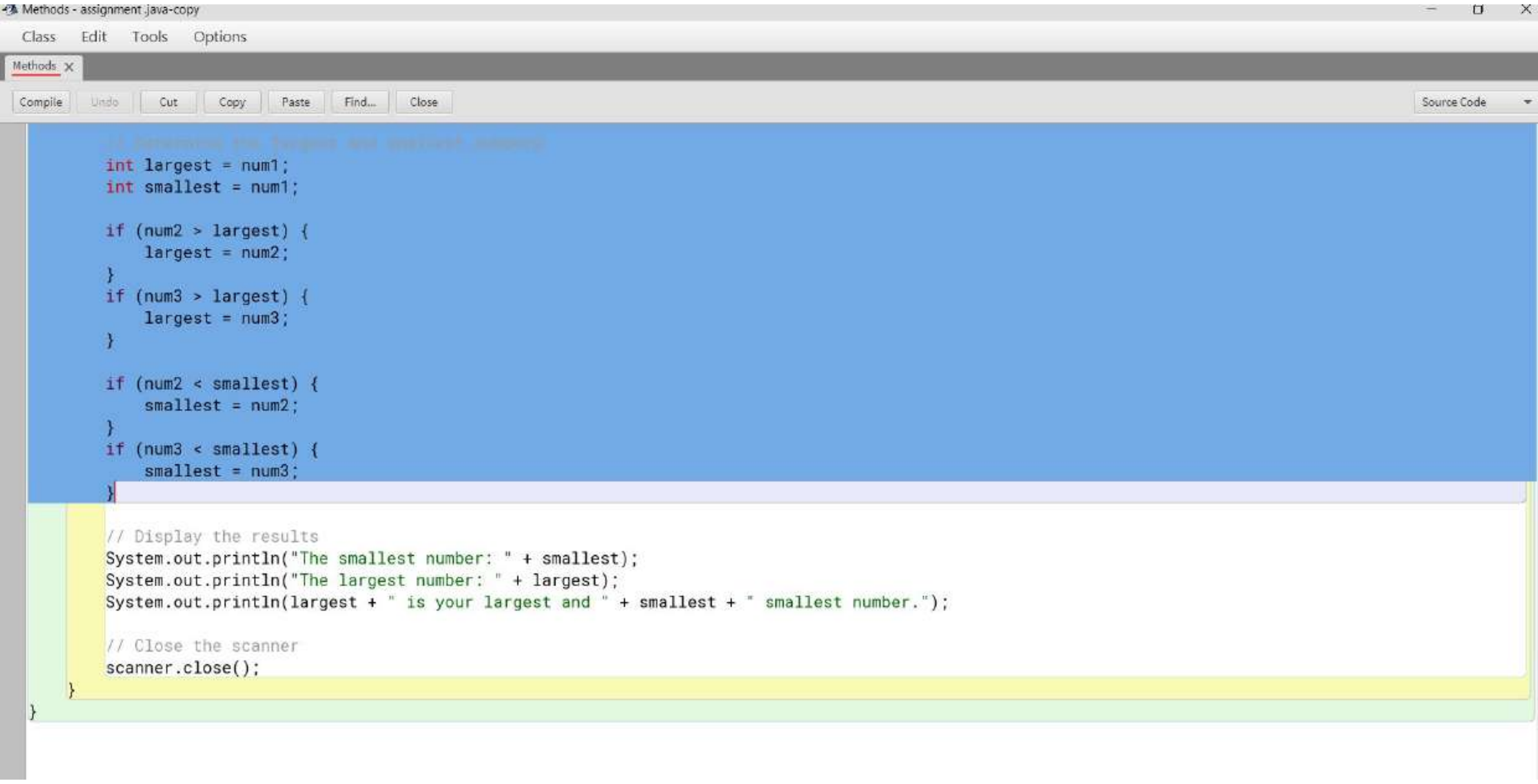
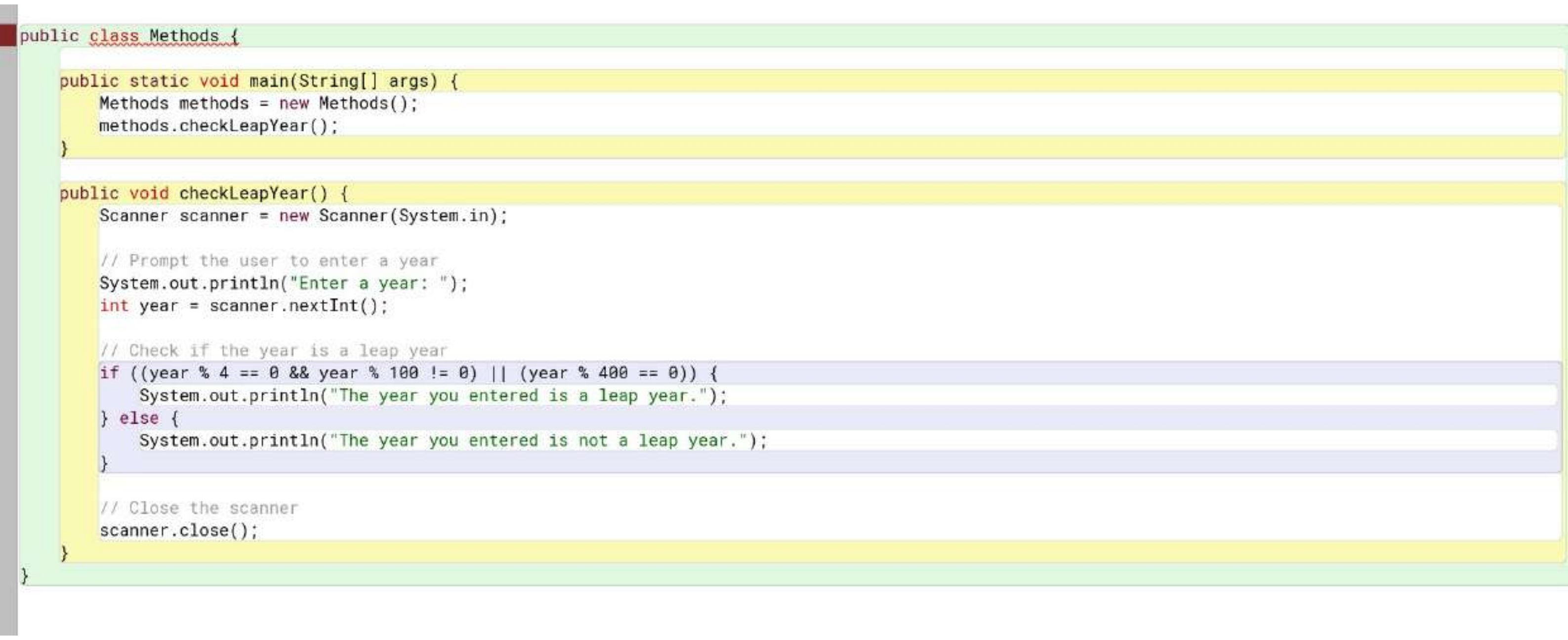


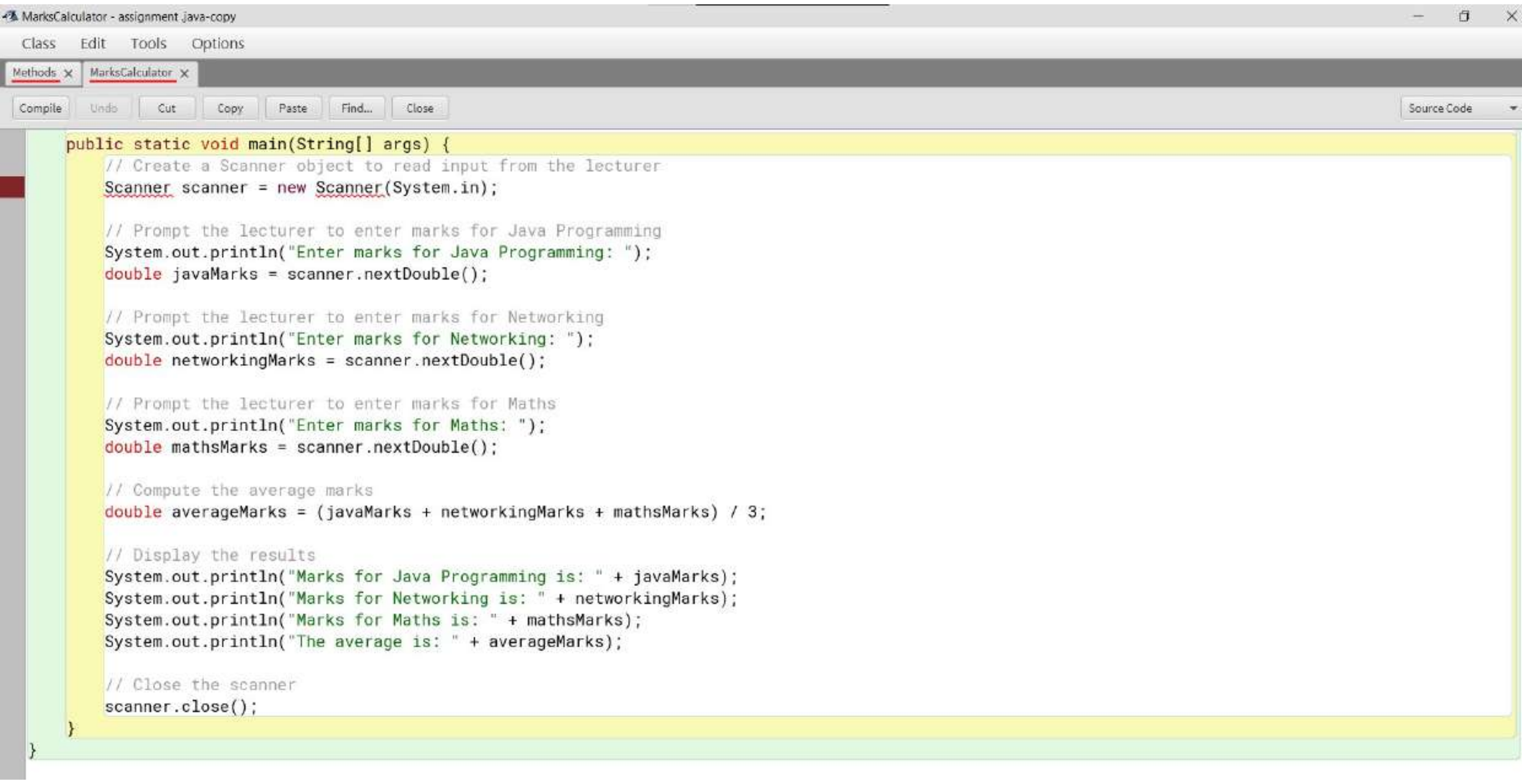
```
/**
 * Write a description of class methods here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */

import java.util.Scanner;

public class Methods {

    public static void main(String[] args) {
        // Create a Scanner object to read input from the user
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter three numbers
        System.out.println("Enter the first number: ");
        int num1 = scanner.nextInt();

        System.out.println("Enter the second number: ");
        int num2 = scanner.nextInt();

        System.out.println("Enter the third number: ");
        int num3 = scanner.nextInt();

        // Determine the largest and smallest numbers
        int largest = num1;
        int smallest = num1;
```



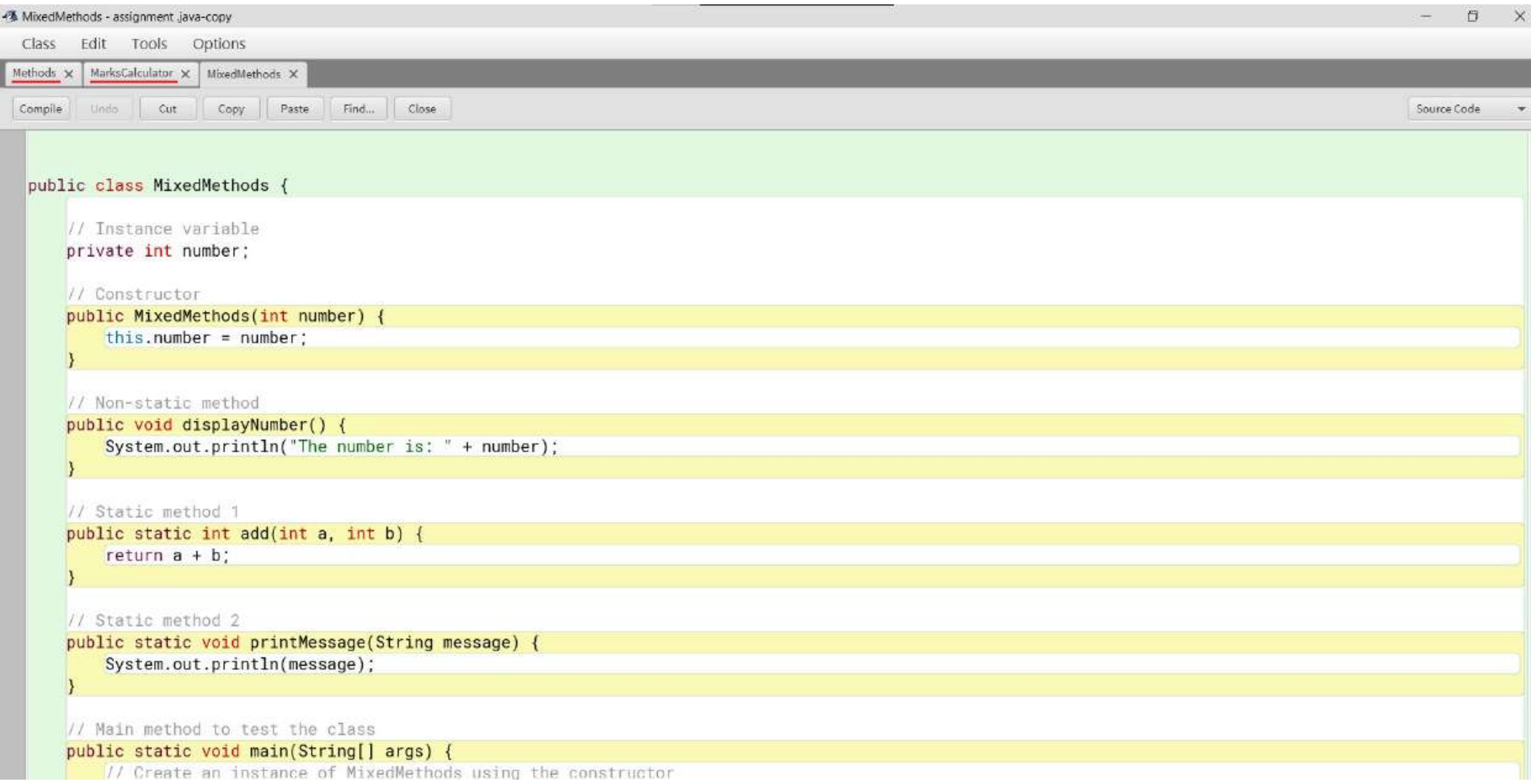
```
        // Determine the largest and smallest numbers
        int largest = num1;
        int smallest = num1;

        if (num2 > largest) {
            largest = num2;
        }
        if (num3 > largest) {
            largest = num3;
        }

        if (num2 < smallest) {
            smallest = num2;
        }
        if (num3 < smallest) {
            smallest = num3;
        }

        // Display the results
        System.out.println("The smallest number: " + smallest);
        System.out.println("The largest number: " + largest);
        System.out.println(largest + " is your largest and " + smallest + " smallest number.");

        // Close the scanner
        scanner.close();
    }
}
```



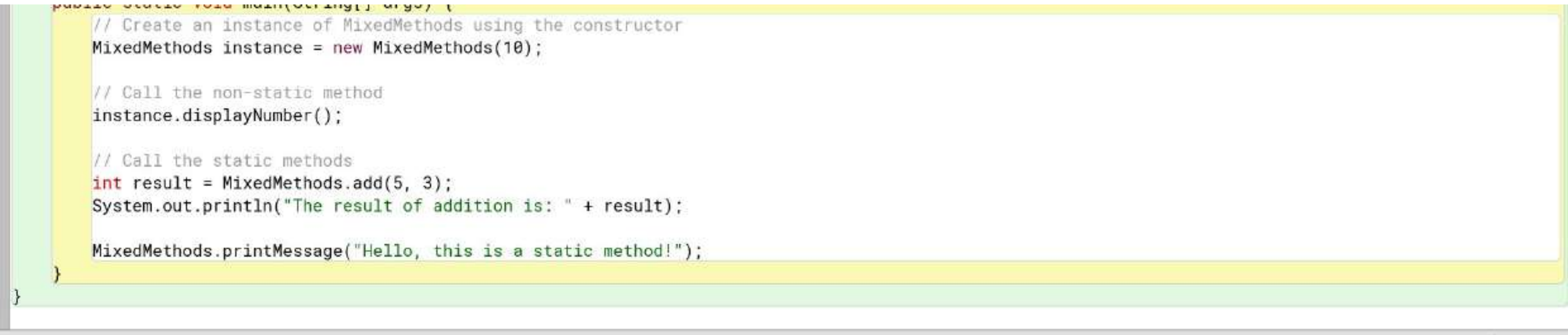
```java
public class Methods {

    public static void main(String[] args) {
        Methods methods = new Methods();
        methods.checkLeapYear();
    }

    public void checkLeapYear() {
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter a year
        System.out.println("Enter a year: ");
        int year = scanner.nextInt();

        // Check if the year is a leap year
        if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
            System.out.println("The year you entered is a leap year.");
        } else {
            System.out.println("The year you entered is not a leap year.");
        }

        // Close the scanner
        scanner.close();
    }
}
```



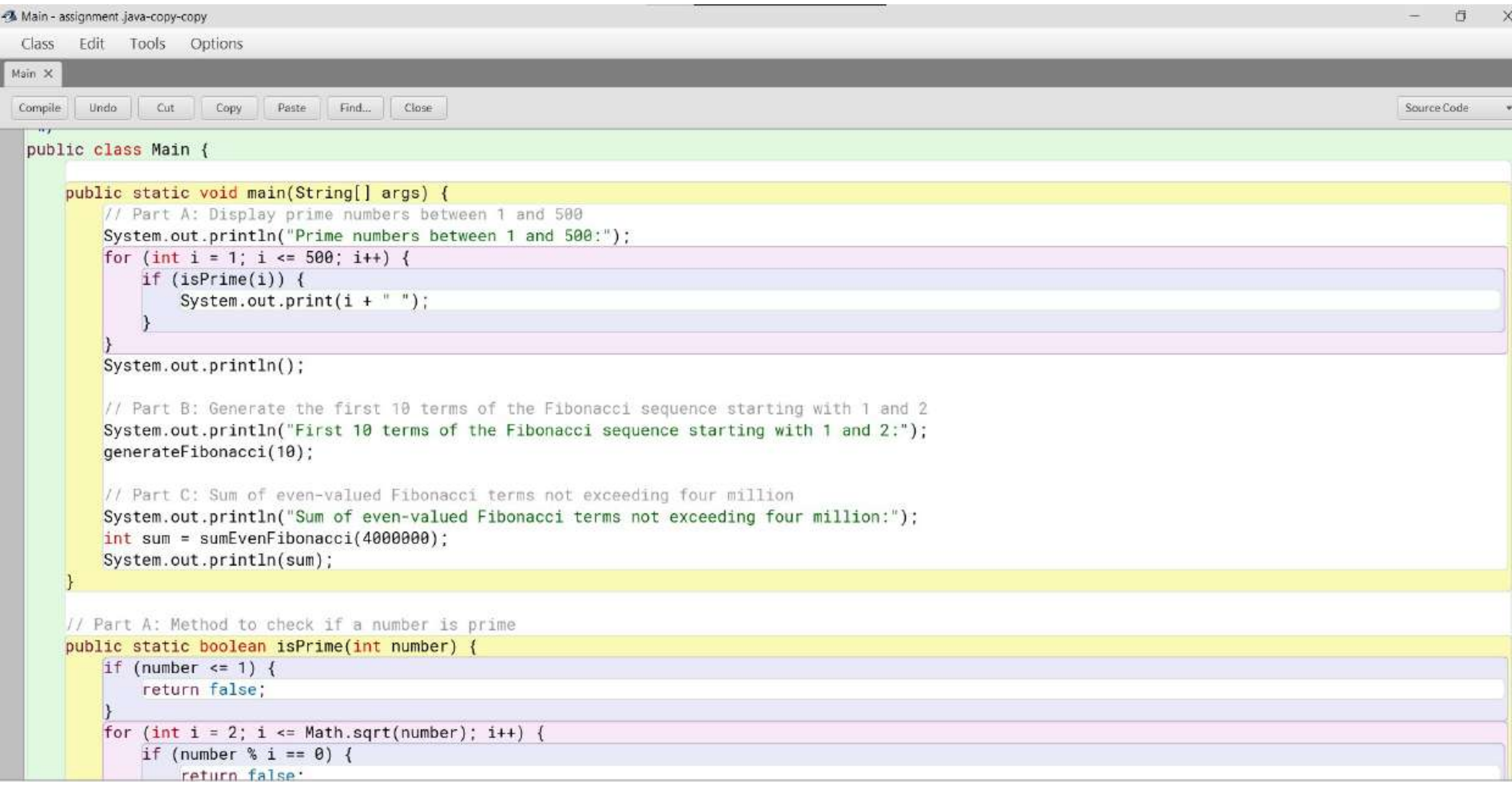
```
    public static void main(String[] args) {
        // Create a Scanner object to read input from the lecturer
        Scanner scanner = new Scanner(System.in);

        // Prompt the lecturer to enter marks for Java Programming
        System.out.println("Enter marks for Java Programming: ");
        double javaMarks = scanner.nextDouble();

        // Prompt the lecturer to enter marks for Networking
        System.out.println("Enter marks for Networking: ");
        double networkingMarks = scanner.nextDouble();

        // Prompt the lecturer to enter marks for Maths
        System.out.println("Enter marks for Maths: ");
        double mathsMarks = scanner.nextDouble();

        // Compute the average marks
        double averageMarks = (javaMarks + networkingMarks + mathsMarks) / 3;

        // Display the results
        System.out.println("Marks for Java Programming is: " + javaMarks);
        System.out.println("Marks for Networking is: " + networkingMarks);
        System.out.println("Marks for Maths is: " + mathsMarks);
        System.out.println("The average is: " + averageMarks);

        // Close the scanner
        scanner.close();
    }
}
```



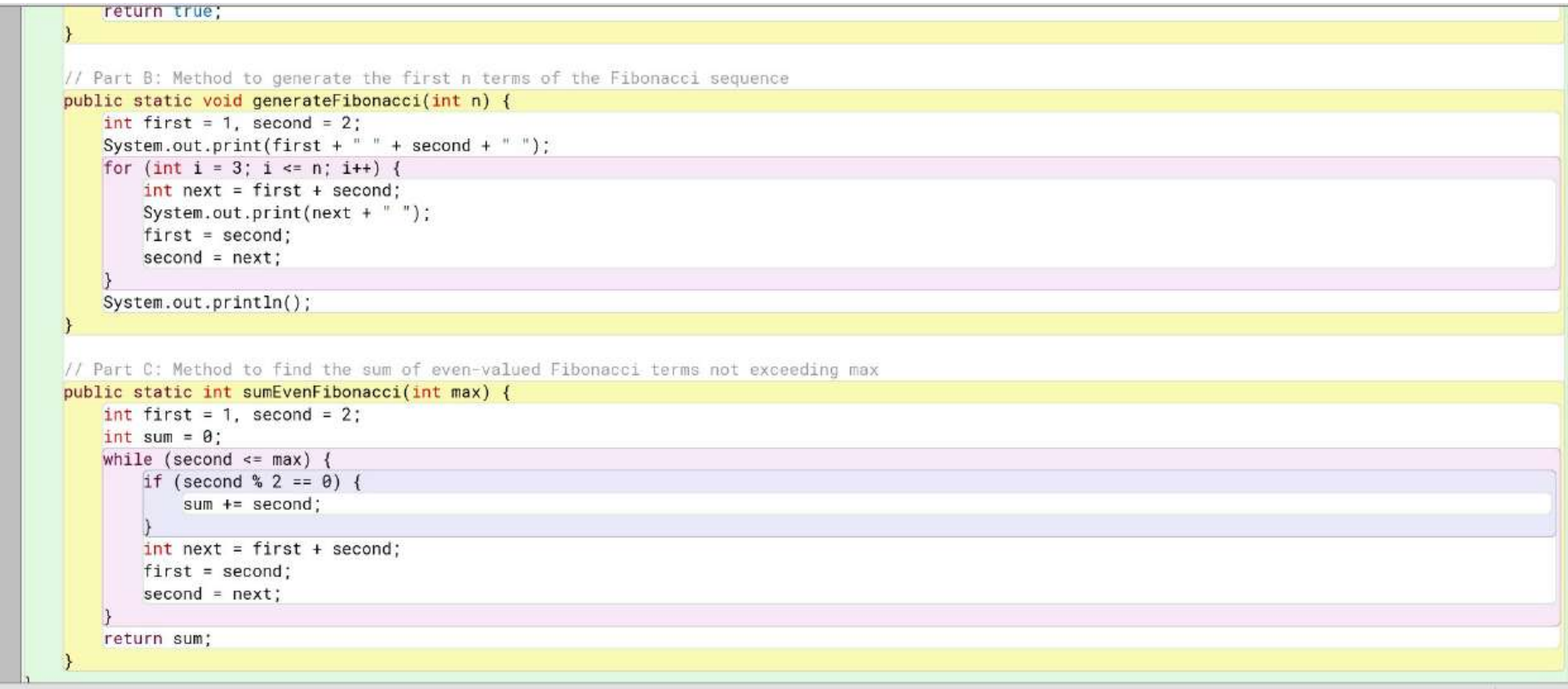
```java
public class MixedMethods {

    // Instance variable
    private int number;

    // Constructor
    public MixedMethods(int number) {
        this.number = number;
    }

    // Non-static method
    public void displayNumber() {
        System.out.println("The number is: " + number);
    }

    // Static method 1
    public static int add(int a, int b) {
        return a + b;
    }

    // Static method 2
    public static void printMessage(String message) {
        System.out.println(message);
    }

    // Main method to test the class
    public static void main(String[] args) {
        // Create an instance of MixedMethods using the constructor
```



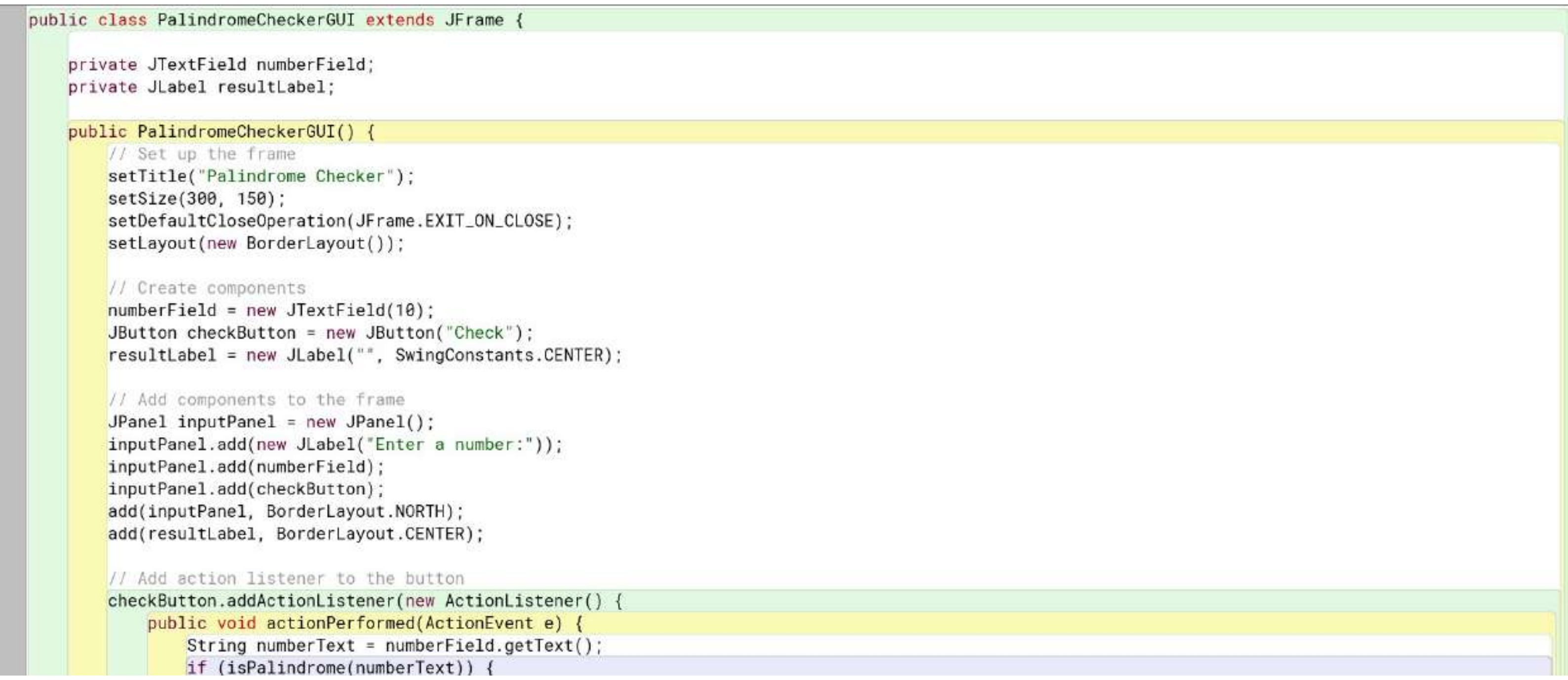
```
    public static void main(String[] args) {
        // Create an instance of MixedMethods using the constructor
        MixedMethods instance = new MixedMethods(10);

        // Call the non-static method
        instance.displayNumber();

        // Call the static methods
        int result = MixedMethods.add(5, 3);
        System.out.println("The result of addition is: " + result);

        MixedMethods.printMessage("Hello, this is a static method!");
    }
}
```



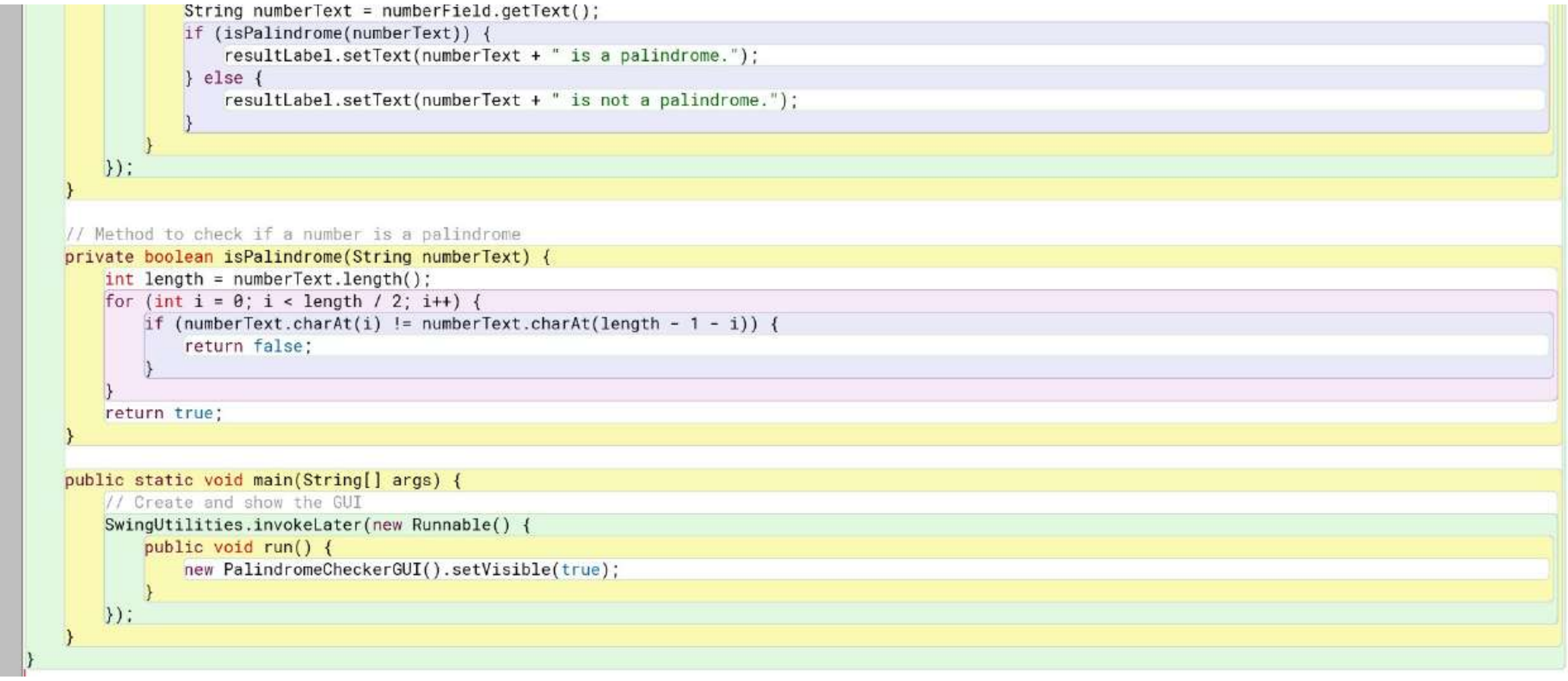
```java
*/
public class Main {

    public static void main(String[] args) {
        // Part A: Display prime numbers between 1 and 500
        System.out.println("Prime numbers between 1 and 500:");
        for (int i = 1; i <= 500; i++) {
            if (isPrime(i)) {
                System.out.print(i + " ");
            }
        }
        System.out.println();

        // Part B: Generate the first 10 terms of the Fibonacci sequence starting with 1 and 2
        System.out.println("First 10 terms of the Fibonacci sequence starting with 1 and 2:");
        generateFibonacci(10);

        // Part C: Sum of even-valued Fibonacci terms not exceeding four million
        System.out.println("Sum of even-valued Fibonacci terms not exceeding four million:");
        int sum = sumEvenFibonacci(4000000);
        System.out.println(sum);
    }

    // Part A: Method to check if a number is prime
    public static boolean isPrime(int number) {
        if (number <= 1) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(number); i++) {
            if (number % i == 0) {
                return false;
```



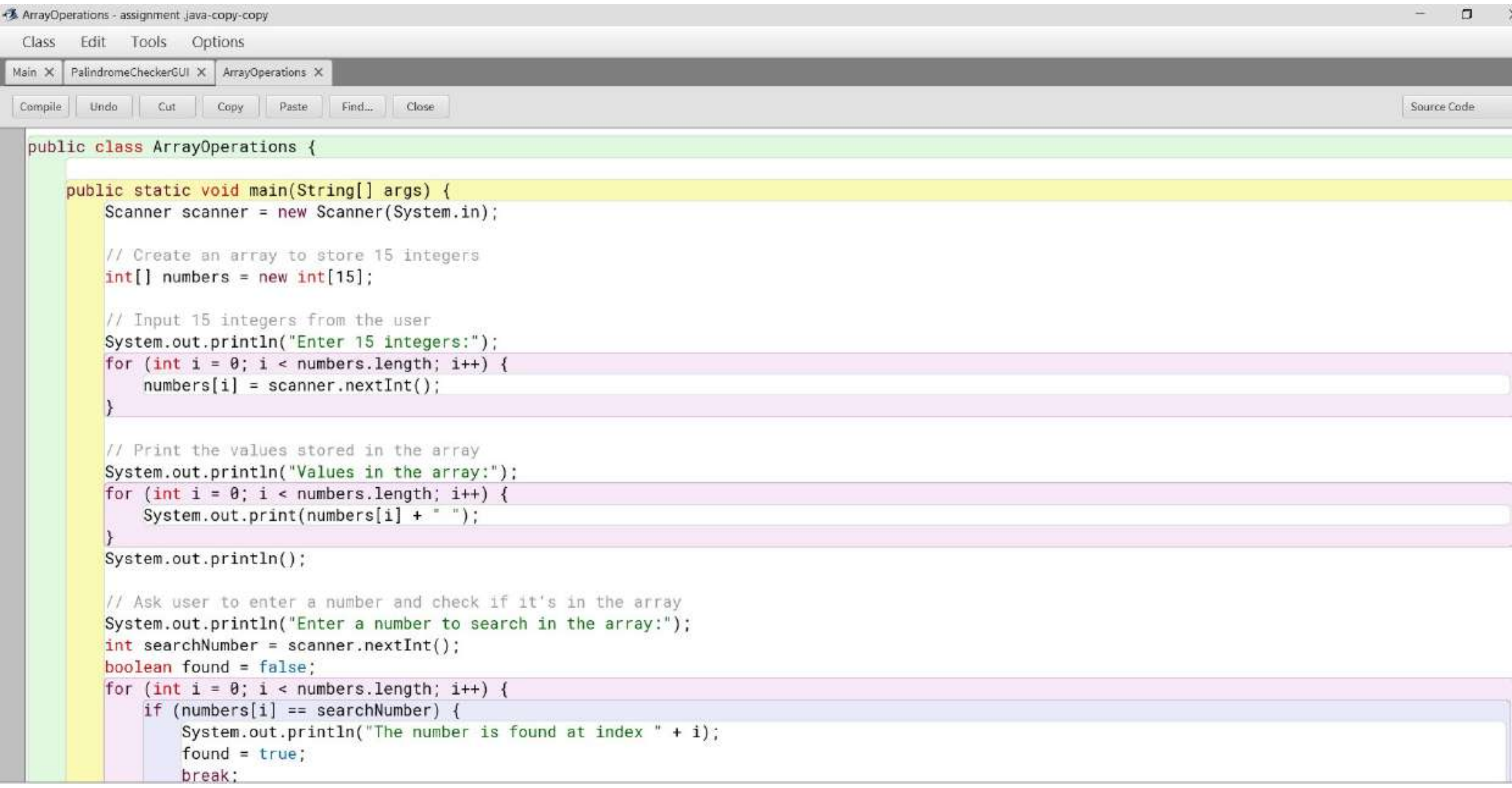
```java
        return true;
    }

    // Part B: Method to generate the first n terms of the Fibonacci sequence
    public static void generateFibonacci(int n) {
        int first = 1, second = 2;
        System.out.print(first + " " + second + " ");
        for (int i = 3; i <= n; i++) {
            int next = first + second;
            System.out.print(next + " ");
            first = second;
            second = next;
        }
        System.out.println();
    }

    // Part C: Method to find the sum of even-valued Fibonacci terms not exceeding max
    public static int sumEvenFibonacci(int max) {
        int first = 1, second = 2;
        int sum = 0;
        while (second <= max) {
            if (second % 2 == 0) {
                sum += second;
            }
            int next = first + second;
            first = second;
            second = next;
        }
        return sum;
    }
```



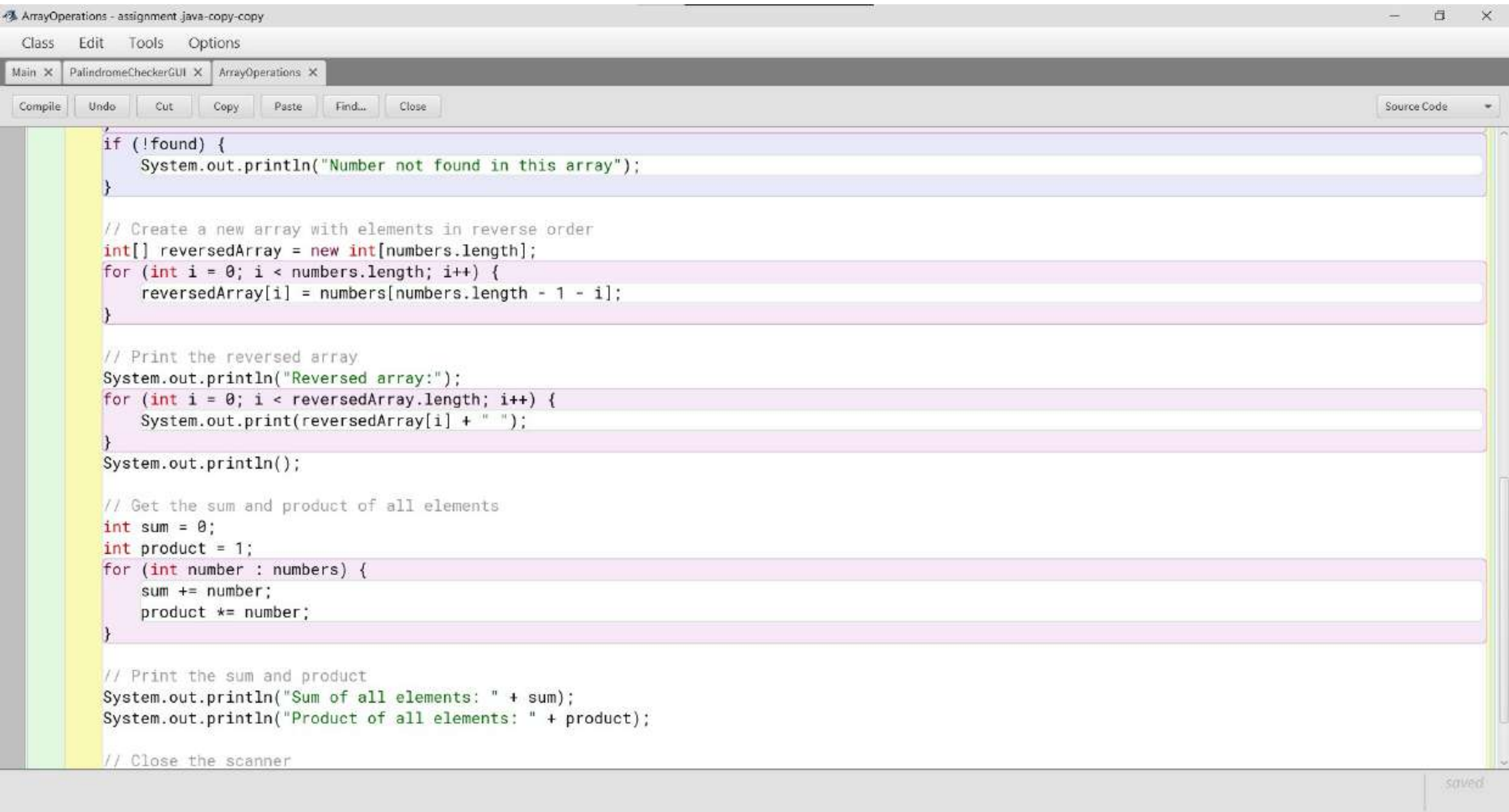
```java
public class PalindromeCheckerGUI extends JFrame {

    private JTextField numberField;
    private JLabel resultLabel;

    public PalindromeCheckerGUI() {
        // Set up the frame
        setTitle("Palindrome Checker");
        setSize(300, 150);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        // Create components
        numberField = new JTextField(10);
        JButton checkButton = new JButton("Check");
        resultLabel = new JLabel("", SwingConstants.CENTER);

        // Add components to the frame
        JPanel inputPanel = new JPanel();
        inputPanel.add(new JLabel("Enter a number:"));
        inputPanel.add(numberField);
        inputPanel.add(checkButton);
        add(inputPanel, BorderLayout.NORTH);
        add(resultLabel, BorderLayout.CENTER);

        // Add action listener to the button
        checkButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String numberText = numberField.getText();
                if (isPalindrome(numberText)) {
```



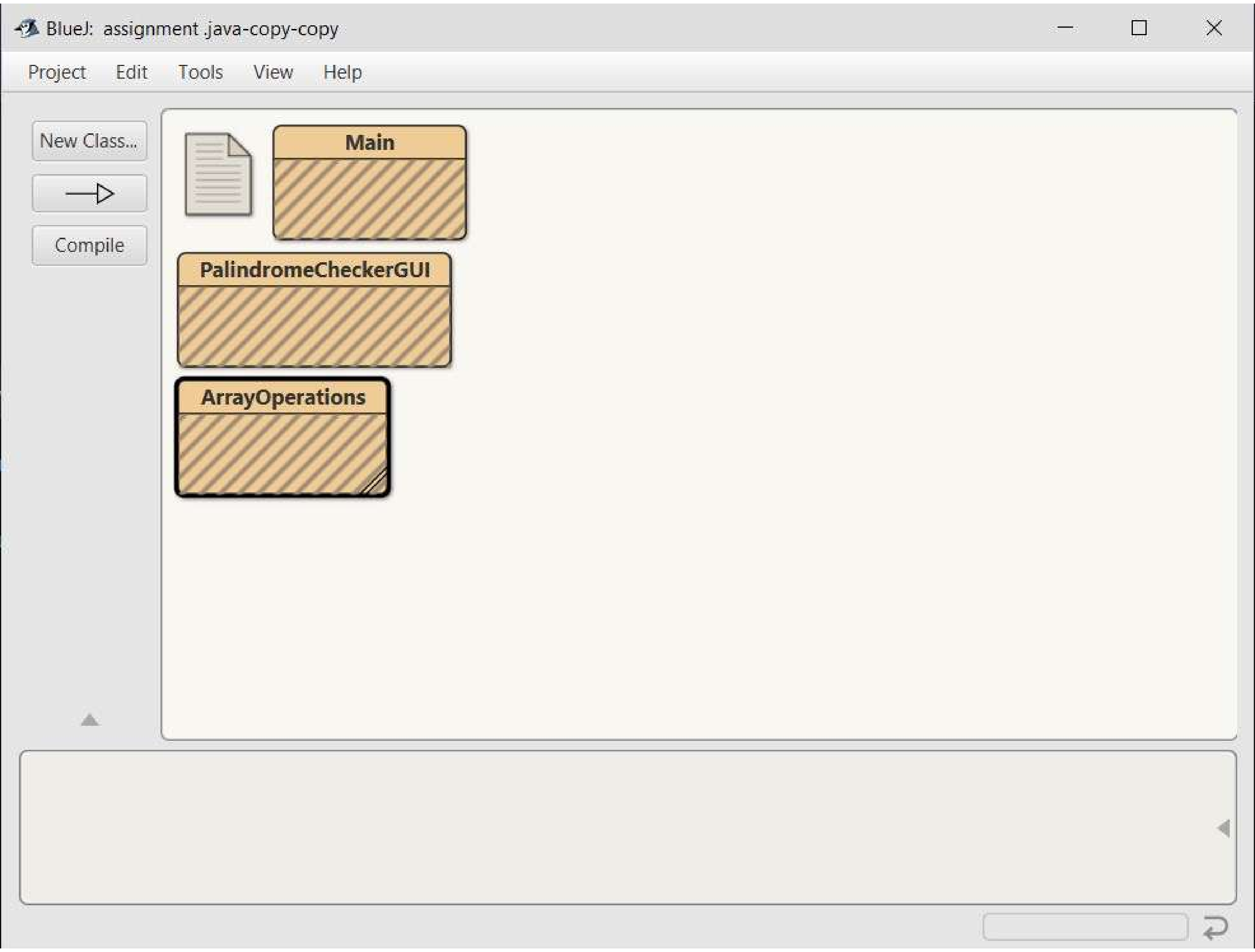
```
                String numberText = numberField.getText();
                if (isPalindrome(numberText)) {
                    resultLabel.setText(numberText + " is a palindrome.");
                } else {
                    resultLabel.setText(numberText + " is not a palindrome.");
                }
            }
        });
    }

    // Method to check if a number is a palindrome
    private boolean isPalindrome(String numberText) {
        int length = numberText.length();
        for (int i = 0; i < length / 2; i++) {
            if (numberText.charAt(i) != numberText.charAt(length - 1 - i)) {
                return false;
            }
        }
        return true;
    }

    public static void main(String[] args) {
        // Create and show the GUI
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new PalindromeCheckerGUI().setVisible(true);
            }
        });
    }
}
```

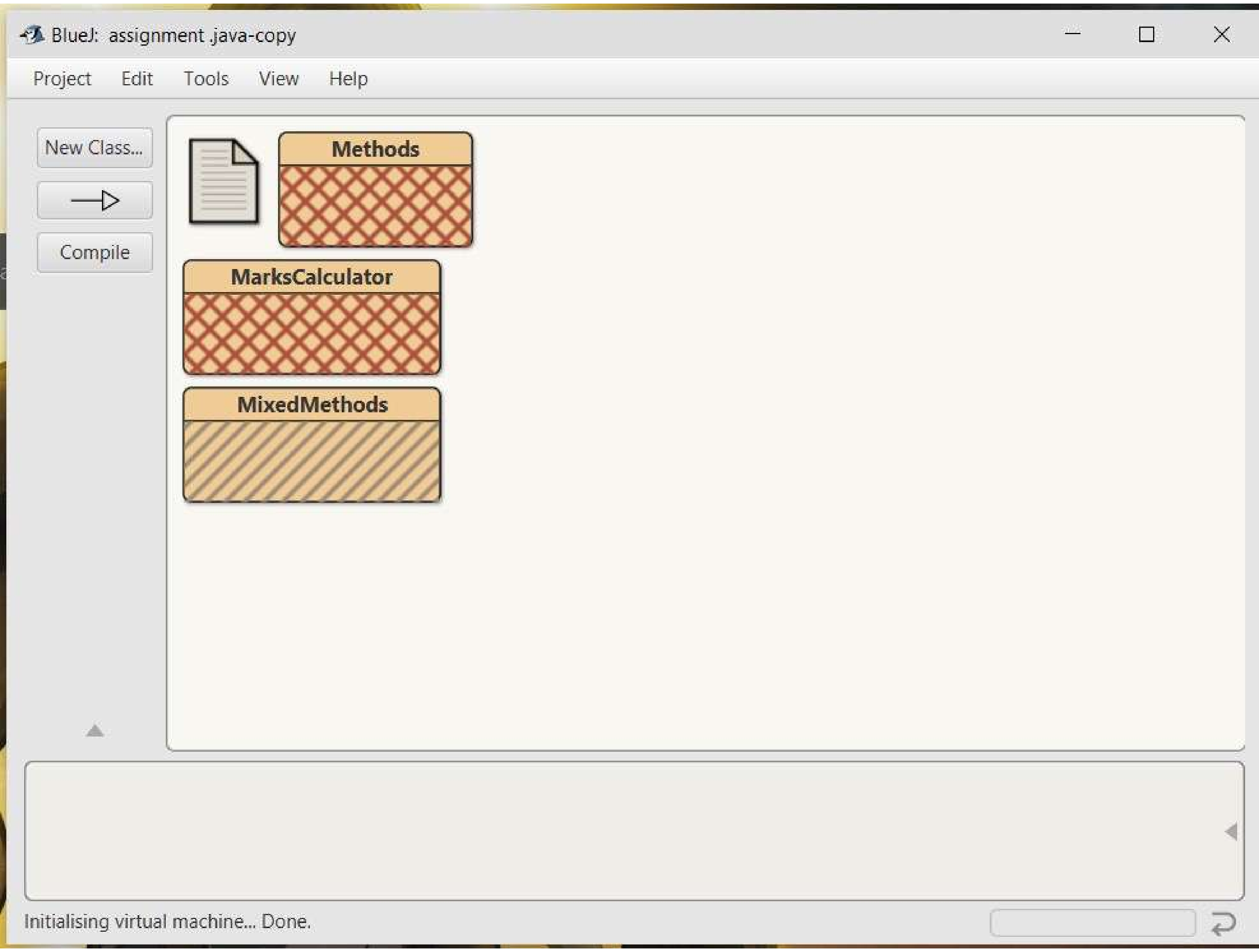


```
public class ArrayOperations {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Create an array to store 15 integers
        int[] numbers = new int[15];

        // Input 15 integers from the user
        System.out.println("Enter 15 integers:");
        for (int i = 0; i < numbers.length; i++) {
            numbers[i] = scanner.nextInt();
        }

        // Print the values stored in the array
        System.out.println("Values in the array:");
        for (int i = 0; i < numbers.length; i++) {
            System.out.print(numbers[i] + " ");
        }
        System.out.println();

        // Ask user to enter a number and check if it's in the array
        System.out.println("Enter a number to search in the array:");
        int searchNumber = scanner.nextInt();
        boolean found = false;
        for (int i = 0; i < numbers.length; i++) {
            if (numbers[i] == searchNumber) {
                System.out.println("The number is found at index " + i);
                found = true;
                break;
```



```
if (!found) {
    System.out.println("Number not found in this array");
}

// Create a new array with elements in reverse order
int[] reversedArray = new int[numbers.length];
for (int i = 0; i < numbers.length; i++) {
    reversedArray[i] = numbers[numbers.length - 1 - i];
}

// Print the reversed array
System.out.println("Reversed array:");
for (int i = 0; i < reversedArray.length; i++) {
    System.out.print(reversedArray[i] + " ");
}
System.out.println();

// Get the sum and product of all elements
int sum = 0;
int product = 1;
for (int number : numbers) {
    sum += number;
    product *= number;
}

// Print the sum and product
System.out.println("Sum of all elements: " + sum);
System.out.println("Product of all elements: " + product);

// Close the scanner
```


BlueJ: assignment .java-copy-copy
Project
Edit
Tools
View
Help
New Class...
Compile
Main
PalindromeCheckerGUI
ArrayOperations



Initialising virtual machine... Done.