

Phạm Trần Gia Hưng MSSV:2331200153

1. Use the attached files “Circle-Radius.java” and “Circle-Radius-Test.java”. Add the code in the main function to accommodate the last test case that is when test data is "abc"

Bug:

```
        } else
            System.out.println("You have entered wrong value.");
        // bug fix: add exception handling for input like "abc"
        // the code didn't have input reader
    } catch (Exception e) {
        System.out.println("You have entered wrong value.");
    }
```

Test:

The screenshot shows an IDE interface with two tabs open: CircleRadiusTest.java and CircleRadius-Test.java.

CircleRadiusTest.java:

```
16     @Test
17     void testMain() {
18         String input="1\n";
19         ByteArrayInputStream in=new
20             ByteArrayInputStream(input.getBytes());
21         System.setIn(in);
22         ByteArrayOutputStream out=new ByteArrayOutputStream();
23         System.setOut(new PrintStream(out));
24         String[] args={};
25         CircleRadius.main(args);
26
27         String consoleOutput="Enter the radius "+System.getProperty("line.separator");
28         consoleOutput+="For a circle with radius 1.0,"+System.getProperty("line.separator");
29         consoleOutput+="The circumference is 6.283185307179586"+System.getProperty("line.separator");
30         consoleOutput+="and the area is 3.141592653589793."+System.getProperty("line.separator");
31
32         assertEquals(consoleOutput,out.toString());
33     }
34 }
```

CircleRadius-Test.java:

```
16     @Test
17     void testMainWithABC() {
18         String input = "abc\n";
19         ByteArrayInputStream in =
20             new ByteArrayInputStream(input.getBytes());
21         System.setIn(in);
22
23         ByteArrayOutputStream out = new ByteArrayOutputStream();
24         System.setOut(new PrintStream(out));
25
26         String[] args = {};
27         CircleRadius.main(args);
28
29         String consoleOutput = "Enter the radius " + System.lineSeparator();
30         consoleOutput += "You have entered wrong value." + System.lineSeparator();
31
32         assertEquals(consoleOutput,out.toString());
33     }
34 }
```

Both tests in CircleRadiusTest.java pass, indicated by green checkmarks in the run history.

2. Consider the file “RomantoInteger.java”. This program implements a program that converts a Roman numeral that is in form of a String, to an integer. The Roman numeral system is based on seven essential numerals: I, V, X, L, C, D, and M (1, 5, 10, 50, 100, 500, and 1,000, respectively). So, rules are
- a. We can't add more than three of the same Roman numeral together
 - b. If a numeral comes after a numeral that is larger or equal in value, then it must be added to the numeral before it.
 - c. Smaller numeral placed before a larger numeral indicates subtraction of that smaller numeral from the larger one.
 - d. We can't subtract more than one value from a Roman numeral. Example of test cases for this program:

Bug:

```
6
7         int currentNumber=map.get(s.charAt(i));
8         int next=i+1 < s.length() ? map.get(s.charAt(i+1)) :0;
9
10
11        // error: inverted logic (- and + swapped)
12        if (currentNumber>=next)
13            convertedNumber-=currentNumber;
14        else
15            convertedNumber+=currentNumber;
16
17        if (currentNumber>=next)
18            convertedNumber+=currentNumber;
19        else
20            convertedNumber-=currentNumber;
```

Test:

```

src
  CircleRadius
  CircleRadiusTest
  Fibonacci
  FibonacciTest
  Line
  Point
  Quadrilateral
  QuadrilateralTest
  Romantointeger
  RomantointegerTest
  Triangle
  TriangleTest
  Vector2D
.gitignore
Lab1_2331200153_CS405.iml
External Libraries
Scratches and Consoles

Run  RomantointegerTest ×
▶  RomantointegerTest 12 ms  1 test passed  1 test total, 12 ms
  ✓ testRomanToInt 12 ms

```

3. Consider Triangle.java file. Test the main method and isTriangle() method

Bug:

```

Point
Quadrilateral
QuadrilateralTest
Romantointeger
RomantointegerTest
Triangle
TriangleTest
Vector2D
.gitignore
Lab1_2331200153_CS405.iml
External Libraries
Scratches and Consoles

public static boolean isTriangle(double a, double b, double c) { 8 usages
    // error: All three conditions must use > and not >=
    if ((a + b > c) &&
        (a + c >= b) &&      //a+c >= b wrong
        (b + c > a)) {
    }
    if ((a + b > c) &&
        (a + c > b) &&
        (b + c > a)) {
        return true;
    }
    return false;
}

```

Test:

The screenshot shows the IntelliJ IDEA interface. On the left, the project structure for 'Lab1_2331200153_CS405' is visible, including files like CircleRadius, CircleRadiusTest, Fibonacci, FibonacciTest, Line, Point, Quadrilateral, QuadrilateralTest, RomantolInteger, RomantolIntegerTest, Triangle, TriangleTest, Vector2D, .gitignore, and Lab1_2331200153_CS405.iml. The 'TriangleTest' file is selected. On the right, the code editor displays the following Java code:

```

6 public void testTriangle() {
7     Triangle tri = new Triangle();
8     assertEquals( expected: true, tri.isTriangle(a: 3, b: 4, c: 5));
9     assertEquals( expected: false, tri.isTriangle(a: 1, b: 3, c: 5));
10    assertEquals( expected: false, tri.isTriangle(a: 6, b: 3, c: 2));
11    assertEquals( expected: false, tri.isTriangle(a: 2, b: 10, c: 4));
12    assertEquals( expected: false, tri.isTriangle(a: 1, b: 4, c: 5));
13    assertEquals( expected: false, tri.isTriangle(a: 6, b: 3, c: 3));
14    assertEquals( expected: false, tri.isTriangle(a: 6, b: 10, c: 4));
15 }
16
17

```

Below the code editor is the 'Run' tab, which shows the execution results for 'TriangleTest'. It indicates 1 test passed in 4 ms, with the specific test 'testTriangle' also taking 4 ms.

4. Look at the Fibonacci class. This class is an attempt at implementing the recursive method fib, which should generate the nth Fibonacci number. Create test cases for this class and run those test cases using Junit

Bug:

The screenshot shows the IntelliJ IDEA interface. On the left, the project structure for 'Lab1_2331200153_CS405' is visible, including files like .idea, out, and src. The 'src' folder contains CircleRadius, CircleRadiusTest, Fibonacci, FibonacciTest, Line, and Point. The 'Fibonacci' file is selected. On the right, the code editor displays the following Java code:

```

1 public class Fibonacci { 2 usages
2     public int fib(int n) { 5 usages
3         switch (n) {
4             case 0: return 0;
5             case 1: return 1;
6             // default: return (fib(n - 1) + fib(n - 2)) + 1; : error: extra +
7             // standard fibonacci sequence: F(n) = F(n-1) + F(n-2)
8             default: return (fib(n - 1) + fib(n - 2));
9         }
10    }
11

```

Test:

The screenshot shows the IntelliJ IDEA interface. On the left is the Project tool window with the following structure:

- Lab1_2331200153_CS405** - D:\CS
 - .idea
 - out
 - src**
 - CircleRadius
 - CircleRadiusTest
 - Fibonacci
 - FibonacciTest**
 - Line
 - Point
 - Quadrilateral
 - QuadrilateralTest
 - Romantolnteger
 - RomantolntegerTest
 - Triangle
 - TriangleTest
 - Vector2D
 - .gitignore
 - Lab1_2331200153_CS405.iml
- External Libraries**
- Scratches and Consoles**

The right pane displays the `FibonacciTest.java` file:

```

1 > import ...
3 ✓ public class FibonacciTest { no usages
4     @Test no usages
5 ✓     public void testFibonacci() {
6         Fibonacci fi = new Fibonacci();
7         assertEquals( expected: 2, fi.fib( n: 3));
8         assertEquals( expected: 8, fi.fib( n: 6));
9         assertEquals( expected: 34, fi.fib( n: 9));
10    }
11 }
12

```

The `Run` tool window at the bottom shows the results of the run:

- FibonacciTest**
- 1 test passed** | **1 test total, 10 ms**
- testFibonacci** | **10 ms**

5. The Quadrilateral class denotes a polygon with four sides. It has two methods, `isRectangle()` and `isSquare()`. Furthermore, it also uses the classes Point, Line and Vector2D. To find if the polygon is a rectangle, vectors and dot products are used to determine if every corner forms a right angle. To find if the polygon is a square, `isRectangle()` is used and check if the lengths of all sides are equal

Bug:

The screenshot shows the IntelliJ IDEA interface. The code in `Quadrilateral.java` contains a bug in the `isSquare()` method:

```

28     public Boolean isSquare() { 2 usages
29         // return (isRectangle() &&
30         //         l1.isSameLengthAs(l3)); -- // error: only checks opposite sides
31         return (isRectangle() &&
32                 l1.isSameLengthAs(l2) &&
33                 l2.isSameLengthAs(l3) &&
34                 l3.isSameLengthAs(l4));
35     }
36 }
37

```

The line `l1.isSameLengthAs(l3))` is highlighted in red, indicating a syntax error.

Test:

The screenshot shows the IntelliJ IDEA interface with the following details:

Project Structure:

- Project: Lab1_2331200153_CS405 - D:\CS
- File .idea
- Folder out
- Folder src
 - CircleRadius
 - CircleRadiusTest
 - Fibonacci
 - FibonacciTest
 - Line
 - Point
 - Quadrilateral
 - QuadrilateralTest
 - RomantolInteger
 - RomantolIntegerTest
 - Triangle
 - TriangleTest
 - Vector2D
- File .gitignore
- File Lab1_2331200153_CS405.iml

Code Editor:

```
1 > import ...
2
3 public class QuadrilateralTest { no usages
4     @Test no usages
5     public void testQuadrilateral(){
6         Point p1 = new Point( x: 0, y: 0);
7         Point p2 = new Point( x: 0, y: 2);
8         Point p3 = new Point( x: 2, y: 2);
9         Point p4 = new Point( x: 2, y: 0);
10
11        Quadrilateral qua1 = new Quadrilateral(p1, p2, p3, p4);
12        assertEquals( expected: true, qua1.isRectangle());
13        assertEquals( expected: true, qua1.isSquare());
14
15        Point p5 = new Point( x: 0, y: 0);
16        Point p6 = new Point( x: 0, y: 2);
17        Point p7 = new Point( x: 3, y: 2);
18        Point p8 = new Point( x: 3, y: 0);
19
20        Quadrilateral qua2 = new Quadrilateral(p5, p6, p7, p8);
21        assertEquals( expected: true, qua2.isRectangle());
22        assertEquals( expected: false, qua2.isSquare());
23    }
24}
25
```

Run Tab:

- Run QuadrilateralTest
- QuadrilateralTest 34 ms
- 1 test passed 1 test total, 34 ms
- C:\Users\Admin\.jdks\openjdk-24.0.1\bin\java.exe ...
- Process finished with exit code 0