

JUnit Testing

Programming Projects:

1) Testing a 'simple' Class

- Start Eclipse and create a new Java project called 'lab4'
- Download the associated **Triangle.java** file (also shown below)
- Add the file to the project you have created.
- Create a separate JUnit5 test class and write individual tests for all the methods of the **Triangle.java** class, including the constructors. Give each test method a sensible name.
- Execute the tests to see whether they all pass.
- Your tests should reveal at least two bugs in the code. Fix these so all your tests pass.
- Try to improve the tests by including some `@RepeatedTest` and `@ParameterizedTest` methods to check multiple parameter values.
- Look again at the test for the **setSides(int length, int other)** method. Can you write a test which shows the associated comment to be inaccurate in some cases?
- Run the tests again using the "Coverage As" option from within the Eclipse IDE, to ensure all methods have been exhaustively tested. i.e. have all possible branches of the `isIsosceles()`, `isEquilateral()` and `isScalene()` been tested?
- Do your tests check what happens if negative length values are passed to the constructors and methods? If not add these as additional tests.
- Can you write a test that causes the (working version) of the **int getPerimeter()** method to fail? If so can you suggest a way to change the method so it will no longer fail?

```

/**
 * A Triangle with three integer lengths
 *
 * @author mdixon
 */
public class Triangle {

    private int sideA, sideB, sideC;        // Length of side A, B and C

    /**
     * An Isosceles triangle is a triangle which has two equal length sides.
     *
     * @return true if the triangle is isosceles
     */
    public boolean isIsosceles() {
        return (sideA == sideB && sideA != sideC) ||
            (sideB == sideC && sideB != sideA) ||
            (sideC == sideA && sideC != sideB);
    }

    /**
     * An Equilateral triangle is a triangle which has three equal length sides.
     *
     * @return true if the triangle is equilateral
     */
    public boolean isEquilateral() {
        return (sideA == sideB && sideA == sideC);
    }

    /**
     * A scalene triangle is a triangle in which all three sides are in different lengths.
     *
     * @return true if the triangle is scalene
     */
    public boolean isScalene() {
        return (sideA != sideB && sideA != sideC && sideB != sideC);
    }

    /**
     * Sets each side of the triangle to a given value.
     *
     * @param a length of side A
     * @param b length of side B
     * @param c length of side C
     */
    public void setSides(int a, int b, int c) {
        sideA = Math.abs(a);
        sideB = Math.abs(b);
        sideC = Math.abs(c);
    }

    /**
     * Sets the triangle to be equilateral with all sides set to the given length
     *
     * @param length the length of side A, B and C
     */
    public void setSides(int length) {
        sideA = sideB = sideC = Math.abs(length);
    }

    /**
     * Sets the triangle to be scalene with two sides set to the same value.
     *
     * @param length the length of side A, B
     * @param other the length of side C
     */
    public void setSides(int length, int other) {

```

```

        sideA = sideB = Math.abs(length);
        sideC = Math.abs(other);
    }

    /**
     * Creates and returns a copy of the triangle. Changing the side lengths of the copy will
     * not effect the original triangle.
     *
     * @return a copy of the Triangle
     */
    public Triangle copy() {
        return new Triangle(sideA, sideB, sideC);
    }

    /**
     * Calculates the total length of all the sides.
     *
     * @return the total length of all the sides.
     */
    public int getPerimeter() {
        return sideA + sideB + sideC;
    }

    /**
     * Calculates the average length of the sides.
     *
     * @return the average length of the sides as an integer
     */
    public int getAverageLength() {
        return sideA + sideB + sideC / 3;
    }

    ////////////////////////////////////////
    /**
     * Constructor
     *
     * Creates a triangle with each side set to a given value.
     *
     * @param a length of side A
     * @param b length of side B
     * @param c length of side C
     */
    public Triangle(int a, int b, int c) {
        sideA = Math.abs(a);
        sideB = Math.abs(b);
        sideC = Math.abs(c);
    }

    /**
     * Constructor
     *
     * Creates an equilateral triangle with all sides set to the given length
     *
     * @param length the length of side A, B and C
     */
    public Triangle(int length) {
        sideA = sideB = sideC = Math.abs(length);
    }

    /**
     * Constructor
     *
     * Creates an equilateral triangle with all sides set to 1
     */
    public Triangle() {
        sideA = sideB = sideC = 1;
    }
}

```