SWE 637 – Assignment 13

Parameterized Unit Test (PUT) for a simple class that calculates sum of two integers

Shaeq Khan, skhan27@gmu.edu

JUnit Parameterized Test is a new feature introduced in JUnit4. It allows parameter values for unit testing. Some methods may have many input parameters to test, so writing a test case for each set of similar values to be tested can result in code bloat. JUnit now supports a way to group similar test values, pass and test them for a class without writing multiple tests.

`@RunWith(Parameterized.class)`
Annotation for a method which provides parameters to be injected into the test class constructor by Parameterized.

The parameters are defined by a static function which is marked with the annotation `@Parameters`.

`public static Collection<Object[]> data()`
The parameters are auto-boxed and stored in a `Collection` and returned using `asList()` which returns a fixed sized list backed by the specified array.

The code developed and executed is as follows –

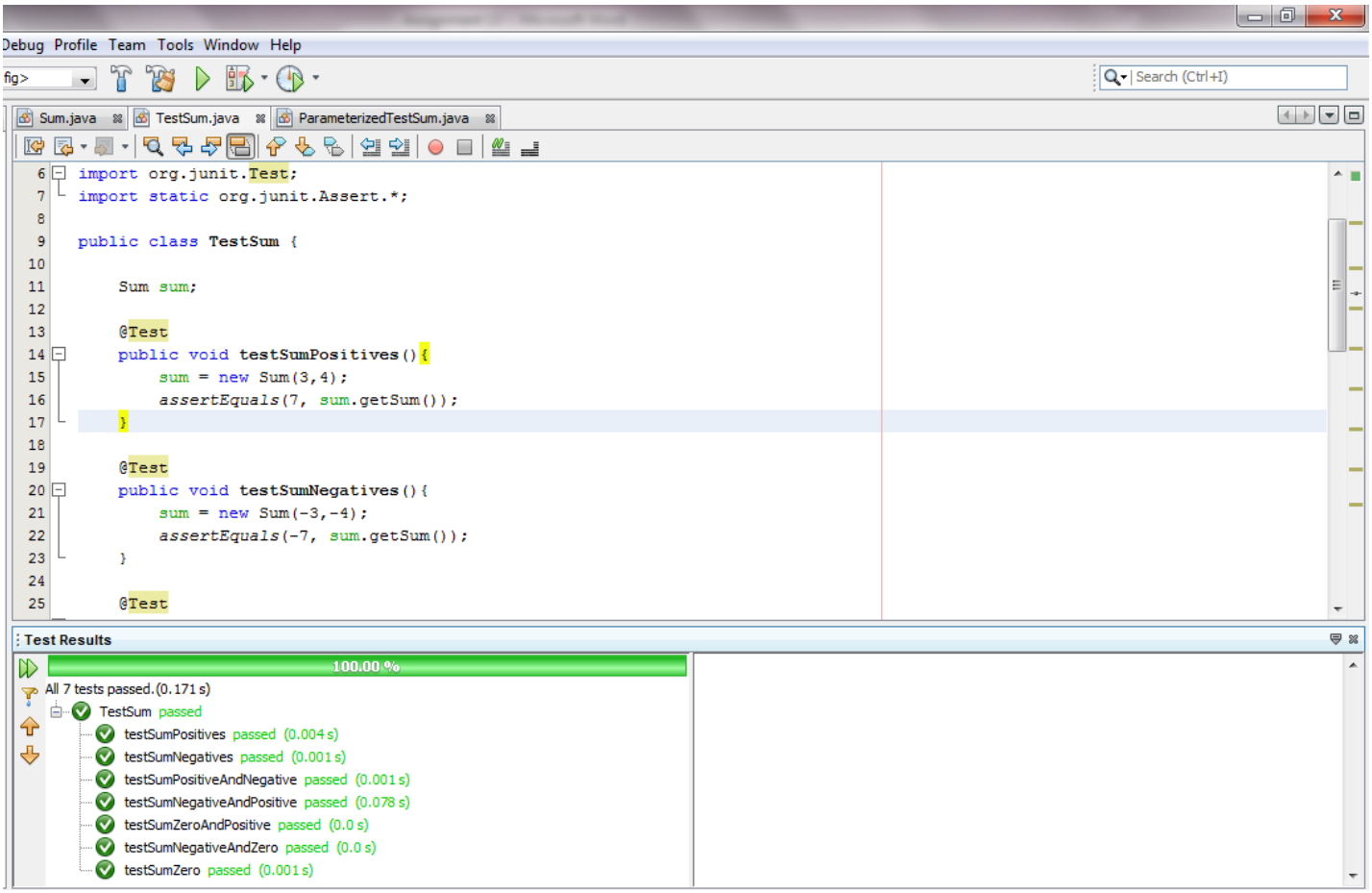
```
/*
 * SWE 637 Assignment 13
 * @author skhan27
 */

public class Sum {
    private int x;
    private int y;

    public Sum(int x, int y){
        this.x = x;
        this.y = y;
    }

    public int getSum(){
        return x+y;
    }
}
```

I wrote seven different test cases to test the sum of two numbers as follows –

```
/*
 * SWE 637 Assignment 13
 * @author skhan27
 */

import org.junit.Test;
import static org.junit.Assert.*;

public class TestSum {

    Sum sum;

    @Test
    public void testSumPositives(){
        sum = new Sum(3,4);
        assertEquals(7, sum.getSum());
    }

    @Test
    public void testSumNegatives(){
        sum = new Sum(-3,-4);
        assertEquals(-7, sum.getSum());
    }

    @Test
    public void testSumPositiveAndNegative(){
        sum = new Sum(3,-4);
        assertEquals(-1, sum.getSum());
    }

    @Test
    public void testSumNegativeAndPositive(){
        sum = new Sum(-3,4);
        assertEquals(1, sum.getSum());
    }

    @Test
    public void testSumZeroAndPositive(){
        sum = new Sum(0,3);
        assertEquals(3, sum.getSum());
    }
```

```
    @Test
    public void testSumNegativeAndZero(){
        sum = new Sum(-3,0);
        assertEquals(-3, sum.getSum());
    }

    @Test
    public void testSumZero(){
        sum = new Sum(-3,3);
        assertEquals(0, sum.getSum());
    }

} //end of TestSum
```

A snapshot of the output is shown below –



I converted these test cases into a Parameterized Test Case as follows –

```
/*
 * SWE 637 Assignment 13
 * @author skhan27
 */

import java.util.Arrays;
import java.util.Collection;
import org.junit.Test;
import org.junit.runners.Parameterized;
import org.junit.runners.Parameterized.Parameters;
import org.junit.runner.RunWith;
import static org.junit.Assert.*;

@RunWith(Parameterized.class)
public class ParameterizedTestSum {

    private Sum sum;
    private int result;

    public ParameterizedTestSum(int x, int y, int result) {
        sum = new Sum(x,y);
        this.result = result;
    }

    @Parameters
    public static Collection<Object[]> data(){
        Object[][] data = new Object[][] {
            {3,4,7},
            {-3,-4,-7},
            {3,-4,-1},
            {-3,4,1},
            {0,3,3},
            {-3,0,-3},
            {-3,3,0}
        };
        return Arrays.asList(data);
    }

    @Test
    public void testSum(){
        assertEquals(result, sum.getSum());
    }
}
```

A snapshot of the output is shown below –