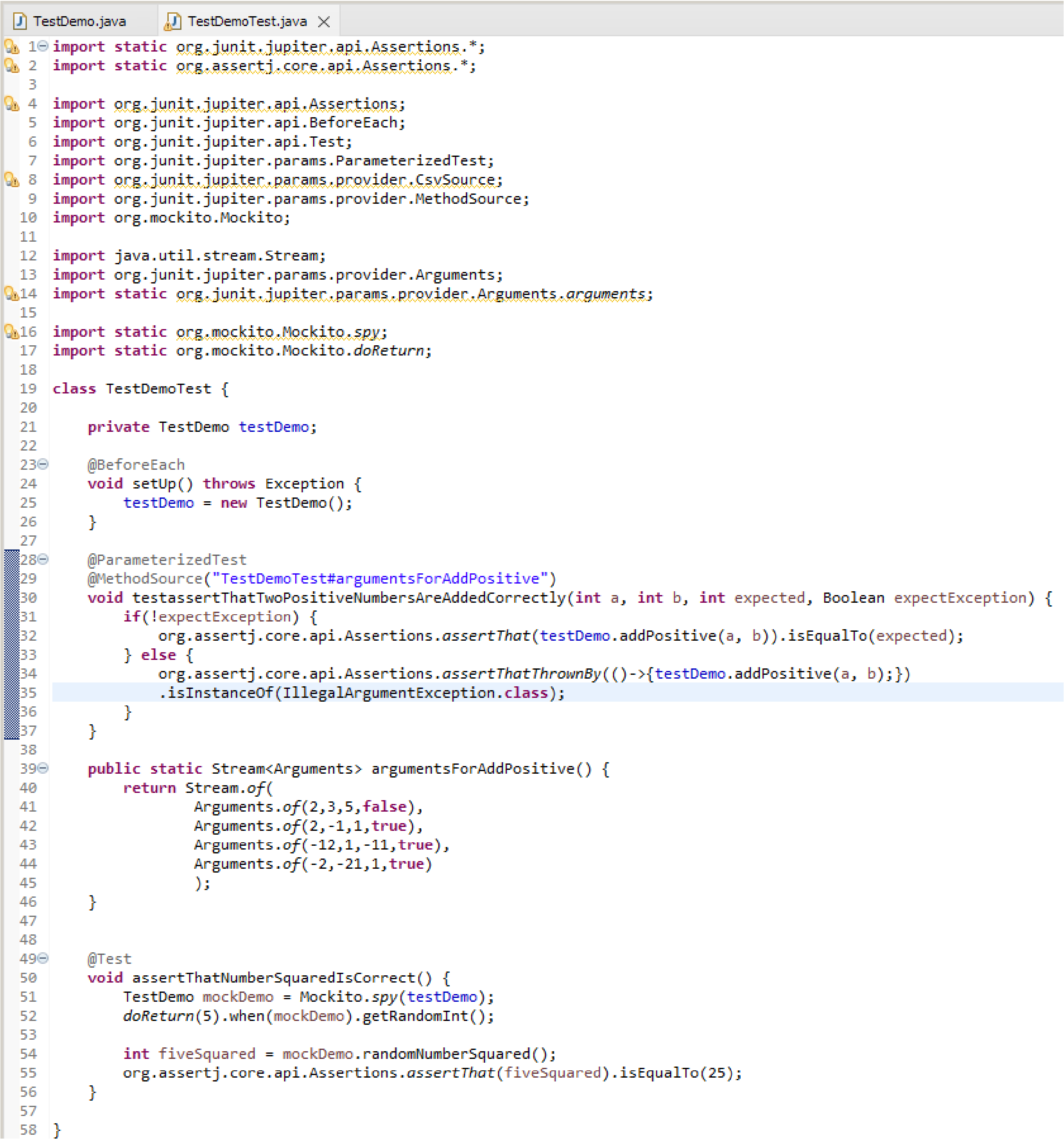
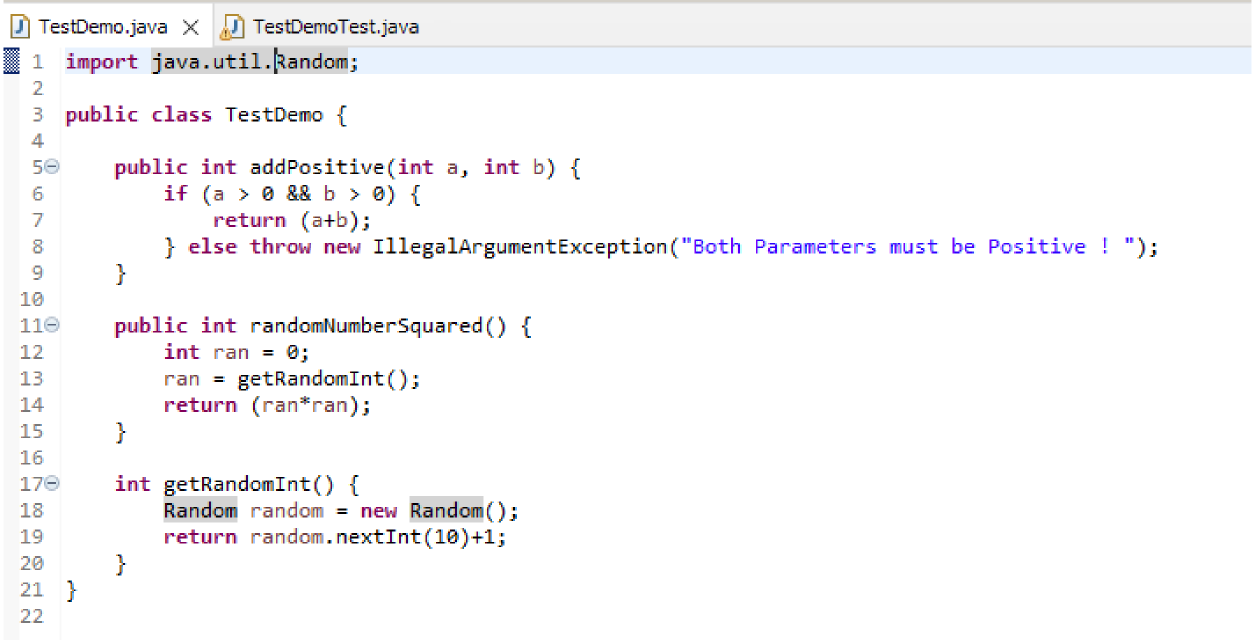
Relational Databases with MySQL Week 6 Coding Assignment

Screenshots of Code

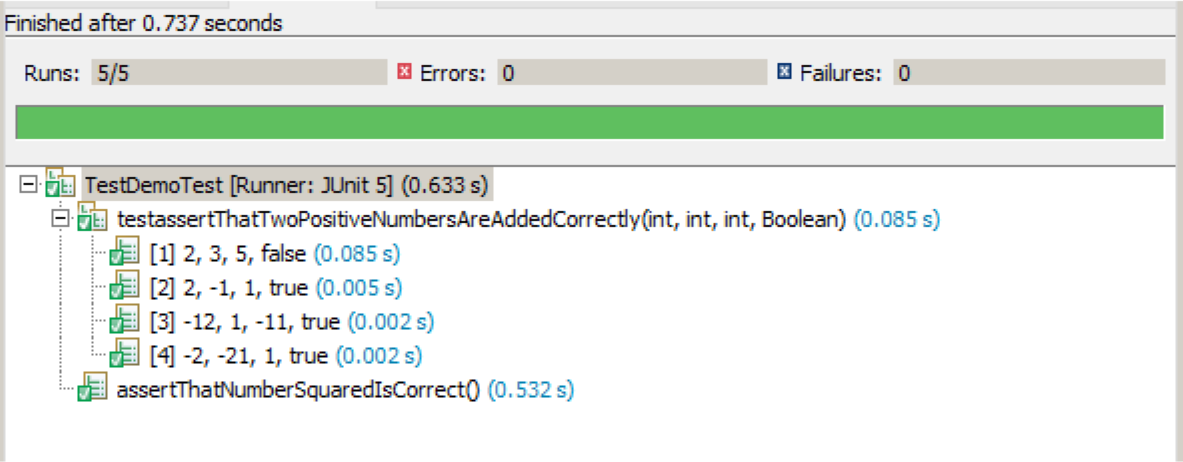


```java
import static org.junit.jupiter.api.Assertions.*;
import static org.assertj.core.api.Assertions.*;

import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.CsvSource;
import org.junit.jupiter.params.provider.MethodSource;
import org.mockito.Mockito;

import java.util.stream.Stream;
import org.junit.jupiter.params.provider.Arguments;
import static org.junit.jupiter.params.provider.Arguments.arguments;

import static org.mockito.Mockito.spy;
import static org.mockito.Mockito.doReturn;

class TestDemoTest {

    private TestDemo testDemo;

    @BeforeEach
    void setUp() throws Exception {
        testDemo = new TestDemo();
    }

    @ParameterizedTest
    @MethodSource("TestDemoTest#argumentsForAddPositive")
    void testassertThatTwoPositiveNumbersAreAddedCorrectly(int a, int b, int expected, Boolean expectException) {
        if(!expectException) {
            org.assertj.core.api.Assertions.assertThat(testDemo.addPositive(a, b)).isEqualTo(expected);
        } else {
            org.assertj.core.api.Assertions.assertThatThrownBy(()->{testDemo.addPositive(a, b);})
            .isInstanceOf(IllegalArgumentException.class);
        }
    }

    public static Stream<Arguments> argumentsForAddPositive() {
        return Stream.of(
                Arguments.of(2,3,5,false),
                Arguments.of(2,-1,1,true),
                Arguments.of(-12,1,-11,true),
                Arguments.of(-2,-21,1,true)
                );
    }


    @Test
    void assertThatNumberSquaredIsCorrect() {
        TestDemo mockDemo = Mockito.spy(testDemo);
        doReturn(5).when(mockDemo).getRandomInt();

        int fiveSquared = mockDemo.randomNumberSquared();
        org.assertj.core.api.Assertions.assertThat(fiveSquared).isEqualTo(25);
    }

}
```



```java
import java.util.Random;

public class TestDemo {

    public int addPositive(int a, int b) {
        if (a > 0 && b > 0) {
            return (a+b);
        } else throw new IllegalArgumentException("Both Parameters must be Positive ! ");
    }

    public int randomNumberSquared() {
        int ran = 0;
        ran = getRandomInt();
        return (ran*ran);
    }

    int getRandomInt() {
        Random random = new Random();
        return random.nextInt(10)+1;
    }
}

```

Screenshots of Running Application:



```
Finished after 0.737 seconds

Runs: 5/5        Errors: 0        Failures: 0

TestDemoTest [Runner: JUnit 5] (0.633 s)
    testassertThatTwoPositiveNumbersAreAddedCorrectly(int, int, int, Boolean) (0.085 s)
        [1] 2, 3, 5, false (0.085 s)
        [2] 2, -1, 1, true (0.005 s)
        [3] -12, 1, -11, true (0.002 s)
        [4] -2, -21, 1, true (0.002 s)
    assertThatNumberSquaredIsCorrect() (0.532 s)
```

URL to GitHub Repository:

https://github.com/david2joh/sqlWeek6.git