СПбНИУ ИТМО
Кафедра ВТ

Лабораторная работа №3
по дисциплине
«Основы программной инженерии»
Тестирование ПО

Выполнил
Широков О.И
гр.2120

Санкт-Петербург
г.2013

1. Текст задания

С помощью пакета JUnit провести модульное тестирование программы, реализующей задание к лабораторной работе #3 по дисциплине "Программирование интернет-приложений". Тестировое покрытие должно быть реализовано для всех классов приложения, тестовые сценарии должны учитывать все возможные варианты попадания или непопадания точек в область на координатной плоскости, а также возможность ввода пользователем некорректных данных.

Отчёт по работе должен содержать:

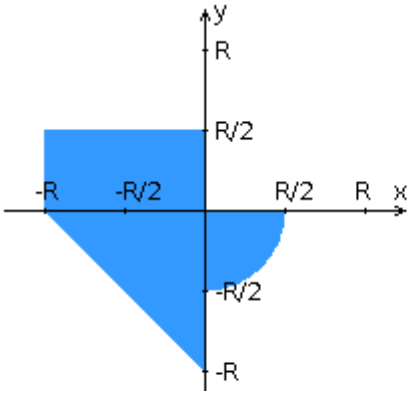
1. *Текст задания.*
2. *Требования к тестируемому приложению (из задания к лабораторной работе).*
3. *Описание тестового покрытия.*
4. *Исходный код классов тестов.*
5. *Описание результатов тестирования.*
6. *Выводы по работе.*

2. Требования к тестируемому приложению

На языке Java написать консольную программу, которая определяет, какие точки из набора A входят в заданную область S.

Приложение должно содержать следующие классы:

- *Класс Mark, представляющий точку с координатами X и Y типа double.*
- *Класс Silhouette, представляющий область с заданным параметром R, в котором должен быть реализован метод, возвращающий для заданной точки значение 1, если точка входит в область, и 0, если не входит. Попадание на границу области не считается попаданием в область.*
- *Класс Lab2, который получает параметр R типа float со стандартного ввода по запросу пользователя. Получение числа из строки реализовать с помощью метода Float.parseFloat().*



3. Описание тестового покрытия

1. *Проверка обработки неверного ввода аргументов командной строки*
2. *Проверка верной обработки точек входящих или не входящих в область*

4. Исходный код классов тестов

```
package RegLocation;

import org.junit.Before;
import org.junit.Test;
import org.junit.rules.ExpectedException;

import java.io.InputStream;

public class Lab2Test
{
    @Test(expected = NumberFormatException.class)
    public void MainTestWrongR()
    {
        Lab2.main(new String[]{"fefefefe", "{{12,34},{12,2}}"});
    }

    @Test(expected = NumberFormatException.class)
    public void MainTestWrongMarks()
    {
        Lab2.main(new String[]{"1", "{{fefefeuhfe}}"});
    }

    @Test(expected = ArrayIndexOutOfBoundsException.class)
    public void MainParseTest()
    {
        Lab2.main(new String[]{"1", "{{12}"});
    }

    @Test(expected = ArrayIndexOutOfBoundsException.class)
    public void MainTestEmptyArgs()
    {
        Lab2.main(new String[]{});
    }

    @Test(expected = NumberFormatException.class)
    public void MainTestNegativeR()
    {
        Lab2.main(new String[]{"-1", "{{12,2}}"});
    }
}
```

```
package RegLocation;

import org.junit.AfterClass;
import org.junit.BeforeClass;
import org.junit.Test;



public class SilhouetteTest
{
    final static float R = 3.0f;
    Silhouette mySilhouette = new Silhouette(R);

    public static Object[][] Marks  = new Object[][]
    {
            {new Mark(0,0),   0},
            {new Mark(0, -R), 0},
            {new Mark(-R, 0), 0},
            {new Mark(R/2,0), 0},
            {new Mark(0,R/2), 0},

            {new Mark(-R/2, R/4), 1},
            {new Mark(R/4, -R/4), 1},
            {new Mark(R,R),       0},
            {new Mark(-R/4,-R/4), 1},

            {new Mark(-R/2, R/2 + 0.1f), 0},
            {new Mark(-R/2, -R/2 - 0.1f), 0},
            {new Mark(R/4, (float)-Math.sqrt(0.25*Math.pow(R,2) - Math.pow(R/4,2)) - 0.1f), 0}
    };

    @Test
    public void testInRegion()
    {
        for(Object item : Marks)
        {
            Object[] ItemArr = (Object[])item;
            Mark mk = (Mark)ItemArr[0];
            Integer ExpectedValue = (Integer)ItemArr[1];
            System.out.println("Test Mark " + mk.toString() + "\n Expected value: " +
ExpectedValue.intValue());
            assert(mySilhouette.InRegion(mk) == ExpectedValue.intValue());
        }
    }

}
```

5. Описание результатов тестирования

Все тесты пройдены успешно.

6. Выводы

В процессе выполнения лабораторной работы были изучены основы модульного тестирования, основы использования пакета Junit, получены понятия о интеграционном и нагрузочном тестированиях.