

Университет ИТМО
Факультет программной инженерии и компьютерной техники
Кафедра вычислительной техники

Лабораторная работа № 1 по дисциплине
”Тестирование программного обеспечения”
Вариант: 756

Выполнили: Айтуганов Д. А.
Чебыкин И. Б.
Группа: Р3301

1 Задание

1. Для указанной функции провести модульное тестирование разложения функции в степенной ряд. Выбрать достаточное тестовое покрытие.
2. Провести модульное тестирование указанного алгоритма. Для этого выбрать характерные точки внутри алгоритма, и для предложенных самостоятельно наборов исходных данных записать последовательность попадания в характерные точки. Сравнить последовательность попадания с эталонной.
3. Сформировать доменную модель для заданного текста. Разработать тестовое покрытие для данной до- менной модели.

2 Выполнение

2.1 Функция $\cos(x)$

Для тестирования были выделены точки, в которых функция меняется одинаково.

```

import org.junit.After;
import org.junit.Assert;
import org.junit.Before;
import org.junit.Test;

import java.util.HashMap;
import java.util.Map;
import java.util.concurrent.Callable;

/**
 * Created by daituganov on 11.03.17.
 */
public class CosinusTest extends Assert {
    private HashMap<Double, Double> arrayTestValues = new HashMap<>();
    private HashMap<Double, Double> arrayWrongTestValues = new HashMap<>();
    private static final double DELTA = 0.001;

    @Before
    public void setUp() {
        //hashMap, key - table result, value - input value
        //zero check
        arrayTestValues.put((double) 1, (double) 0);

        //right side check
        arrayTestValues.put(Math.sqrt(3) / 2, Math.PI / 6);
        arrayTestValues.put((double) 0, Math.PI / 2);
        arrayTestValues.put(-Math.sqrt(2) / 2, 3 * Math.PI / 4);

        //parity check
        arrayTestValues.put(Math.sqrt(3) / 2, -Math.PI / 6);

        //test coverage boundaries
        arrayTestValues.put((double) -1, -Math.PI);
        arrayTestValues.put((double) -1, Math.PI);

        //test NaN, Infinity
        arrayTestValues.put(Double.NaN, Double.NaN);
        arrayTestValues.put(Double.NaN, Double.POSITIVE_INFINITY);

        //test values outside the coverage boundaries
        arrayWrongTestValues.put(Math.sqrt(3) / 2, 7 * Math.PI / 6);
        arrayWrongTestValues.put(Math.sqrt(3) / 2, -7 * Math.PI / 6);
    }

    @After
    public void tearDown() {

```

```
        arrayTestValues.clear();
    }

    @Test
    public void testCos() {
        double expected, actual;
        for(Map.Entry entry : arrayTestValues.entrySet()) {
            expected = (Double) entry.getKey().doubleValue();
            actual = Cosinus.cos( (Double) entry.getValue().doubleValue());
            assertEquals(expected, actual, DELTA);
        }
        for(Map.Entry entry : arrayWrongTestValues.entrySet()) {
            expected = (Double) entry.getKey().doubleValue();
            actual = Cosinus.cos( (Double) entry.getValue().doubleValue());
            assertEquals(expected, actual, DELTA);
        }
    }
}
```

2.2 **Расширяющееся дерево**

Данное дерево, основанное на бинарном, позволяет быстро осуществлять доступ к недавно добавленным элементам.

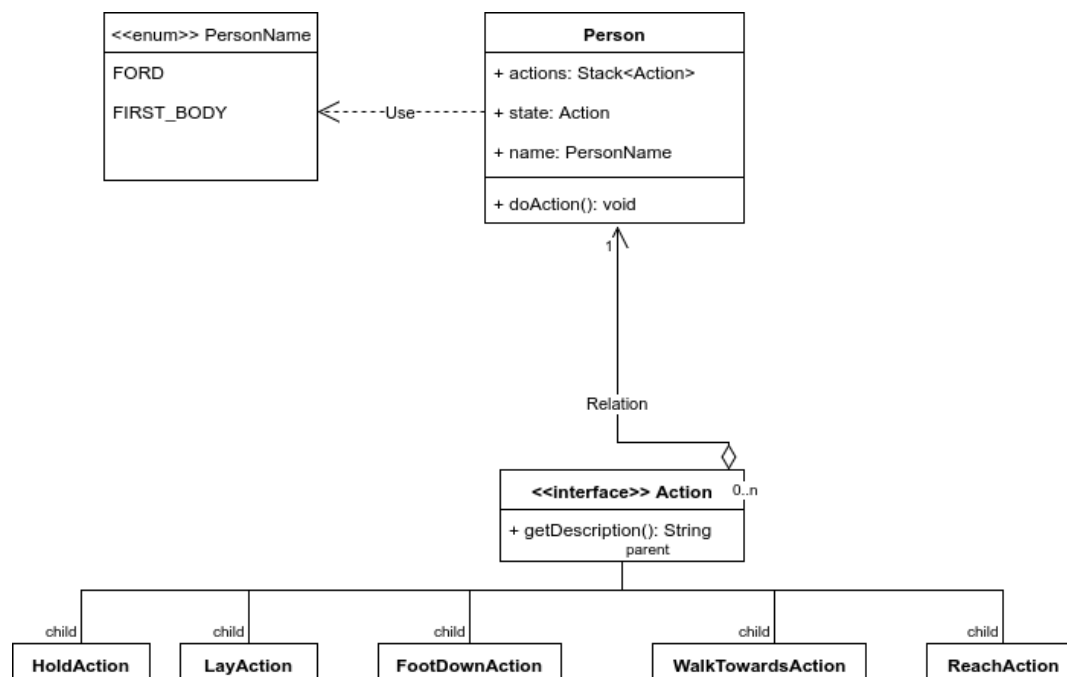
Это реализуется за счет перемещения вершины в корень. Таким образом можно рассмотреть три случая:

- Если родительская вершина – корень, тогда достаточно поменять их местами.
- Если родительская вершина не корень, и если она и текущая вершина являются левыми или правыми сыновьями, тогда сначала меняются местами родитель и родитель родителя, а затем текущая вершина меняется с родителем, становясь корнем.
- Если родительская вершина и текущая не одинаково левые или правые сыновья, тогда они меняются местами, затем текущая вершина меняется с родителем родительской вершины.

3 **Описание предметной области**

Медленно и осторожно он подошел к первому телу. Оно лежало обнадеживающе тихо, и продолжало лежать так, когда он приблизился к нему вплотную и поставил ногу на килобац, который оно все еще сжимало скрюченными пальцами.

4 UML диаграмма



5 Вывод

В ходе выполнения данной лабораторной работы было проведено тестирование разработанных программных модулей с использованием средств JUnit4. Данная библиотека предоставляет удобные средства тестирования.