Министерство науки и высшего образования РФ

Севастопольский государственный университет

Кафедра информационных систем

ОТЧЕТ

по лабораторной работе №3

Исследование способов модульного тестирования программного обеспечения

по дисциплине «Тестирование программного обеспечения»

Выполнил:

Студент группы ИС/б 17-2-о

Черняев Н.Г.

Проверил:

Строганов В.А.

г. Севастополь 2019

**Цель работы**

Исследовать основные подходы к модульному тестированию программного обеспечения. Приобрести практические навыки составления модульных тестов для объектно-ориентированных программ.

**Вариант №18**

Выбрать в качестве тестируемого один из классов, спроектированных в лабораторной работе № 1.

Составить спецификацию тестового случая для одного из методов выбранного класса.

Реализовать тестируемый класс и необходимое тестовое окружение.

Выполнить тестирование с выводом результатов на экран и сохранением в log-файл.

**Ход работы**

Выберем класс LinesSumWithoutNegative.

**package** lab3;

**public** **class** LinesSumWithoutNegative {

**public** **int** Solve(**int**[][] matrix) {

**int** Sum = 0;

**for** (**int** i = 0; i < matrix.length; i++) {

**int** CurrentSum = 0;

**for** (**int** j = 0; j < matrix.length; j++) {

**if** (matrix[i][j] >=0 )

CurrentSum += matrix[i][j];

**else** {

CurrentSum = 0;

**break**;

}

}

Sum += CurrentSum;

}

**return** Sum;

}

}

Составить спецификацию тестового случая для метода Solve и реализуем тестируемый класс и необходимое тестовое окружение.

Класс LinesSumWithoutNegativeTest.

**package** lab3;

**import** java.io.IOException;

**import** java.util.ArrayList;

**public** **class** LinesSumWithoutNegativeTest {

**public** **static** **void** TestsComparision(ArrayList<Test> tests) **throws** IOException {

FileLogger fileLogger = **new** FileLogger();

**for** (Test t : tests) {

String result;

LinesSumWithoutNegative obj = **new** LinesSumWithoutNegative();

**if** (obj.Solve(t.Input) == t.Result) {

result = t.Name + " is Correct";

}**else** {

result = t.Name + " is Wrong, expected: " + Integer.*toString*(t.Result) + " actual: " + Integer.*toString*(obj.Solve(t.Input));

}

fileLogger.Log(result);

}

fileLogger.Close();

}

}

Класс Test.

**package** lab3;

**public** **class** Test {

**public** **int** Name;

**public** **int**[][] Input;

**public** **int** Result;

}

Класс FileLogger.

**package** lab3;

**import** java.io.BufferedWriter;

**import** java.io.File;

**import** java.io.FileWriter;

**import** java.io.IOException;

**public** **class** FileLogger {

**private** String filename = "log.txt";

**private** File fileout = **null**;

**private** BufferedWriter bufferedWriter = **null**;

**private** FileWriter fileWriter = **null**;

**public** **void** Log(String str) **throws** IOException {

fileout = **new** File("E:\\" + filename);

**if** (!fileout.exists()) {

fileout.createNewFile();

}

fileWriter = **new** FileWriter(fileout, **true**);

bufferedWriter = **new** BufferedWriter(fileWriter);

bufferedWriter.write(str + "\r\n");

bufferedWriter.flush();

}

**public** **void** Close() **throws** IOException {

bufferedWriter.close();

}

}

Класс Program.

**package** lab3;

**import** java.io.IOException;

**import** java.util.ArrayList;

**public** **class** Program {

**public** **static** **void** main(String[] args) **throws** IOException {

Test TestSet1 = **new** Test();

Test TestSet2 = **new** Test();

Test TestSet3 = **new** Test();

Test TestSet4 = **new** Test();

Test TestSet5 = **new** Test();

TestSet1.Name = 1;

TestSet1.Input = **new** **int**[][]{{2}};

TestSet1.Result = 2;

TestSet2.Name = 2;

TestSet2.Input = **new** **int**[][]{{2,2},{3,1}};

TestSet2.Result = 8;

TestSet3.Name = 3;

TestSet3.Input = **new** **int**[][]{{29,32,-2},{3,1,1},{1,2,3}};

TestSet3.Result = 11;

TestSet4.Name = 4;

TestSet4.Input = **new** **int**[][]{{20,92,-2},{-3,31,12},{6,-7,38}};

TestSet4.Result = 0;

TestSet5.Name = 5;

TestSet5.Input = **new** **int**[][]{{0,9,3},{7,1,-126},{46,17,-9}};

TestSet5.Result = 7;

ArrayList<Test> TestSet = **new** ArrayList<Test>();

TestSet.add(TestSet1);

TestSet.add(TestSet2);

TestSet.add(TestSet3);

TestSet.add(TestSet4);

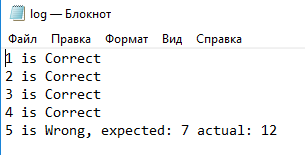
TestSet.add(TestSet5);

LinesSumWithoutNegativeTest.*TestsComparision*(TestSet);

}

}

Результаты(содержимое файла log.txt):



**Вывод**

Вданной лабораторной работе были исследованы основные подходы к модульному тестированию программного обеспечения, а также приобретены практические навыки составления модульных тестов для объектно-ориентированных программ.