Лабораторная работа No3

**Исследование способов модульного тестирования программного обеспечения**

**Цель работы**

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

**Вариант – 8**

1. Выбрать в качестве тестируемого один из классов, спроектированных в лабораторных работах NoNo 1 – 4.

2.Составить спецификацию тестового случая для одного из методов выбранного класса, как показано в разделе 2.2.

3. Реализовать тестируемый класс и необходимое тестовое окружение на языке С#.

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

5. Проанализировать результаты тестирования, сделать выводы.

**Ход работы**

**Спецификация тестового случая:**

В качестве одного из методов класса был выбран метод поворота матрицы k раз.

1. Название тестируемого класса: SquareMatrix

2) Название тестового случая: SquareMatrixTest.1

3) Описание тестового случая: Тест проверяет правильность работы метода rotate– получение перевёрнутой k раз матрицы. В тесте подаются следующие значение исходных матриц:

a) [[0,0], [0,1]],

б) [[], [[0,0], [0,1]],

в) [[0,0], [0,1], [0,3], [0,4]],

г) [[0]]

В одном из тестов передаётся k отрицательное, что является запрещённым элементом.

**Код программы:**

var fs = require('fs');

class TPO {

// MARK: FIRST TASK

// Rotate by 90 \* k degrees function

rotate(matrix, k) {

if(k <0 ) return 'k is wrong'

let isSquare = this.isSquareMatrix(matrix)

if (isSquare == 'neverno') return 'matrix is no square'

var result = [...matrix];

let size = matrix[0].length

var delta = matrix

for (let d = 0; d < k; d++) {

for (let i = 0; i < size; i++) {

for (let j = 0; j < size; j++) {

result[i][j] = delta[size - j - 1][i]

}

}

delta = result

}

return result

}

isSquareMatrix(matrix) {

for(let item in matrix){

if(matrix[item].length != matrix[Number(item + 1)].length){

return 'neverno'

}

}

}

// MARK: SECOND TASK

is5CharactersE(stroka) {

if(stroka.includes("eeeee")){

return true

}

else{

return false

}

}

// MARK: THREE TASK

readFile(name){

var data = fs.readFileSync(name, 'utf8');

let count = 0;

for(let stroka of data.split('\n')){

let curStroke = String(stroka);

if(curStroke.length <= 1) count++;

}

return count

}

}

const Test = new TPO()

// Один тест на количество

let matrix = [

[0,0,2],

[0,1]

]

let k = 4;

let rotated = Test.rotate(matrix, k)

console.log(`Rotateded matrix`, rotated)

// В начале, в конце, в середине, присутствуют не все e, отсутствуют e

let str = 'i tut'

console.log('Is have 5 e', Test.is5CharactersE(str) )

// В начале, в конце, в середине, присутствуют не все e, отсутствуют e

console.log('Count space string',Test.readFile('file.txt'))

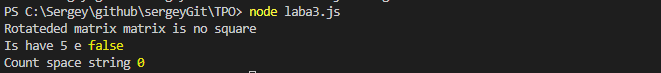


Рисунок 1 – Вывод программы

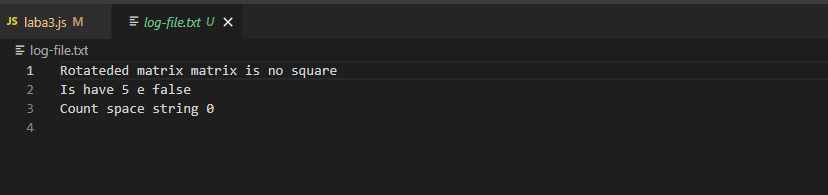


Рисунок 2 – log-file.txt

**Контрольные вопросы**

1. Для чего применяется тестирование программного обеспечения?

Цель тестирования программных модулей состоит в том, чтобы удостовериться, что каждый модуль соответствует своей спецификации.

1. Какие существуют разновидности тестирования?

Существует два основных подхода к тестированию классов: просмотр (review) программного кода и тестовые прогоны.

3. Что такое модульное тестирование?

Тестирование модулей обычно представляет собой некоторое сочетание проверок и прогонов тестовых случаев. Можно составить план тестирования модуля, в котором учесть тестовые случаи и построение тестового драйвера.

4. Что понимается под модулем при модельном тестировании?

В процедурно-ориентированном программировании модулем называется процедура или функция, иногда группа процедур.

5. Каков порядок мольного тестирования классов?

Тестирование классов обычно выполняется путем разработки тестового драйвера, который создает экземпляры классов и окружает эти экземпляры соответствующей средой (тестовым окружением), чтобы стал возможен прогон соответствующего тестового случая. Драйвер посылает сообщения экземпляру класса в соответствии со спецификацией тестового случая, а затем проверяет исход этих сообщений.

6. Какие разделы включает спецификация тестового случая?

Спецификация тестового случая должна включать следующие пункты:

1) Название тестируемого класса: TCommand.

2) Название тестового случая: TCommandTest1.

3) Описание тестового случая

**Вывод:**

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