Министерство науки и высшего образования РФ

Севастопольский государственный университет

Кафедра информационных систем

ОТЧЕТ

по лабораторной работе №1

Исследование способов анализа областей эквивалентности и построения тестовых последовательностей

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 способы анализа областей эквивалентности входных данных для тестирования программного обеспечения. Приобрести практические навыки составления построения тестовых последовательностей.

**Вариант №18**

Задача 1. Дана целочисленная квадратная матрица. Определить сумму элементов в тех строках, которые не содержат отрицательных элементов.

Задача 2. Дана строка. Удалить в данной строке символы, стоящие до заданной позиции.

Задача 3. Программа, которая считывает текст из файла и выводит его на экран, меняя местами каждые две соседних буквы.

Для каждой из программ необходимо:

1) Написать программу, выполняющую заданные действия.

2) Определить области эквивалентности входных данных.

3) Составить примеры тестовых последовательностей.

**Ход работы**

**Задача 1.**

**Составим программу.**

int M[N,N]; //Целочисленная квадратная матрица

int Sum=0; //Сумма

for(int i=0; i<N; i++)

Sum = 0;

for(int j=0; j<N; j++)

if(M[i,j]>=0)

Sum+=M[i,j];

else

Sum = 0;

break;

**Определим области эквивалентности.**

1)По размеру матрицы:

а)Матрица состоит из одного элемента;

б)Матрица состоит более чем из одного элемента.

2)По наличию и расположению положительных элементов:

а)Все элементы отрицательные;

б)Существует один положительный элемент;

в)Существует одна положительная строка;

г)Существует несколько положительных строк.

**Составим тестовые последовательности.**

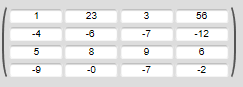
1а, 2а) [-1];

1а, 2б) [2];

1б,2а) 

1б,2б) 

1б,2в) 

1б,2г) 

**Задача 2.**

**Составим программу.**

String deleteSomeCharacters(String str, int from) {

return str.substring(from,(str.length()-1));

}

**Определим области эквивалентности.**

1) По размеру строки:

а) Пустая строка;

б) Длина строки - 1;

в) Длина строки >1.

2)По вводимой позиции:

а) Номер позиции выходит за пределы множества индексов строки;

б)Номер позиции является первым элементом строки;

в)Номер позиции является последним элементом строки;

г)Номер позиции является средним элементом строки.

**Составим тестовые последовательности.**

1а) ""

1б,2а) "А",3.

1б,2б) "В",0.

1б,2в) "С",0.

1в,2а) "acd",6.

1в,2б) "erg",0.

1в,2в) "awdf",3.

1в,2г) "qethfs",2.

**Задача 3.**

**Составим программу.**

String str = "";

Scanner in = new Scanner(new File(ПУТЬ К ФАЙЛУ));

while(in.hasNext())

str += in.nextLine() + "\r\n";

in.close();

String temp;

for(int i=0;i<str.lentgh();i+=2){

temp=str[i];

str[i]=s[i+1];

str[i+1]=temp;

}

**Определим области эквивалентности.**

1) По содержимому файла:

а) Пустая строка;

б) Длина строки - 1;

в) Длина строки >1, нечетное количество символов;

г) Длина строки >1, четное количество символов.

**Составим тестовые последовательности.**

1а) ""

1б) "А".

1в) "acd".

1г) "awdf".

**Вывод**

Вданной лабораторной работе были исследованы способы анализа областей эквивалентности входных данных для тестирования программного обеспечения. Были также приобретены практические навыки составления построения тестовых последовательностей.