

Министерство науки и высшего образования Российской Федерации
Севастопольский государственный университет
Кафедра ИС

Отчет
по лабораторной работе №2
«Исследование способов структурного тестирования программного
обеспечения»
по дисциплине
«ТЕСТИОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

Выполнил студент группы ИС/б-17-2-о
Горбенко К. Н.
Проверила
Тлуховская Н.П.

Севастополь
2019

1 ЦЕЛЬ РАБОТЫ

Исследовать основные подходы к структурному тестированию программного обеспечения. Приобрести практические навыки построения графа потоков управления и определения независимых ветвей программы.

2 ЗАДАНИЕ НА РАБОТУ

Для **варианта № 5** заданы следующие требования к программам:

1. Дана целочисленная квадратная матрица. Определить сумму элементов в тех столбцах, которые не содержат отрицательных элементов.
2. Дана строка. Удалить в данной строке символ, стоящий на заданной позиции.
3. Программа, которая считывает текст из файла и выводит его на экран, меняя местами каждые два соседних слова.

Для каждой из программ необходимо:

1. Написать программу, выполняющую заданные действия.
2. Построить граф потоков управления.
3. Вычислить цикломатическое число для построенного графа потоков управления.
4. Определить независимые ветви программы.

3 ТЕКСТ ПРОГРАММНЫХ МОДУЛЕЙ

3.1 Операции над матрицами

Была написана следующая программа:

```
1 public static class MatrixOperations
2 {
3     public static IEnumerable<int> GetSumOfColumnsWithoutNegativeElements(int
        [,] matrix)
4     {
5         if (matrix == null) throw new ArgumentNullException($"{nameof(matrix)}
            instance was null");
6         if (matrix.GetLength(0) != matrix.GetLength(1)) throw new
            ArgumentException($"Only square martixes are allowed");
7
8         return matrix
```

```

9         .GetColumns()
10        .Where(column => column.All(items => items >= 0))
11        .Select(column => column.Sum());
12    }
13 }
14
15 public static IEnumerable<int[]> GetColumns(this int[,] array)
16 {
17     if (array == null) throw new ArgumentNullException($"{nameof(array)}
18         instance was null");
19
20     for (var j = 0; j < array.GetLength(1); j++)
21     {
22         var column = new int[array.GetLength(0)];
23
24         for (var i = 0; i < array.GetLength(0); i++)
25             column[i] = array[i, j];
26
27         yield return column;
28     }
29 }

```

3.2 Операции над строками

Была написана следующая программа:

```

1 public static class StringOperations
2 {
3     public static string RemoveAt(string source, int position) => source.Remove
4         (position, 1);
5 }

```

3.3 Операции над текстом

```

1 public class TextOperations
2 {
3     private readonly StreamReader streamReader;
4     private readonly string[] delimiters = { " ", ".", ",", "?", "!", "(", ")",
5         ":", ";", Environment.NewLine };
6
7     public TextOperations(StreamReader streamReader)
8         => this.streamReader = streamReader ?? throw new ArgumentNullException(
9             $"{nameof(streamReader)} instance was null");
10
11     public string ReverseEveryTwoWords()
12     {
13         var source = streamReader.ReadToEnd();
14
15         // ...
16     }
17 }

```

```

12
13     return string.Join(" ", source
14                             .Split(delimiters, StringSplitOptions.
15                                 RemoveEmptyEntries)
16                             .Select((word, i) => new { Value = word,
17                                 Index = i })
18                             .GroupBy(x => x.Index / 2)
19                             .Select(group => group.Select(x => x.Value)
20                                 .Reverse())
21                             .SelectMany(x => x)
22     );
23 }
24 }

```

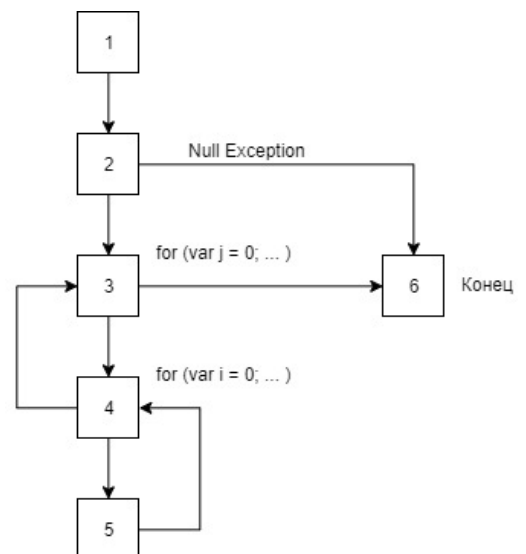
4 ТЕСТИРОВАНИЕ ПРОГРАММ

4.1 Тестирование программы № 1

Для программы работы над массивами составим граф потоков управления.



(a) Метод `GetSumOfColumns...`



(b) Метод `GetColumns`

Рисунок 1 – Графы потоков управления первой программы

Вычислим цикломатическое число графа :

$$C(G_a) = 6 - 5 + 2 = 3.$$

1. 1, 2, 5.
2. 1, 2, 3, 5.
3. 1, 2, 3, 4, 5.

Вычислим цикломатическое число графа:

$$C(G_b) = 8 - 6 + 2 = 4.$$

1. 1, 2, 6.
2. 1, 2, 3, 6.
3. 1, 2, 3, 4, 3, 6.
4. 1, 2, 3, 4, 5, 4, 3, 6.

4.2 Тестирование программы № 2

Для программы работы над строками составим граф потоков управления.

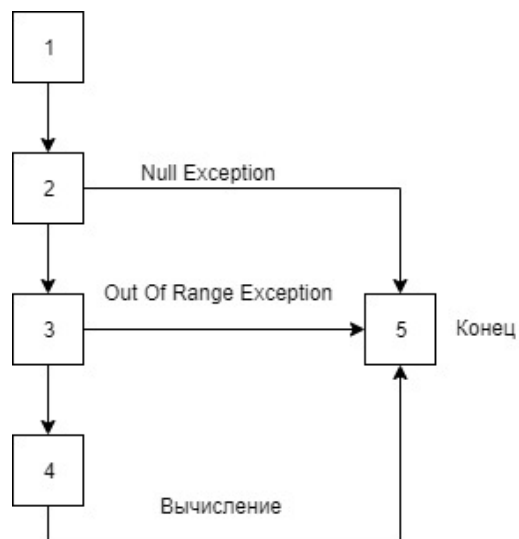


Рисунок 2 – Граф потоков управления второй программы

Вычислим цикломатическое число графа :

$$C(G) = 6 - 5 + 2 = 3.$$

1. 1, 2, 5.
2. 1, 2, 3, 5.
3. 1, 2, 3, 4, 5.

4.3 Тестирование программы № 3

Для программы работы над файлами составим граф потоков управления.

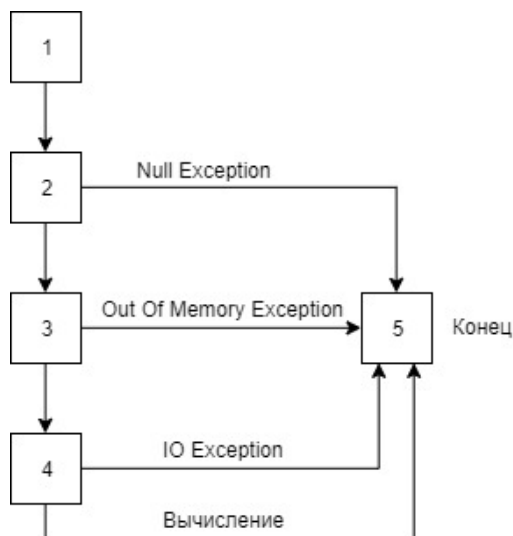


Рисунок 3 – Граф потоков управления третьей программы

Вычислим цикломатическое число графа :

$$C(G) = 7 - 5 + 2 = 4.$$

1. 1, 2, 5.
2. 1, 2, 3, 5.
3. 1, 2, 3, 4, 5.
4. 1, 2, 3, 4, 5.

5 ВЫВОДЫ

В ходе лабораторной работы было изучено структурное тестирование ПО (метод белого ящика) с использованием графа потоков управления.

Преимуществом такого способа тестирования является уменьшение необходимой трудоемкости для тестирования программ. Обнаружились следующие недостатки:

1. Недостаточная глубина тестирования.
2. Невозможность использования способа для крупных программ.

Из-за слабого покрытия возможных вариантов входных данных, структурное тестирование, по моему мнению, не является достаточным способом тестирования

программных модулей. Вместо него или, возможно, в дополнение к нему можно применить тестирование черным ящиком.