# Отчет о структурном тестировании

1. *Указание названия и функций тестируемого приложения.*

Приложение: **MinMaxLen.**

Функционал: позволяет определять в введенном предложении слова минимальной и максимальной длины.

1. *Предоставление кода тестируемого метода.*

/\*1\*/

[TestClass]

public class MinMaxLenTest

{

[TestMethod]

public void MinTest()

{

string input = "Люблю грозу в начале мая";

string[] str;

int max, min;

MinMaxLen.Proverka(input, out str, out max, out min);

Assert.AreEqual(2, min);

Assert.AreEqual(3, max);

Assert.IsTrue(str.SequenceEqual(new String[] { "Люблю", "грозу", "в", "начале", "мая" }));

}

/\*2\*/

[TestMethod]

public void MinTest2()

{

string input = "";

string[] str;

int max, min;

MinMaxLen.Proverka(input, out str, out max, out min);

Assert.AreEqual(0, min);

Assert.AreEqual(0, max);

Assert.IsTrue(str.SequenceEqual(new String[] {}));

}

/\*3\*/

[TestMethod]

public void MinTest3()

{

string input = "!";

string[] str;

int max, min;

MinMaxLen.Proverka(input, out str, out max, out min);

Assert.AreEqual(0, min);

Assert.AreEqual(0, max);

Assert.IsTrue(str.SequenceEqual(new String[] { }));

}

/\*4\*/

[TestMethod]

public void MinTest4()

{

string input = "Seattle is a big city";

string[] str;

int max, min;

MinMaxLen.Proverka(input, out str, out max, out min);

Assert.AreEqual(2, min);

Assert.AreEqual(0, max);

Assert.IsTrue(str.SequenceEqual(new String[] { "Seattle", "is", "a", "big", "city" }));

}

/\*5\*/

[TestMethod]

public void MinTest5()

{

string input = "В городе было тепло";

string[] str;

int max, min;

MinMaxLen.Proverka(input, out str, out max, out min);

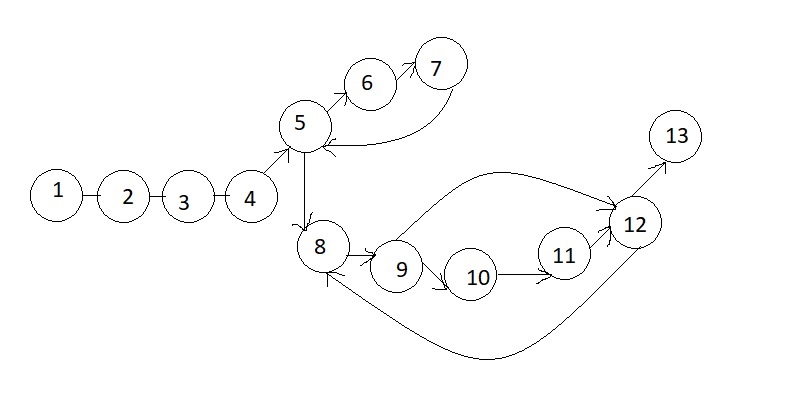
Assert.AreEqual(0, min);

Assert.AreEqual(1, max);

Assert.IsTrue(str.SequenceEqual(new String[] { "В", "городе", "было", "тепло" }));

}

1. *Наличие изображения потокового графа.*



1. *Наличие цикломатической сложности потокового графа.*

Кол-во дуг: 15

Кол-во узлов: 13

15-13+2 = 4

F = 2(кол-во предикатных узлов) +1 = 3

1. *Наличие определенного базового множества независимых линейных путей.*

3 предикатных узла + 1 = **4 независимых линейных пути**

**1 путь:** 1-2-3-4-5-8-9-12-13

**2 путь:** 1-2-3-4-5-6-7-8-9-12-13

**3 путь:** 1-2-3-4-5-6-7-8-9-10-11-12-13

**4 путь:** 1-2-3-4-5-6-7-8-9-10-11-12-8-9-10-11-12-13

1. *Заполненная таблица для тестовых вариантов.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №  теста | Тестовая строка | Эталонный вариант  max | Эталонный вариант  min | Полученный результат  max | Полученный результат  min |
| 1 | Люблю грозу в начале мая | начале | в | начале | в |
| 2 | Пустая строка | Пустая строка | Пустая строка | Пустая строка | Пустая строка |
| 3 | ! | Пустая строка | Пустая строка | Пустая строка | Пустая строка |
| 4 | Seattle is a big city | Seattle | a | Seattle | a |

Вывод: таким образом, созданные структурные тесты демонстрируют успешную работу программы при разных вариантах входных данных.