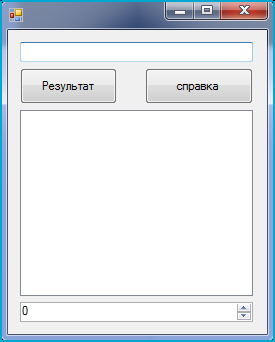
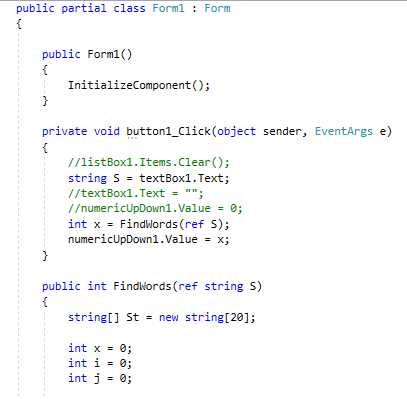
# Отчет о функциональном тестировании

1. Указание название и функций тестируемого приложения.

"Сколько слов с буквы Т?" Программа выполняет поиск слов, начинающихся с букв «t» и «Т»

1. Снимок интерфейса приложения.



1. Описание классов эквивалентности для приложения

В сумме получилось 3 класса эквивалентности. Класс эквивалентности – набор тестовых данных с общими свойствами. Программа ведет себя одинаково, когда обрабатывает разные элементы класса. Если один из тестов выявит ошибку, остальные, скорее всего, тоже это сделают и наоборот.

1. Анализ граничных значений

[TestMethod]

public void TestMethod1\_eng\_slov()

{

{

// исходные данные

string S = "Testing this test";

int expected = 3;

// получение значения с помощью тестируемого метода

programm1.Form1 test = new programm1.Form1();

int actual = test.FindWords(ref S);

// сравнение ожидаемого результата с полученным

Assert.AreEqual(expected, actual);

}

}

[TestMethod]

public void TestMethod2\_eng\_slov\_znaki()

{

{

// исходные данные

string S = "I said, test done";

int expected = 1;

// получение значения с помощью тестируемого метода

programm1.Form1 test = new programm1.Form1();

int actual = test.FindWords(ref S);

// сравнение ожидаемого результата с полученным

Assert.AreEqual(expected, actual);

}

}

[TestMethod]

public void TestMethod3\_nicego()

{

{

// исходные данные

string S = "";

int expected = 0;

// получение значения с помощью тестируемого метода

programm1.Form1 test = new programm1.Form1();

int actual = test.FindWords(ref S);

// сравнение ожидаемого результата с полученным

Assert.AreEqual(expected, actual);

}

}

[TestMethod]

public void TestMethod4\_rus\_slov()

{

{

// исходные данные

string S = "Тест выполнен";

int expected = 0;

// получение значения с помощью тестируемого метода

programm1.Form1 test = new programm1.Form1();

int actual = test.FindWords(ref S);

// сравнение ожидаемого результата с полученным

Assert.AreEqual(expected, actual);

}

}

[TestMethod]

public void TestMethod5\_rus\_and\_eng()

{

{

// исходные данные

string S = "тест tetris tetris";

int expected = 2;

// получение значения с помощью тестируемого метода

programm1.Form1 test = new programm1.Form1();

int actual = test.FindWords(ref S);

// сравнение ожидаемого результата с полученным

Assert.AreEqual(expected, actual);

}

}

[TestMethod]

public void TestMethod6\_znaki()

{

{

// исходные данные

string S = "!@#$%^ test ^&\*()";

int expected = 1;

// получение значения с помощью тестируемого метода

programm1.Form1 test = new programm1.Form1();

int actual = test.FindWords(ref S);

// сравнение ожидаемого результата с полученным

Assert.AreEqual(expected, actual);

}

}

1. Заполненная таблица для тестовых сценариев.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер**  **теста** | **Описание** | **Тестовые варианты** | **Ожидаемые результаты** | **Полученные результаты** |
| 1 | Англ. Слова с большой и маленькой буквы | Testing this test | 3 | 3 |
| 2 | Слово с буквы t после знака препинания | I said, test done | 1 | 0 |
| 3 | Ничего не вводится |  | 0 | 0 |
| 4 | Русские слова c T | Тест выполнен | 0 | 0 |
| 5 | Русские и английские слова с T | тест tetris tetris | 2 | 2 |
| 6 | Слово среди знаков препинания | !@#$%^ test ^&\*() | 1 | 1 |

Вывод: Тест номер 2 (Слово с буквы t после знака препинания) не прошел проверку.