Задание.

Ввести строку символов (не более 30). Подсчитать количество слов в строке с нечетным количеством символов. Если количество этих слов более трех, то подсчитать количество гласных, если нет, то согласных. Результаты вывести на экран.

Создаем Модульные тесты:

Module1 – Проверка количества символов в строке , подсчет слов

Module2 – Проверка количества гласных и согласных, и проверка на гласную и согласную.

**План Тестирования**

|  |  |
| --- | --- |
| Вход | Выход |
| Module1.countNotVowels("aa бб дд ее"), | 4 |
| (Module1.checkStrLength("при при при при при при при при при") | false |
| Module2.countVowels("здклкртттт") | 0 |

В результате должны получиться следующие методы:

using System;

using System.Windows.Forms;

namespace lab2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

String s = textBox1.Text;

if (!Module1.checkStrLength(s))

{

textBox2.Text = "error";

return;

}

if (Module1.isOddWordsMore(s))

{

textBox2.Text = "" + Module2.countVowels(s) + " Количество гласных";

textBox3.Text = "Количество слов с нечетным количествов символов больше 3";

}

else {

textBox2.Text = "" + Module2.countNotVowels(s) + " Количество согласных";

textBox3.Text = "Количество слов с нечетным количествов символов меньше 3";

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

}

public class Module1 // Модуль один на проверку количества символов в строке, строки должны быть меньше 30,

{

public static Boolean checkStrLength(String s)

{

if (s.Length < 30)

return true;

else

return false;

}

public static Boolean isOddWordsMore(String s) // Подсчет слов , у которых нечетное количество сиволов и если их больше 3 то тру

{

Boolean result=false;

Char delimiter = ' ';

String[] substrings = s.Split(delimiter);// делим строку по пробелам

int count=0;

foreach(var str in substrings) // проходим по новому массиву строк

{

if (str.Length % 2 != 0) // если количество символов слова нечетное количество, то увеличиваем счетчик

count++;

}

if (count > 3)

result = true;

return result;

}

}

public class Module2

{

public static int countVowels(String s)

{

int result = 0;

s=s.ToLower();

for(int i=0; i<s.Length; i++)

{

if (isVowel(s[i]))

result++;

}

return result;

}

public static Boolean isVowel(Char c)

{

Boolean result = false;

char[] array = new char[] {'а', 'е', 'ё', 'и', 'й', 'о', 'у', 'ы', 'э', 'ю', 'я', };

foreach(var ch in array)

{

if (ch == c)

return true;

}

return result;

}

public static int countNotVowels(String s)

{

int result = 0;

s = s.ToLower();

for (int i = 0; i < s.Length; i++)

{

if (isNotVowel(s[i]))

result++;

}

return result;

}

public static Boolean isNotVowel(Char c)

{

Boolean result = false;

char[] array = new char[] { 'б', 'в', 'г', 'д', 'й', 'ж', 'з', 'к',

'л', 'м', 'н', 'п', 'р', 'с', 'т', 'ф', 'х', 'ц', 'ч', 'ш', 'щ'};

foreach (var ch in array)

{

if (ch == c)

return true;

}

return result;

}

}

}

Код Тестого модуля

public class Module1Test

{

[TestMethod]

public void CheckLenght\_Test1()

{

Assert.AreEqual(Module1.checkStrLength("при при при при при при при при при"), false); // больше 30

Assert.AreEqual(Module1.checkStrLength("you are my really good friend you know it"), false); // больше 30

Assert.AreEqual(Module1.checkStrLength("ааа ааа ааа"), true); // меньше 30

Assert.AreEqual(Module1.checkStrLength("Hello world"), true); // меньше 30

}

[TestMethod]

public void CheckCountOfOddWords\_Test2()

{

Assert.AreEqual(Module1.isOddWordsMore("я вижу теб я когда т ы"), true); // больше 3

Assert.AreEqual(Module1.isOddWordsMore("Даа дааа дааа аа"), false); // меньше 3

Assert.AreEqual(Module1.isOddWordsMore("чем чем чем чем"), true); // больше 3

Assert.AreEqual(Module1.isOddWordsMore("меньше чем ты только я"), false);// меньше 3

}

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using lab2;

namespace UnitTestProject1

{

[TestClass]

public class Module2Test

{

[TestMethod]

public void CheckVowels\_Test1()

{

Assert.AreEqual(Module2.countVowels("ееббддее"), 4);

Assert.AreEqual(Module2.countVowels("здклкртттт"), 0);

Assert.AreEqual(Module2.countVowels(""), 0);

}

[TestMethod]

public void CheckNotVowels\_Test1()

{

Assert.AreEqual(Module2.countNotVowels("aa бб дд ее"), 4);

Assert.AreEqual(Module2.countNotVowels("ее ее ее еее"), 0);

Assert.AreEqual(Module2.countNotVowels(""), 0);

}

}

}