**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

**ЛАБОРАТОРНАЯ РАБОТА №4**

по дисциплине: «Объектно-ориентированное программирование»

на тему: «Обработка строк данных. Регулярные выражения»

Выполнил: студент гр. ИТП-21

Король В.Н.

Принял: преподаватель-стажер

Гуменников Е.Д.

Гомель 2022

**Цель работы:** изучить такой тип данных как строки на языке программирования *C*#. Изучить основные методы работы с ними, регулярные выражения для изменения и проверки строк и такой класс как StringBuilder для оптимизации работы со строками в *C*#.

**Задание**

1. Разработать приложения, решающие поставленную задачу (таблица 4.1 и таблица 4.2).

2. Разработать не менее 15 модульных тестов для тестирования.

**Вариант 14**

**1 Задание**

С клавиатуры вводится предложение, слова в котором разделены символом ‘\_’. Напечатать все предложения, которые получаются путем перестановки слов исходного текста.

**2 Задание**

Задан текст. Сформировать массив из предложений (строк класса StringBuilder), в которых встречается заданное слово.

**Ход Работы**

В ходе выполнения первого задания пользователь вводит строку. При помощи регулярных выражения из предложения вырезаются все знаки пунктуации. Далее при помощи метода *Split* строка разбивается на массив слов. Для того чтобы получить число всех возможных комбинаций из этих слов необходимо найти факториал количество слов. Для этого был создан метод *GetFactor* который принимает число и возвращает его факториал. Далее мы задаем цикл на факториал длинны комбинаций и генерируем случайный набор слов. Если такой комбинации нет в массиве предложений то добавляем эту комбинацию если такая комбинация есть то генерируем новую до тех пор пока длинна массива не окажется равной факториалы слов в исходном предложении. Пример работы первого задания указан на рисунке 1.

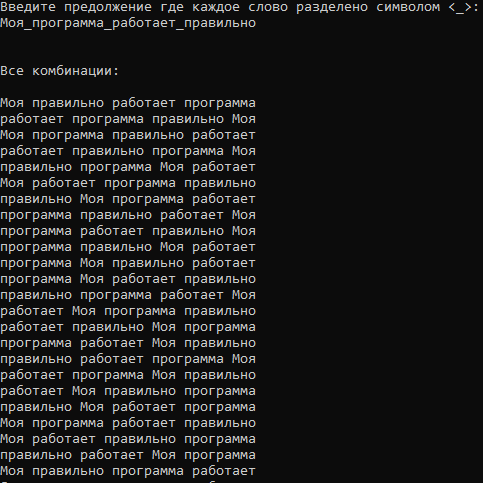


Рисунок 1 – Пример работы первого задания

После написания программ были написаны модульные тесты которые проверяют работу функции *GetFactor, CheckInArr* и *MainFunction*. Пример работы тестов указан на рисунке 2.

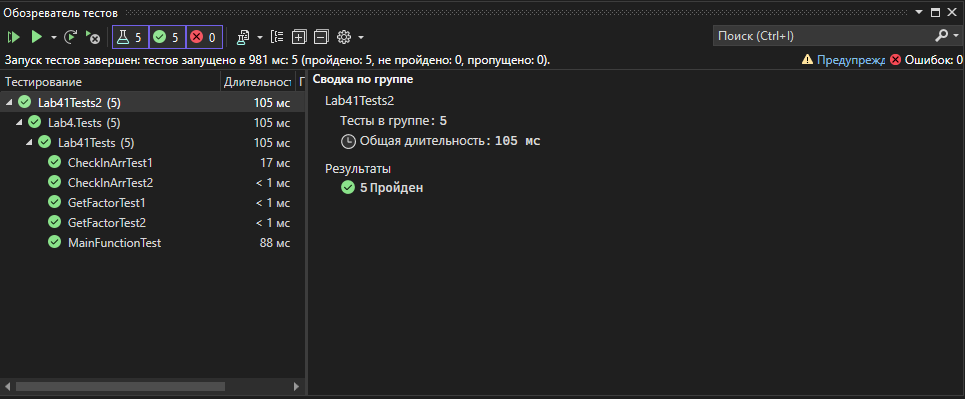


Рисунок 2 - Пример работы тестов к первому заданию

Во вором задании пользователем вводится текст. Текст разбивается на предложения по знак пунктуации. Далее из строки удаляются все лишние пробелы. Вводится ключевое слово и при помощи метода *Contains* определяется есть ли такое слово в предложении. Если оно есть то предложение добавляется в переменную класса *StringBuilder* при помощи метода *Append* после проверки всех предложений на экран выводятся те в которых есть ключевое слово. Пример работы второго задания указа на рисунке 2.

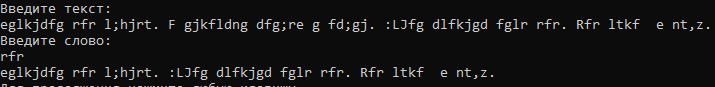


Рисунок 3 – Пример работы второго задания

**Вывод:** в ходе выполнения лабораторной работы был изучен такой тип данных как строки на языке программирования *C*#. Основные методы работы с ними. Также были выполнения задания в соответствии со своим вариантом.

**Листинг класса *Lab41***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

namespace Lab4

{

public class Lab41

{

public string UserString;

public Lab41(string userString)

{

UserString = userString;

}

public int GetFactor(int num)

{

int resault = 1;

for (int i = 1; i <= num; i++)

{

resault \*= i;

}

return resault;

}

public bool CheckInArr(string[] strArr, string str)

{

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

{

if (strArr[i] == str)

{

return false;

}

}

return true;

}

public string[] MainFunction()

{

Random random = new Random();

UserString = Regex.Replace(UserString, "[-.?!)(,:]", "");

string[] arrString = UserString.Split('\_');

string[] resaultString = new string[GetFactor(arrString.Length)];

string[] indexString = new string[GetFactor(arrString.Length)];

int kol = 0;

while (kol < GetFactor(arrString.Length))

{

string tmp = "";

string indexs = "";

int i = 0;

while (i < arrString.Length)

{

int word = random.Next(0, arrString.Length);

if (CheckInArr(indexs.Split(' '), Convert.ToString(word)))

{

indexs += Convert.ToString(word) + " ";

i++;

}

}

for (int j = 0; j < arrString.Length; j++)

{

tmp += arrString[Convert.ToInt32(indexs.Split(' ')[j])] + " ";

}

if (CheckInArr(indexString, indexs))

{

indexString[kol] = indexs;

resaultString[kol] = tmp;

kol++;

}

}

return resaultString;

}

public static bool operator ==(Lab41 str1, string[] str2)

{

string[] str = str1.MainFunction();

if (str.Length == str2.Length)

{

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

{

if (!str1.CheckInArr(str, str2[i]))

{

return false;

}

}

return true;

}

return false;

}

public static bool operator !=(Lab41 str1, string[] str2)

{

string[] str = str1.MainFunction();

if (str.Length == str2.Length)

{

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

{

if (!str1.CheckInArr(str, str2[i]))

{

return true;

}

}

return false;

}

return true;

}

}

}

**Листинг тестов к первому заданию**

using Microsoft.VisualStudio.TestTools.UnitTesting;

using Lab4;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab4.Tests

{

[TestClass()]

public class Lab41Tests

{

[TestMethod()]

public void GetFactorTest1()

{

Lab41 classTest = new Lab41("qwe");

int except = 120;

int act = classTest.GetFactor(5);

Assert.IsTrue(act == except);

}

[TestMethod()]

public void GetFactorTest2()

{

Lab41 classTest = new Lab41("qwe");

int except = 24;

int act = classTest.GetFactor(5);

Assert.IsTrue(act != except);

}

[TestMethod()]

public void CheckInArrTest1()

{

Lab41 classTest = new Lab41("qwe");

bool except = false;

bool act = classTest.CheckInArr(new string[] {"q", "e", "w"}, "e");

Assert.IsTrue(act == except);

}

[TestMethod()]

public void CheckInArrTest2()

{

Lab41 classTest = new Lab41("qwe");

bool except = true;

bool act = classTest.CheckInArr(new string[] { "q", "h", "w" }, "e");

Assert.IsTrue(act == except);

}

[TestMethod()]

public void MainFunctionTest()

{

Lab41 classTest = new Lab41("q\_w");

string[] expect = { "q w", "w q" };

Assert.IsTrue(classTest == expect);

}

}

}

**Листинг 2 задания**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.CompilerServices;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

namespace Lab42

{

internal class Program

{

static void Main(string[] args)

{

StringBuilder sb = new StringBuilder();

Console.WriteLine("Введите текст: ");

string userString = Regex.Replace(Console.ReadLine(), "[.?!]", ".");

userString = Regex.Replace(userString, @"\+", " ");

string[] strings = userString.Split('.');

Console.WriteLine("Введите слово: ");

string word = Console.ReadLine();

foreach (var s in strings)

{

string[] tmp = s.Split(' ');

bool wordExis = false;

int i = 0;

while(i < tmp.Length && !wordExis)

{

if ((tmp[i].ToLower()).Contains(word) && word.Length == tmp[i].Length)

{

wordExis = true;

}

i++;

}

if (wordExis)

{

sb.Append(s + ".");

}

}

Console.WriteLine(sb);

}

}

}