Лабораторная работа №7

Тема: Разработка программ с использованием символов и строк.

Цель работы: Формирование умений и навыков в разработке программ с использованием строк динамической длины.

Время на выполнение работы: 4 часа

Этапы работы:

I. Ознакомиться с теоретическими сведениями.
II. Выполнить задания, предложенные преподавателем.
III. Выполнить общее задание для всех на отметку
IV. Ответить на контрольные вопросы.

I. Краткие теоретические сведения

String

Строки в C# - это экземпляры класса System.String. В C# есть тип string, но класс System.String является более мощным, так что его использование часто оказывается более оправданным и простым. Этот класс имеет множество методов и свойств, некоторые из которых перечислены ниже:

1.1.1 Свойство Length

Возвращает длину строки. Пример использования:

```
String s="qqq";
int k=s.Length; //В k запишется 3
```

1.1.2 Свойство Compare

Статический метод, сравнивающий две строки. Возвращает 0, есть строки равны, отрицательное значение, если первая строка меньше второй и положительное значение, если первая строки больше второй (больше и меньше в алфавитном смысле, разумеется). Пример использования:

```
namespace test
{
    class Test
    {
        public static void Main()
        {
            String s1="arbour", s2="ace", s3="azote";
```

```
            System.Console.WriteLine(String.Compare(s1,  s1));
//Выдаст 0, т. к. "arbour" равно "arbour".
            System.Console.WriteLine(String.Compare(s1,  s2));
//Выдаст -1, т. к. "arbour" меньше "ace".
            System.Console.WriteLine(String.Compare(s1,  s3));
//Выдаст 1, т. к. "arbour" больше "azote".
        }
    }
}
```

1.1.3 Метод Equals

Метод, возвращает true, если строки равны, false - если не равны. Пример использования:

```
String s1="qqq", s2="www";
System.Console.WriteLine(String.Equals(s1, s2).ToString());
```

1.1.4 Метод Substring

Поволяет извлечь из строки подстроку. Пример использования:

```
String s1="abcdefg", s2;
s2=s1.Substring(3, 2);
System.Console.WriteLine(s2); //Напечатается "de"
```

Где: первый параметр- означает, с какого места извлекаем (нумерация с нуля), а второй - сколько символов извлекаем.

1.1.5 Метод Insert

Вставляет в строку другую строку. Пример использования:

```
String s1="abcdefg", s2;
s2=s1.Insert(1, "xyz");
System.Console.WriteLine(s2); //Напечатается "axyzbcdefg"
```

Где, первый параметр означает, куда вставляем подстроку(нумерация, как всегда, с нуля), второй - что за подстроку вставляем.

1.1.6 Метод IndexOf

Позволяет найти в строке подстроку. Пример использования:

```
String s1="abcabcab", s2="bc", s3="zzz";
System.Console.WriteLine(s1.IndexOf(s2)); //Напечатается 1
System.Console.WriteLine(s1.IndexOf(s3)); //Напечатается -1
```

Этот метод возвращает номер позиции, на котором в строке находится передаваемая в качестве параметра подстрока. Если такой подстроки нет, то возвращается -1.

1.1.7 Метод Replace

Производит замену в строке. Пример использования:

```
String s1="abcabcab", s2="bc", s3;
s3=s1.Replace(s2, "q");
System.Console.WriteLine(s3); //Напечатается aqaqab
```

1.1.8 Методы EndWith и StartsWith

Проверяют, не завершается ли или не начинается ли строка с или на заданную строку соответственно. Пример использования:

```
String s1="arbour";
if(s1.StartsWith("ar"))
   System.Console.WriteLine("Строка начинается на \"ar\"");
else
   System.Console.WriteLine("Строка не начинается на \"ar\"");
```

1.1.9 Методы ToUpper и ToLower

Переводят строку в верхний или нижний регистр соответственно. Пример использования:

```
String s1="aRbRur";
s1=s1.ToLower();
```

1.1.10 Методы Trim, TrimEnds и TrimStart

Удаляют пробельные символы из начала и конца строки (Trim), только с конца строки (TrimEnds) и только с начала строки (TrimStart). Пример использваония:

```
String s1="  ar  brur ";
System.Console.Write(s1.Trim());
```

При изменении строки фактически старый экземпляр класса System.String уничтожается, и создается новый с тем же именем и измененным содержанием. Это означает, что при интенсивной работе со строками программа может работать не так быстро, как хотелось бы. Если мы не хотим, чтобы каждый раз создавался новый экземпляр класса, то вместо класса System.String надо использовать класс StringBuilder.

1.1.10. Методы Join и Split

Методы Join и Split выполняют над строкой текста взаимно обратные преобразования. Динамический метод Split позволяет осуществить разбор текста на элементы. Статический метод Join выполняет обратную операцию, собирая строку из элементов.

Заданный строкой текст зачастую представляет собой совокупность структурированных элементов - абзацев, предложений, слов, скобочных выражений и

т.д. При работе с таким текстом необходимо разделить его на элементы, пользуясь специальными разделителями элементов, - это могут быть пробелы, скобки, знаки препинания. Практически подобные задачи возникают постоянно при работе со структурированными текстами. Методы Split и Join облегчают решение этих задач.

Динамический метод Split, как обычно, перегружен. Наиболее часто используемая реализация имеет следующий синтаксис:

```
public string[] Split(params char[])
```

На вход методу Split передается один или несколько символов, интерпретируемых как разделители. Объект string, вызвавший метод, разделяется на подстроки, ограниченные этими разделителями. Из этих подстрок создается массив, возвращаемый в качестве результата метода. Другая реализация позволяет ограничить число элементов возвращаемого массива.

Синтаксис статического метода Join:

```
public static string Join(string delimiters, string[] items )
```

В качестве результата метод возвращает строку, полученную конкатенацией элементов массива items, между которыми вставляется строка разделителей delimiters. Как правило, строка delimiters состоит из одного символа, который и разделяет в результирующей строке элементы массива items; но в отдельных случаях ограничителем может быть строка из нескольких символов.

StringBuilder

Класс StringBuilder принадлежит пространству имен System.Text.

Этот класс работает быстрее, чем класс String, так как при изменении строки, созданной как экземпляр класса String, создается каждый раз новый экземпляр класса, а старый уничтожается, при использовании же класса StringBuilder расходов на создание-уничтожение экземпляра класса нет - работа всегда происходит с одним экземпляром.

Для этого класса нельзя использовать простое присваивание:

```
StringBuilder s="abc"; //Неправильно!
```

В этом случае надо выполнять следующие действия:

```
StringBuilder s=new StringBuilder("abc"); //Правильно
```

У класса StringBuilder нет статических методов. Все его методы - динамические. Ниже перечислены основные свойства и методы класса StringBuilder:

1.2.1 Свойство Length

Возвращает длину строки. Пример использования:

```
int k=s.Length;
```

1.2.2 Свойство MaxCapacity

Свойство только для чтения. Дает максимальное количество символов, которые можно записать в объект типа StringBuilder. Пример использования:

```
System.Console.WriteLine(s.MaxCapacity);
```

1.2.3 Метод Append

Прибавляет строку к существующей. Пример использования:

```
StringBuilder s1=new StringBuilder("Cogito ");
 StringBuilder s2=new StringBuilder("ergo ");
      s1.Append(s2);
      s1.Append("sum");
      System.Console.WriteLine(s1);  //Напечатается  "Cogito ergo sum"
```

1.2.4 Метод Equals

Служит для сравнения двух строк. Возвращает true или false. Пример использования:

```
if(s1.Equals(s2))
    System.Console.WriteLine("Строки равны");
else
    System.Console.WriteLine("Строки не равны");
```

1.2.5 Метод Insert

Вставляет символы в заданную позицию (Нумерация идет с нуля). Пример использования:

```
StringBuilder s1=new StringBuilder("abcde");
s1.Insert(2, "xyz");
System.Console.WriteLine(s1); //Напечатается "abxyzcde"
```

1.2.6 Метод Remove

Удаляет символы из строки. Пример использования:

```
StringBuilder s1=new StringBuilder("abcde");
s1.Remove(1, 2);
System.Console.WriteLine(s1); //Напечатается "ade"
```

Певый параметр - это с какой позиции удаляем (нумерация с нуля), второй - сколько символов удаляем.

1.2.7. Метод Replace

Заменяет символы. Пример использования:

```
StringBuilder s=new StringBuilder("abcdeabcde");
s.Replace("abc", "ZZZ");
System.Console.WriteLine(s); //Напечатается "ZZZdeZZZde"
```

Пример использования String VS StringBuilder

```
class Program
    {
        static void Main(string[] args)
        {
            PrintArray(StringSplitToArray("1 2 3 4 5 6 7 8 9"));
            PrintArray(StringSplitToArrayWithConverterALL("1 2 3 4 5 6 7 8 9"));
            StringVsStrinBuilder();
            AddString();
            AddStringBuilder();
            RefStringVsStringbuilder();
            TimeCreateStringVsStringBuilder();
            OperationWithStringBuilder();
            ArrayForSBAndNoStringBuilder(100000);
            RunEqulsStrinBuilder();
        }
        private static double[] StringSplitToArrayWithConverterALL(string line)
        {
            return Array.ConvertAll(line.Split(), new Converter<string,
double>(StringToDouble));
        }

        private static double[] StringSplitToArray(string line)
        {
            string[] strArray = line.Split();
            double[] array = new double[strArray.Length];
            for (int i = 0; i < array.Length; i++)
            {
                array[i] = double.Parse(strArray[i]);
            }
            Console.WriteLine(array.GetType());
            return array;

        }

        private static double StringToDouble(string input)
        {
            return double.Parse(input);
        }

        static void PrintArray(double[] array)
        {
            foreach (var item in array)
            {
                Console.Write(item + " ");
            }
            Console.WriteLine();
        }

        private static void RunEqulsStrinBuilder()
        {
            StringBuilder sb = new StringBuilder("Hello");
            StringBuilder sb2 = new StringBuilder("Hello");
            if (EqualsStringBuilder(sb, sb2))
                Console.WriteLine("sb = sb2");
            else
                Console.WriteLine("sb != sb2");
        }

        private static void ArrayForSBAndNoStringBuilder(int count)
        {
            ExampleStringBuilder(count);
            Console.WriteLine("----------");
            ExampleNoStringBuilder(count);
        }

        private static bool EqualsStringBuilder(StringBuilder sb, StringBuilder sb2)
        {
```

```
    return sb == sb2 ? true : false;
    //return sb.Equals(sb2) ? true : false;
    //return sb.GetHashCode() == sb2.GetHashCode() ? true : false;
}

private static void ExampleNoStringBuilder(int count)
{
    DateTime start = DateTime.Now;
    int[] array = new int[count];
    Random random = new Random();

    string s = ", ";
    Console.WriteLine("{" + string.Join(", ", array) + "}");
    //s+="{";
    //for (int i = 0; i < array.Length - 1; i++)
    //{
    //    array[i] = random.Next(1, 10);
    //    s+=(array[i] + ", ");
    //}
    //s += array[array.Length - 1];
    //s+="}";

    DateTime end = DateTime.Now;
    Console.WriteLine(end - start);


}

private static void ExampleStringBuilder(int count)
{
    DateTime start = DateTime.Now;
    int[] array = new int[count];
    Random random = new Random();
    StringBuilder stringBuilder = new StringBuilder("{");

    for (int i = 0; i < array.Length; i++)
    {
        array[i] = random.Next(1, 10);
        if (i != array.Length - 1)
            stringBuilder.Append(array[i]).Append(", ");
        else
            stringBuilder.Append(array[i]);
    }
    stringBuilder.Append("}");
    Console.WriteLine(stringBuilder);
    DateTime end = DateTime.Now;
    Console.WriteLine(end - start);

}

private static void OperationWithStringBuilder()
{
    StringBuilder stringBuilder = new StringBuilder("Привет, StrigBuilder!");
    StringBuilder stringBuilder2 = new StringBuilder(30);
    //GetLengthAndCapacity(stringBuilder, stringBuilder2);
    InsertRemoveReplaceAppendFormat(stringBuilder);
    Console.WriteLine(stringBuilder);
}

private static void InsertRemoveReplaceAppendFormat(StringBuilder stringBuilder)
{
    stringBuilder.Append("!!!");
    stringBuilder.Insert(7, "класс"); // вставка на 7 индекс слово класс

    //заменяем слово
    stringBuilder.Replace("Привет", "Приветствую тебя");
```

```
            //удаляем подстроку
            stringBuilder.Remove(17, 5);

            int i = 10;
            stringBuilder.AppendFormat($"переменная i = {i}");
            stringBuilder.Append($"переменная i = {i}");
        }

        private static void GetLengthAndCapacity(StringBuilder stringBuilder, StringBuilder
stringBuilder2)
        {
            Console.WriteLine($"Длина stringBuilder: {stringBuilder.Length}");
            Console.WriteLine($"Емкость stringBuilder: {stringBuilder.Capacity}");
            Console.WriteLine("----------------------");
            stringBuilder2.Append("aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa");
            Console.WriteLine($"Длина stringBuilder2: {stringBuilder2.Length}");
            Console.WriteLine($"Емкость stringBuilder2: {stringBuilder2.Capacity}");
        }

        private static void TimeCreateStringVsStringBuilder()
        {
            Console.WriteLine($"Время создания StringBuiled
{TimeCreateBigStringBuilder(100000)}");
            Console.WriteLine($"Время создания String {TimeCreateBigString(100000)}");

        }

        private static void RefStringVsStringbuilder()
        {
            StringBuilder sb = new StringBuilder();
            string s = "first";
            WorkStringBuilder(sb);
            WorkString(s);
            Console.WriteLine(sb);
            Console.WriteLine(s);
        }

        private static void WorkString(string s)
        {
            s += "HelloString";
            s += 45687.78;
            s += "sdfaf";
            Console.WriteLine(s);
        }

        private static void WorkStringBuilder(StringBuilder sb)
        {
            sb.Append("Hello StringBuilder").Append(54864.23);
        }

        private static void AddString()
        {
            string s = "";
            s += false;
            s += 's';
            s += 458.45;
            Console.WriteLine(s);
        }

        private static void AddStringBuilder()
        {
            StringBuilder stringBuilder = new StringBuilder();


stringBuilder.Append(false).Append("Hello").Append('D').Append(46848.89f).AppendLine();
            Console.WriteLine(stringBuilder);
        }
```

```
        private static TimeSpan TimeCreateBigString(int len)
        {
            DateTime start = DateTime.Now;
            string s = "";
            for (int i = 0; i < len; i++)
            {
                s += i;
            }
            DateTime end = DateTime.Now;
            return end - start;
        }

        private static TimeSpan TimeCreateBigStringBuilder(int len)
        {
            DateTime start = DateTime.Now;
            StringBuilder stringBuilder = new StringBuilder();
            for (int i = 0; i < len; i++)
            {
                stringBuilder.Append(i);
            }
            DateTime end = DateTime.Now;
            return end - start;
        }

        private static void StringVsStrinBuilder()
        {
            string s1 = "Hello";
            string s2 = "Hello";
            s1.Replace("He", "dsafadasdf");
            Console.WriteLine(s1);
            StringBuilder sb = new StringBuilder(s1);
            StringBuilder sb2 = new StringBuilder(s1);

        }
    }
```

Практическая часть

II. Индивидуальные задания

Реализовать программу согласно своему варианту, описать функции выполнения своих заданий.
Дана строка – это значит, что пользователь сам ее вводит, а программа ее обрабатывает
Можно использовать string и StringBuilder, на свое усмотрение.
Можно использовать различные вспомогательные функции и методы. Также вы сами можете их создавать и вызывать.

Вариант	Индивидуальное задание
1	Дана строка. Показать третий, шестой, девятый и так далее символы. Строка состоит из слов, разделенных одним или несколькими пробелами. Найдите слово наибольшей длины. Вывести три слова победителя (самое длинное, чуть покороче и еще покороче).
2	Сформировать строку из 10 символов. На четных позициях должны находится четные цифры, на нечетных позициях - буквы. Проверить, соблюдается ли в заданном тексте баланс открывающих и закрывающих круглых скобок, то есть можно ли установить взаимно

	однозначное соответствие открывающих и закрывающих скобок, причем открывающая скобка всегда предшествует соответствующей закрывающей.
3	В данной строке найти количество цифр. Дана строка, которое содержит квадратное уравнение, состоящая из цифр, символа умножения '*', символа 'x' неизвестной, символов '+', '-', '=', '^' . Найдите корни данного уравнения.
4	Замените в строке все вхождения 'word' на 'letter'. В строке записано целое число. Запишите данное число римскими цифрами.
5	Дана строка. Найти сумму имеющихся в нем цифр. Написать алгоритм генерации пароля, (пользователь вводит количество символов минимум 8), а программа выдает генерируемый пароль, в пароле обязательно, как минимум по одной букве, верхнего регистра, цифре, спецсимволу, пароль вводится на английском.
6	Дана строка. Найти сумму имеющихся в нем чисел. Дано натуральное число. Получить строку, в которой тройки цифр этого числа разделены пробелом, начиная с правого конца. Например, число 1234567 преобразуется в 1 234 567.
7	Дан массив строк. Упорядочить массив по длине строк от меньшего к большему. Написать генерацию строк длины 10, причем первые 4 символа - цифры, следующие два символы - различные буквы, следующие 4 символа - нули или единицы, причем одна единица точно присутствует.
8	Дана строка. Удалите k-ый символ в ней. Дана строка, зашифровать ее методом нумерации, т.е. в сообщении, состоящем из одних русских букв и пробелов, каждую букву заменить ее порядковым номером в русском алфавите (А — 1, Б — 2, ..., Я — 33), а пробел — нулем.
9	Дана строка. Заменить все символы на верхний регистр

	В строке найдите все серии подряд идущих пробелов и замените каждую на один пробел.
10	Дана строка. Заменить все символы на нижний регистр Исключить из строки группы символов, расположенные между символами «/*», «*/» включая границы . Предполагается, что нет вложенных скобок.
11	Дана строка. Заменить все символы верхнего регистра на нижний регистр, а символы нижнего регистра на верхний Дана строка. Если символы в ней упорядочены по алфавиту, то вывести 'yes', иначе вывести первый символ, нарушающий алфавитный порядок.
12	Дана строка. Вставить после каждого символа пробел. Реализуйте метод, осуществляющий сжатие строки на основе счетчика повторяющихся символов. Например, строка aaabbcccc должна превратиться в a3b2c5.
13	Удалить в строке все лишние пробелы, то есть серии подряд идущих пробелов заменить на одиночные пробелы. Крайние пробелы в строке удалить. Удалить в строке все цифры.
14	Строка состоит из слов, разделенных одним или несколькими пробелами. Поменяйте местами наибольшее по длине слово и наименьшее. Дана строка, имеющая вид x1−x2+x3−x4+..., где xi - цифра или двузначное число. Найдите значение выражения.
15	Строка состоит из слов, разделенных одним или несколькими пробелами. Переставьте слова по убыванию их длин. Даны две строки. Определите, содержится ли меньшая по длине строка в большей.
16	Строка состоит из слов, разделенных одним или несколькими пробелами. Переставьте слова в алфавитном порядке. Дана строка, имеющая вид x1+x2+...+xn, где xi - цифра или двузначное

	число. Найдите значение выражения.
17	Дана строка, состоящая из слов, разделенных символами, которые перечислены во второй строке. Показать все слова. Дана строка, в которой нет начальных и конечных пробелов. Необходимо изменить ее так, чтобы длина строки стала равна заданной длине, больше чем текущая длина строки. Это следует сделать путем вставки между словами дополнительных пробелов. Количество пробелов между отдельными словами не должно отличаться более чем на один пробел (то есть пробелы добавляются равномерно).
18	Даны две строки. Вывести большую по длине строку столько раз, на сколько символов отличаются строки. Дана строка. Разделить строку на фрагменты по три подряд идущих символа. В каждом фрагменте средний символ заменить на случайный символ, не совпадающий ни с одним из символов этого фрагмента. Показать фрагменты, упорядоченные по алфавиту.

IIIОбщее задание на отметку в журнал

Создать класс Poem, в нем описать статическое поле

```
public static string text = "У лукоморья дуб зелёный;\n" +
                            "Златая цепь на дубе том:\n" +
                            "И днём и ночью кот учёный\n" +
                            "Всё ходит по цепи кругом;\n" +
                            "Идёт направо – песнь заводит,\n" +
                            "Налево – сказку говорит.\n" +
                            "Там чудеса: там леший бродит,\n" +
                            "Русалка на ветвях сидит;\n" +
                            "Там на неведомых дорожках\n" +
                            "Следы невиданных зверей;\n" +
                            "Избушка там на курьих ножках\n" +
                            "Стоит без окон, без дверей;\n" +
                            "Там лес и дол видений полны;\n" +
                            "Там о заре прихлынут волны\n" +
                            "На брег песчаный и пустой,\n" +
                            "И тридцать витязей прекрасных\n" +
                            "Чредой из вод выходят ясных,\n" +
                            "И с ними дядька их морской;\n" +
                            "Там королевич мимоходом\n" +
                            "Пленяет грозного царя;\n" +
                            "Там в облаках перед народом\n" +
                            "Через леса, через моря\n" +
                            "Колдун несёт богатыря;\n" +
                            "В темнице там царевна тужит,\n" +
                            "А бурый волк ей верно служит;\n" +
                            "Там ступа с Бабою Ягой\n" +
```

```
"Идёт, бредёт сама собой,\n" +
"Там царь Кащей над златом чахнет;\n" +
"Там русский дух… там Русью пахнет!\n" +
"И там я был, и мёд я пил;\n" +
"У моря видел дуб зелёный;\n" +
"Под ним сидел, и кот учёный\n" +
"Свои мне сказки говорил.";
```

1. В каждом слове текста 4-ю и 7-ю буквы заменить символами #. Для слов короче 4 символов корректировку не выполнять. Для слов короче 7 символов заменять только 4-ю букву.
2. Определить, сколько раз повторяется в тексте каждое слово, которое встречается в нем, использую массивы. Вывести результаты на консоль в формате **слово = количество повторов**.
3. Вывести в консоль все слова текста, начинающихся согласной и заканчивающихся гласной буквой (нужно сделать для проверки этого условия приватный метод без regex (регулярного выражения))
4. Сформировать и вернуть в методе ***string Slow(string text)*** из случайных слов исходного текста строку ***string*** минимум в сто тысяч символов путем конкатенации. Слова должны быть склеены через один пробел. Затем в ***Main*** определить время работы метода ***Slow*** и вывести его(т.е. время) на консоль. Ускорить процесс, используя ***StringBuilder***(сделать новый метод ***string Fast(string text)*** Итог строки должен быть точно такой же как и в методе **Slow**, до буквы) в ***Main*** повторить вывод. Сравнить время работы двух этих методов.

IV. Контрольные вопросы

1. Пояснить особенность класса string.
2. Как можно создать строки динамической длины?
3. Привести примеры методов работы со строками.

Литература

1. Полный справочник по C#. Г. Шилдт. Издательский дом «Вильямс», 2004.
2. C# в подлиннике. Наиболее полное руководство. Х.Дейтел.
3. C# в задачах и примерах. Культин. Н.Б.
4. C# учебный курс. Г.Шилдт. СПб.: Питер, 2002.
5. C# программирование на языке высокого уровня Павловская Т.А. СПб.: БХВ-Петербург.