Лабораторная работа №7

Тема: Разработка программ с использованием символов и строк.

Цель: Научиться разрабатывать алгоритмы и программы по обработке

строк, использовать возможности классов Char, Char[], String и StringBuilder при работе с символьными и строковыми данными.

Выполнение работы:

Пример 1. Дана символьная строка s, известно, что в этой строке есть хотя бы один символ «точка». Преобразовать строку, удалив из неё запятые, предшествующие первой точке, и заменив знаком * все символы пробела, встречающиеся после первой точки.

```
Возможная реализация:
using System;
namespace Prim_String
class Program
public static void Main(string[] args)
{
string s = "qwerty, t, r . abc f t";
Console.WriteLine("Исходная строка:\n"+s);
int k = s.IndexOf(".");
Console.WriteLine("Первая точка находится в позиции: "+k);
string s1 = s.Substring(0, k);
Console.WriteLine(s1);
int i;
do{
i = s1.IndexOf(',');
if(i >= 0)
s1 = s1.Remove(i, 1);
while(i >= 0);
Console.WriteLine(s1);
string s2 = s.Substring(k);
Console.WriteLine(s2);
s2 = s2.Replace('', '*');
Console.WriteLine(s2);
```

```
s = s1 + s2;
Console.WriteLine("Результирующая строка:\n"+s);
Console.Write("Press any key to continue . . . ");
Console.ReadKey(true);
}
}
}
       Исходная строка:
        qwerty, t, r . abc f t
        Первая точка находится в позиции: 13
        qwerty, t, r
        qwerty t r
        . abc f t
        .*abc*f*t
        Результирующая строка:
        qwerty t r .*abc*f*t
        Press any key to continue . .
```

Рисунок 1 - Результаты работы программы

Пример 2. Дана символьная строка, содержащая хотя бы один символ '?' (вопросительный знак). Удалить все символы '!' (восклицательный знак) или '.' (точка), расположенные до первого символа '?' и заменить все пробелы, расположенные за первым символом '?', на символ подчёркивания ' '

```
Возможная реализация:
using System;
using System.Text;
namespace Prim_StringBuilder
{
class Program
{
public static void Main(string[] args)
{
Console.WriteLine("Исходная строка:");
string s = Console.ReadLine();
int k = s.IndexOf('?');
Console.WriteLine("k={0}", k);
StringBuilder b1 = new
```

```
StringBuilder(s.Substring(0, k));
Console.WriteLine(b1);
StringBuilder b2 = new
StringBuilder(s.Substring(k));
Console.WriteLine(b2);
for(int i = 0; i < b1.Length; i++)
{
if(b1[i] == '!' | | b1[i] == '.')
b1.Remove(i, 1);
i--;
}
}
b2.Replace(' ', '_');
s = b1.ToString() + b2.ToString();
Console.WriteLine("Результат:\n"+s);
Console.Write("Press any key to continue . . . ");
Console.ReadKey(true);
}
}
}
         Исходная строка:
         1a!2.3!!.. 4.!.?6 7! ..?
         k = 15
         1a!2.3!!.. 4.!.
         ?6 7! ..?
         Результат:
         1a23 4?6 7! ..?
         Press any key to continue . . .
```

Рисунок 2 - Результаты работы программы

Пример 3. Программа по использованию методов класса String. Листинг программы: using System; using System.Collections.Generic; using System.Ling;

```
using System.Text;
      namespace ConsoleApplication1
      {
        class Program
        {
          static void Main(string[] args)
          {
            // Сравним первые две строки
            string s1 = "это строка";
            string s2 = "это текст, а это строка";
            if (String.CompareOrdinal(s1, s2) != 0)
              Console.WriteLine("Строки s1 и s2 не равны");
            if (String.Compare(s1, 0, s2, 13, 10, true) == 0)
              Console.WriteLine("При этом в них есть одинаковый текст");
            // Конкатенация строк
            Console.WriteLine(String.Concat("\n" + "Один,
                                                                         ","три,
                                                                  два
четыре"));
            // Поиск в строке
            // Первое вхождение подстроки
             if (s2.IndexOf("это") != -1)
              Console.WriteLine("Слово \"это\" найдено в строке, оно "+
               "находится на: {0} позиции", s2.IndexOf("это"));
             // Последнее вхождение подстроки
             if (s2.LastIndexOf("это") != -1)
               Console.WriteLine("Последнее вхождение
                                                                         \"это\"
                                                                слова
находится "
                 + "на {0} позиции", s2.LastIndexOf("это"));
             // Поиск из массива символов
             char[] myCh = \{'bl', 'x', '\tau'\};
             if (s2.IndexOfAny(myCh) != -1)
```

```
Console.WriteLine("Один из символов из массива ch "+
                       "найден
                                                                                                   {0}",
                                             текущей
                                       В
                                                             строке
                                                                           на
                                                                                   позиции
s2.IndexOfAny(myCh));
                 // Определяем начинается ли строка с заданной подстроки
                 if (s2.StartsWith("это текст") == true)
                    Console.WriteLine("Подстрока найдена!");
                 // Определяем содержится ли в строке подстрока
                 // на примере определения ОС пользователя
                 string myOS = Environment.OSVersion.ToString();
                 if (myOS.Contains("NT 5.1"))
                    Console.WriteLine("Ваша операционная система Windows XP");
                 else if (myOS.Contains("NT 6.1"))
                   Console.WriteLine("Ваша операционная система Windows 7");
                Console.ReadLine();
             }
          }
           🔳 file:///C:/projects/test/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1.... 🖃 🖳
           Строки s1 и s2 не равны
При этом в них есть одинаковый текст
           Один, два три, четыре
Слово "это" найдено в строке, оно находится на: О позиции
Последнее вхождение слова "это" находится на 13 позиции
Один из символов из массива ch найден в текущей строке на позиции 1
Подстрока найдена!
Ваша операционняя система Uindows 7
               операционная система Windows 7
```

Рисунок 2 – Результаты по использованию методов класса String

Пример 4. Программа по использованию методов Compare(), Equals(), CompareOrdinal(), а также операторов == и != для сравнения символьных строк.

```
Листинг программы: using System; class Example {
```

```
static void Main()
  string str1 = "alpha";
  string str2 = "Alpha";
  string str3 = "Beta";
  string str4 = "alpha";
  string str5 = "alpha, beta";
  int result;
 // Сначала продемонстрировать отличия между сравнением строк
 // с учетом культурной среды и порядковым сравнением
  result = String.Compare(str1, str2, StringComparison.CurrentCulture);
  Console.Write("Сравнение строк с учетом культурной среды: ");
  if (result < 0)
    Console.WriteLine(str1 + " меньше " + str2);
  else if(result > 0)
    Console.WriteLine(str1 + " больше " + str2);
  else
    Console.WriteLine(str1 + " равно " + str2);
  result = String.Compare(str1, str2, StringComparison.Ordinal);
  Console.Write("Порядковое сравнение строк: ");
  if (result < 0)
    Console.WriteLine(str1 + " меньше " + str2);
  else if(result > 0)
    Console.WriteLine(str1 + " больше " + str2);
  else
    Console.WriteLine(str1 + " равно " + str4);
  // Использовать метод CompareOrdinal()
  result = String.CompareOrdinal(str1, str2);
  Console.Write("Сравнение строк методом CompareOrdinal():\n");
  if (result < 0)
    Console.WriteLine(str1 + " меньше " + str2);
  else if(result > 0)
    Console.WriteLine(str1 + " больше " + str2);
```

```
Console.WriteLine(str1 + " равно " + str4);
          Console.WriteLine();
          // Определить равенство строк с помощью оператора ==
          // Это порядковое сравнение символьных строк
          if (str1 == str4)
            Console.WriteLine(str1 + " == " + str4);
          // Определить неравенство строк с помощью оператора !=
          if(str1 != str3)
            Console.WriteLine(str1 + " != " + str3);
          if(str1 != str2)
            Console.WriteLine(str1 + " != " + str2);
          Console.WriteLine();
          // Выполнить порядковое сравнение строк без учета регистра,
          // используя метод Equals()
          if(String.Equals(str1, str2, StringComparison.OrdinalIgnoreCase))
            Console.WriteLine("Сравнение строк методом Equals() с " +
              "параметром OrdinalIgnoreCase:\n" +
              str1 + " равно " + str2);
          Console.WriteLine ();
          // Сравнить части строк
          if(String.Compare(str2, 0, str5, 0, 3,
            StringComparison.CurrentCulture) > 0) {
            Console.WriteLine("Сравнение строк с учетом текущей культурной
среды:"+
              "\n3 первых символа строки " + str2 +
              " больше, чем 3 первых символа строки " + str5);
          }
        }
      }
```

else

Рисунок 4 – Результат выполнения программы

```
Пример 5. Использование класса StringBuilder:
     using System;
     using System.Collections.Generic;
     using System.Ling;
     using System.Text;
      namespace ConsoleApplication1
        class Program
          static void Main(string[] args)
            StringBuilder hello = new StringBuilder("Привет, меня зовут
Александр Ерохин",120);
            hello.AppendFormat(" Я рад вас приветствовать на моем сайте
professorweb.ru");
            // Зашифруем строку, хранящуюся в переменной hello
            Console.WriteLine("Исходная строка: \n {0}\n",hello);
            for (int i = 's'; i >= 'a'; i--)
              hello = hello.Replace((char)i,(char)(i+3));
```

Console.WriteLine("Зашифрованная строка:\n {0}",hello);

```
Console.ReadLine();

}

}

In file:///C:/projects/test/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1....

| File:///C:/projects/test/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1....

| Mcxoдная строка: Привет, меня зовут Александр Ерохин Я рад вас приветствовать на моем сайте professorweb.ru

| Зашифрованная строка: Тулеих, пир? ксецх Гоинфгрзу Иусшлр в угз егф тулеихфхесегхя рг псип фгмхи surihuvruzhe.ux
```

Рисунок 5 – Результат выполнения программы

Содержание отчета:

- 1. Номер и тема лабораторной работы.
- 2. Цель лабораторной работы.
- 3. Техническое оснащение.
- 4. Скриншоты выполнения примеров
- 5. Вывод по лабораторной работе