ФГБОУ ВПО

ДВГУПС

Кафедра: ИТИС

Лабораторная работа №6

Выполнил: Точилкин М.А.

Группа 220

Проверил: Володажский А.А.

Хабаровск, 2015

Цель: Познакомиться со строковыми функциями.

Задание 1. Пользователь вводит предложение со знаками препинания. Требуется проставить пробелы против знаков препинания.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication5

{

class Program

{

static void Main(string[] args)

{

string stroka = Console.ReadLine();

stroka = stroka.Replace(" ", string.Empty);

stroka = stroka.Replace(",", ", ");

Console.WriteLine(stroka);

Console.ReadKey();

}

}

}

Задание 2. Пользователь вводит текст. Требуется вывести на экран все слова заключенные в двойные или одинарные кавычки.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication13

{

class Program

{

static void Main(string[] args)

{

string slovo = Console.ReadLine();

string[] massiv = slovo.Split(' ');

int massiv\_len = massiv.Length;

for (int i = 0; i < massiv\_len; i++)

{

if ((massiv[i].Substring(0, 1) == ("'") && (massiv[i].Substring(massiv[i].Length - 1, 1) == ("'"))))

{

Console.WriteLine(massiv[i]);

}

}

for (int i = 0; i < massiv\_len; i++)

{

if ((massiv[i].Substring(0, 1) == ("\"") && (massiv[i].Substring(massiv[i].Length - 1, 1) == ("\""))))

{

Console.WriteLine(massiv[i]);

}

}

Console.ReadKey();

}

}

}

Задание 3. Пользователь вводит текст, требуется в зеркальном порядке каждое слово вывести на экран. Порядок слов остается тем же.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication5

{

class Program

{

static void Main(string[] args)

{

string stroka = Console.ReadLine();

string word = "";

int chislo;

string[] words = stroka.Split(' ');

int words\_len = words.Length;

for (int z = 0; z < words\_len; z++)

{

int len\_massiv\_now = words[z].Length;

for (int i = 0; i < len\_massiv\_now; i++)

{

chislo = words.Length;

word = word + words[z][len\_massiv\_now - 1 - i];

}

words[z] = word;

word = "";

}

for (int z = 0; z < words\_len; z++)

{

Console.Write(words[z] + " ");

}

Console.ReadKey();

}

}

}

Задание 4. Пользователь вводит текст. Проверить является ли введённый текст палиндромом.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication13

{

class Program

{

static void Main(string[] args)

{

string slovo = Console.ReadLine();

int i = 0;

slovo = slovo.Replace(" ", "").Replace(",", "");

for (int k = 0; k < slovo.Length; k++)

{

if (slovo.Substring(k, 1) == slovo.Substring(slovo.Length - k - 1, 1))

{

i++;

}

}

if (i == slovo.Length)

{

Console.WriteLine("Фразу можно читать наоборот. Это палиндром.");

}

else

{

Console.WriteLine("Фразу нельзя читать наоборот. Это не палиндром.");

}

Console.ReadKey();

}

}

}

Задание 5. Пользователь вводит Ф.И.О требуется вывести на экран фамилию и инициалы. Предусмотреть вариант с отсутствием отчества.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication5

{

class Program

{

static void Main(string[] args)

{

string stroka = Console.ReadLine();

string[] words = stroka.Split(' ');

int words\_len = words.Length;

string familya = words[0];

string imya = words[1];

char otchastvo\_first = ' ';

if (words\_len == 3)

{

string otchastvo = words[2];

otchastvo\_first = otchastvo[0];

}

string fio = familya + " " + imya[0] + " " + otchastvo\_first;

Console.WriteLine(fio);

Console.ReadKey();

}

}

}

Задание 6. Пользователь вводит две строки найти максимальную длинную последовательность символов имеющуюся в обоих строках.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

//Главные переменные

string line\_1 = Console.ReadLine();

string line\_2 = Console.ReadLine();

//Вторичные перевернутые

string line\_1\_rev = ReverseArray(line\_1);

string line\_2\_rev = ReverseArray(line\_2);

//Длины строк

int line\_1\_L = line\_1.Length;

int line\_2\_L = line\_2.Length;

//Буферные массивы, с размерностью треугольных чисел

int treug\_1 = TriangularNumber(line\_1\_L);

int treug\_2 = TriangularNumber(line\_2\_L);

string[] buffer\_line\_1 = new string[treug\_1];

string[] buffer\_line\_2 = new string[treug\_2];

string[] buffer\_line\_rev\_1 = new string[treug\_1];

string[] buffer\_line\_rev\_2 = new string[treug\_2];

//Длинна буферных массивов

int buffer\_line\_1\_L = buffer\_line\_1.Length;

int buffer\_line\_2\_L = buffer\_line\_2.Length;

int buffer\_line\_rev\_1\_L = buffer\_line\_rev\_1.Length;

int buffer\_line\_rev\_2\_L = buffer\_line\_rev\_2.Length;

//Код вывода

Console.WriteLine();

Console.WriteLine("Первая строка: " + line\_1);

Console.WriteLine("Вторая строка: " + line\_2);

Console.WriteLine("Первая строка наоборот: " + line\_1\_rev);

Console.WriteLine("Вторая строка наоборот: " + line\_2\_rev);

EnumerationValues(ref line\_1\_L, ref buffer\_line\_1, ref line\_1);

EnumerationValues(ref line\_2\_L, ref buffer\_line\_2, ref line\_2);

EnumerationValues(ref line\_1\_L, ref buffer\_line\_rev\_1, ref line\_1\_rev);

EnumerationValues(ref line\_2\_L, ref buffer\_line\_rev\_2, ref line\_2\_rev);

Console.WriteLine();

PrintArray(buffer\_line\_1);

PrintArray(buffer\_line\_2);

PrintArray(buffer\_line\_rev\_1);

PrintArray(buffer\_line\_rev\_2);

//Цикл сравнения

string buffer\_slovo = "";

int bufer\_slovo\_len = 0;

for (int a = 0; a < buffer\_line\_1\_L; a++)

{

for (int b = 0; b < buffer\_line\_2\_L; b++)

{

for (int c = 0; c < buffer\_line\_rev\_1\_L; c++)

{

for (int s = 0; s < buffer\_line\_rev\_2\_L; s++)

{

if (buffer\_line\_1[a] == buffer\_line\_2[b])

{

if (buffer\_line\_1[a].Length > bufer\_slovo\_len)

{

bufer\_slovo\_len = buffer\_line\_1[a].Length;

buffer\_slovo = buffer\_line\_1[a];

}

}

if (buffer\_line\_1[a] == buffer\_line\_rev\_2[s])

{

if (buffer\_line\_1[a].Length > bufer\_slovo\_len)

{

bufer\_slovo\_len = buffer\_line\_1[a].Length;

buffer\_slovo = buffer\_line\_1[a];

}

}

if (buffer\_line\_1[a] == buffer\_line\_rev\_1[c])

{

if (buffer\_line\_1[a].Length > bufer\_slovo\_len)

{

bufer\_slovo\_len = buffer\_line\_1[a].Length;

buffer\_slovo = buffer\_line\_1[a];

}

}

if (buffer\_line\_2[b] == buffer\_line\_rev\_1[c])

{

if (buffer\_line\_2[b].Length > bufer\_slovo\_len)

{

bufer\_slovo\_len = buffer\_line\_2[b].Length;

buffer\_slovo = buffer\_line\_2[b];

}

}

if (buffer\_line\_2[b] == buffer\_line\_rev\_2[s])

{

if (buffer\_line\_2[b].Length > bufer\_slovo\_len)

{

bufer\_slovo\_len = buffer\_line\_2[b].Length;

buffer\_slovo = buffer\_line\_2[b];

}

}

if (buffer\_line\_rev\_1[c] == buffer\_line\_rev\_2[s])

{

if (buffer\_line\_rev\_1[c].Length > bufer\_slovo\_len)

{

bufer\_slovo\_len = buffer\_line\_rev\_1[c].Length;

buffer\_slovo = buffer\_line\_rev\_1[c];

}

}

}

}

}

}

Console.WriteLine("Комбинация часто встречающаяся: " + ReverseArray(buffer\_slovo));

Console.ReadKey();

}

//Функция переворота

static string ReverseArray(string line)

{

string buffer = "";

int line\_L = line.Length;

for (int i = 0; i < line\_L; i++)

{

buffer = buffer + line[line\_L - i - 1];

}

line = buffer;

return line;

}

//Цикл перебора

static string[] EnumerationValues(ref int len, ref string[] buffer, ref string line)

{

int z = 0;

for (int x = 0; x < len; x++)

{

for (int y = len - x; y > 0; y--)

{

buffer[z] = line.Substring(x, y);

z++;

}

}

return buffer;

}

//Печать на экран

static void PrintArray(string[] array)

{

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

{

Console.Write(array[i] + " ");

}

Console.WriteLine();

}

//Получение треугольных чисел

static int TriangularNumber(int n)

{

double n1 = (0.5) \* n \* (n + 1);

n = Convert.ToInt32(n1);

return n;

}

}

}