**6 СТРОКИ. КЛАССЫ STRING И STRINGBUILDER**

Задание №1. Дан фрагмент текста, запрашиваемый у пользователя. Написать программу, находящую наибольшее количество цифр, идущих в нем подряд.

Листинг программы:

try

{

Console.Write("Введите строку: ");

string input = Console.ReadLine();

int carent = default;

int max = default;

int count = 0;

int maxChar = 0;

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

{

if (char.IsDigit(input[i]))

{

carent += 1;

}

else if (carent > max)

{

max = carent;

carent = 0;

}

}

if (carent > max)

{

max = carent;

}

Console.WriteLine(max);

Console.ReadLine();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Таблица 6.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 123a1234 | 4 |

Анализ результатов:



Рисунок 6.1 – Результаты работы программы

Источник: собственная разработка

Задание №2 Дано предложение. Определить количество букв «a», предшествующих первой запятой предложения. Рассмотреть два случая: 1) известно, что запятые в предложении есть; 2) запятых в предложении может не быть.

Листинг программы:

try

{

Console.Write("Введите строку: ");

string input = Console.ReadLine();

var pos = input.IndexOf(',');

int count = 0;

if (pos > 0)

{

for (int i = 0; i < pos; i++)

{

if (input[i] == 'a')

{

count += 1;

Console.WriteLine(input[i]);

}

}

}

else

{

Console.WriteLine("Запятые отсутствуют");

}

Console.WriteLine(count);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Таблица 6.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| data,123 | a  a  2 |

Анализ результатов:

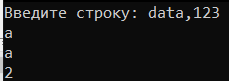


Рисунок 6.2 – Результаты работы программы

Источник: собственная разработка

Задание №3. С клавиатуры вводится предложение. Результаты всех действий вывести на экран. Поменять местами первое и последнее слова в предложении. Склеить второе и третье слова в предложении. Третье слово в предложении вывести в обратном порядке. В первом слове предложения вырезать первые две буквы.

Листинг программы:

try

{

Console.Write("Enter line: ");

string line = Console.ReadLine().ToLower();

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

if (words.Length >= 2)

{

var temp = words[words.Length - 1];

words[words.Length - 1] = words[0];

words[0] = temp;

line = String.Join(' ', words);

Console.WriteLine(line);

if (words.Length >= 3)

{

line = line.Remove(line.LastIndexOf(' '), 1);

Console.WriteLine(line);

words = line.Split(' ');

words[2] = Reverse(words[2]);

line = String.Join(' ', words);

Console.WriteLine(line);

}

else { Console.WriteLine("Less then 3 words!"); }

}

if (words.Length >= 1 && words[0].Length >= 2)

{

line = line.Substring(2);

Console.WriteLine(line);

}

else

{

Console.WriteLine("First word lower then 2 chars!");

}

}

catch (Exception e)

{

Console.WriteLine($"Some error ({e})");

}

static string Reverse(string s)

{

char[] charArray = s.ToCharArray();

Array.Reverse(charArray);

return new string(charArray);

}

Таблица 6.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| hiqwe moww qwen qwejn | qwejn moww qwen hiqwe  qwejn moww qwenhiqwe  qwejn moww ewqihnewq  ejn moww ewqihnewq |

Анализ результатов:

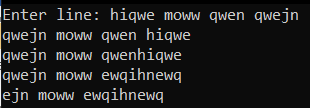


Рисунок 6.3 – Результаты работы программы

Источник: собственная разработка

Задание №4. Известны фамилия, имя и отчество пользователя. Найти его код личности. Правило получения кода личности: каждой букве ставится в соответствие число - порядковый номер буквы в алфавите. Эти числа складываются. Если полученная сумма не является однозначным числом, то цифры числа снова складываются и так до тех пор, пока не будет получено однозначное число. Например, исходные данные: Александр Сергеевич Пушкин.

Листинг программы:

using System.Text;

internal class Program

{

public static string Alphabet = GetAplhabetString();

static string GetAplhabetString()

{

var str = new StringBuilder();

for (char i = 'а'; i <= 'я'; i++)

{

str.Append(i.ToString());

}

return str.ToString();

}

public static int[] GetNumberCodes(string text)

{

int[] array = new int[text.Length];

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

{

array[i] = Alphabet.IndexOf(text[i]) + 1;

}

return array;

}

public static int SumDigits(int Number)

{

int Sum = 0;

while (Number > 0)

{

Sum += Number % 10;

Number /= 10;

}

return Sum;

}

static void Main()

{

try

{

string Text;

Console.Write("Введите строку: ");

Text = Convert.ToString(Console.ReadLine());

Text = Text.Replace(" ", "").ToLower();

int[] mas = GetNumberCodes(Text);

int Sum = mas.Sum();

while (Sum > 9)

{

Sum = SumDigits(Sum);

}

Console.WriteLine(Sum);

Console.ReadLine();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

}

Таблица 6.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Александр Сергеевич Пушкин | 3 |

Анализ результатов:



Рисунок 6.4 – Результаты работы программы

Источник: собственная разработка