**6 Строки. Классы String и StringBuilder**

Задание №1. Дан фрагмент текста, запрашиваемый у пользователя. Написать программу, проверяющую, является ли она палиндромом, то есть читается ли одинаково слева направо и справа налево. При проверке регистр букв, знаки препинания и пробелы не учитываются. Например, строка «Аргентина манит негра.» – палиндром.

Листинг программы:

try

{

Console.Write("Enter line: ");

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

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

if (text == Reverse(text))

{

Console.WriteLine("True");

} else { Console.WriteLine("False"); }

}

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 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Аргентина манит негра | True |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.1.



Рисунок 1.1 – Результат работы программы

Источник: собственная разработка

Задание №2. Ввести текст, состоящий только из цифр и букв. Выяснить, верно ли, что сумма числовых значений цифр, находящихся в тексте, равна длине текста.

Листинг программы:

string inputStr = Console.ReadLine();

if (!IsCorrectString(inputStr))

{

Console.WriteLine("Необходимо ввести только цифры и буквы");

return;

}

int sum = 0;

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

{

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

{

sum += Convert.ToInt32(inputStr[i].ToString());

}

}

if (sum == inputStr.Length)

{

Console.WriteLine("Сумма цифр в строке = длине строки");

}

else

{

Console.WriteLine("Сумма цифр в строке != длине строки");

}

static bool IsCorrectString(string inputStr)

{

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

{

if (!char.IsLetterOrDigit(inputStr[i]))

{

return false;

}

}

return true;

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Uhygvftrg55hy4 | True |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.2.



Рисунок 1.2 – Результат работы программы

Источник: собственная разработка

Задание №3. С клавиатуры вводится предложение. Результаты следующих действий вывести на экран:

поменять местами первое и последнее слова в предложении;

склеить второе и третье слова в предложении;

третье слово предложения 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 1.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Hey how are you | u how yehera |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.3.

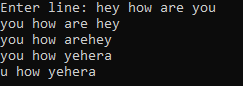


Рисунок 1.3 – Результат работы программы

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 число складываются. Если полученная сумма не является однозначным числом, то цифры числа снова складываются и так до тех пор, пока не будет получено однозначное число.

Листинг программы:

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()

{

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();

}

}

Таблица 1.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| String | 3 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.4.



Рисунок 1.4 – Результат работы программы

Источник: собственная разработка