**6 СТРОКИ. КЛАССЫ STRING И STRINGBUILDER**

Задание №1. Удалить из сообщения все однобуквенные слова (вместе с лишними пробелами).

Листинг программы:

static void Main(string[] args)

{

try

{

Console.Write("Enter string: ");

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

sentence = Delete(sentence).Trim();

string[] letters = sentence.Split(' ');

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

{

if (letters[i].Length == 1)

{

letters[i] = "";

}

}

sentence = String.Join(' ', letters).Trim();

Console.WriteLine(Delete(sentence));

}

catch (Exception e)

{

Console.WriteLine($"Error! ({e})");

}

}

static string Delete(string s, int k = 1)

{

char[] rs = new char[] { ' ', '(', '[', '{' };

if (k >= s.Length)

{

if (s[k - 1] == ' ') s = s.Remove(k - 1, 1);

return s;

}

else

{

if ((s[k] == ' ') && (rs.Contains(s[k - 1])))

{

s = s.Remove(k, 1);

}

else

{

k++;

}

return Delete(s, k);

}

}

Таблица 6.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Мы не купались, а бегали | Мы не купались, бегали |

Источник: собственная разработка

Анализ результатов представлен на рисунке 6.1.



Рисунок 6.1 – Результат работы программы

Источник: собственная разработка

Задание №2. Составить программу, которая будет вводить строку в переменную string. Найти слово, встречающееся в каждом предложении, или сообщить, что такого слова нет.

Листинг программы:

static void Main(string[] args)

{

string s, s1;

int c = 0;

Console.Write("Enter any sentece: ");

s = Console.ReadLine();

Console.Write("Enter word: ");

s1 = Console.ReadLine();

string[] ss = s.Split(' ');

foreach (var str in ss)

{

if (str == s1)

{

++c;

}

}

Console.WriteLine($"Amount of words: {Convert.ToString(c)}");

}

Таблица 6.2 – Входные и выходные данные

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

Источник: собственная разработка

Анализ результатов представлен на рисунке 6.2.

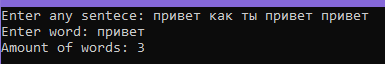


Рисунок 6.2 – Результат работы программы

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:

static void Main(string[] args)

{

try

{

Console.Write("Enter any sentence: ");

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

string[] letters = sentence.Split(' ');

if (letters.Length >= 2)

{

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

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

letters[0] = temp;

sentence = String.Join(' ', letters);

Console.WriteLine(sentence);

if (letters.Length >= 3)

{

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

Console.WriteLine(sentence);

letters = sentence.Split(' ');

letters[2] = ReverseWord(letters[2]);

sentence = String.Join(' ', letters);

Console.WriteLine(sentence);

}

else { Console.WriteLine("under 3 letters"); }

}

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

{

sentence = sentence.Substring(2);

Console.WriteLine(sentence);

}

else

{

Console.WriteLine("First word is not in correct format(< 2 letters)");

}

}

catch (Exception e)

{

Console.WriteLine($"Error! ({e})");

}

}

static string ReverseWord(string s)

{

char[] charArray = s.ToCharArray();

Array.Reverse(charArray);

return new string(charArray);

}

Таблица 6.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| String | eksey my eman ishello |

Источник: собственная разработка

Анализ результатов представлен на рисунке 6.3.

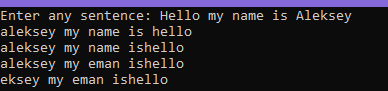


Рисунок 6.3 – Результат работы программы

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:

public static string Alphabet = GetAplhabetString();

static string GetAplhabetString()

{

StringBuilder 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()

{

Console.Write("Enter Name, Patronymic, Surname: ");

string text = Console.ReadLine().Replace(" ", "").ToLower();

int[] mas = GetNumberCodes(text);

int Sum = mas.Sum();

while (Sum > 9)

{

Sum = SumDigits(Sum);

}

Console.WriteLine(Sum);

}

Таблица 6.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| String | 4 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 6.4.



Рисунок 6.4 – Результат работы программы

Источник: собственная разработка