**6.** **Строки. Классы String и StringBuilder**

Задание №1. Дан фрагмент текста, запрашиваемый у пользователя. Фрагмент содержит слова, разделенные пробелами (одним или несколькими). Написать программу, выводящую те же слова через один пробел в обратном порядке.

Листинг программы:

using System.Text.RegularExpressions;

Console.Write("Введите текст: ");

string text = Console.ReadLine();

Regex regex = new Regex(@"[\w]+");

MatchCollection matches = regex.Matches(text);

List<string> words = new List<string>();

foreach (Match match in matches)

{

words.Add(match.Value);

}

words.Reverse();

foreach (string word in words)

{

Console.Write($"{word} ");

}

Таблица 6.1 – Входные и выходные данные программы задание №1

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Первое фыва фывафыва фыва фывоарфлыовар | фывоарфлыовар фыва фывафыва фыва Первое |

Анализ результатов:



Рисунок 6.1 – Результат выполнения программы задание №1

Задание №2. Среди цифр введенной строки распечатать ту, которая появлялась чаще других. Если таких цифр было несколько, распечатать ту, что встретилась первой.

Листинг программы:

using System.Text.RegularExpressions;

Console.Write("Введите числа: ");

string text = Console.ReadLine();

Dictionary<int, int> nums = CountNumberOfNumbersIsStr(text);

WriteDictionary(nums);

int keyOfMaxValue = GetKeyAtMaxValueInDictionary(nums);

Console.WriteLine();

Console.WriteLine($"key: {keyOfMaxValue} value: {nums[keyOfMaxValue]}");

Dictionary<int, int> CountNumberOfNumbersIsStr(string text)

{

Regex regex = new Regex(@"[\w]+ ");

MatchCollection matches = regex.Matches(text);

Dictionary<int, int> nums = new Dictionary<int, int>();

foreach (Match match in matches)

{

Console.WriteLine(match.Value);

if (Int32.TryParse(match.Value, out int num))

{

if (nums.TryGetValue(num, out int value))

{

nums[num]++;

}

else

{

nums.Add(num, 1);

}

}

}

return nums;

}

void WriteDictionary(Dictionary<int, int> nums)

{

foreach (var num in nums)

{

Console.WriteLine($"key: {num.Key} value: {num.Value}");

}

}

int GetKeyAtMaxValueInDictionary(Dictionary<int, int> nums) => nums.Aggregate

((x, y) => x.Value >= y.Value ? x : y).Key;

Таблица 6.2 – Входные и выходные данные программы задание №2

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 123 232 2 45 2 33 3 3 | Ключ: 2 Значение: 2 |

Анализ результатов:

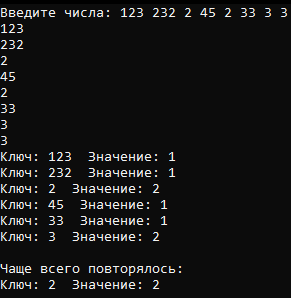


Рисунок 6.2 – Результат выполнения программы задание №2

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 предложения вывести в обратном порядке

- В первом слове предложения вырезать первые две буквы.

Листинг программы:

Console.Write("Введите текст: ");

string text = Console.ReadLine();

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

words = SwapFirstWithSecond(words);

words[1] += words[2];

words[2].Reverse();

words[0] = words[0].Remove(0,2);

foreach (string word in words)

{

Console.Write($"{word} ");

}

string[] SwapFirstWithSecond(string[] arr)

{

(arr[0], arr[arr.Length - 1]) = (arr[arr.Length - 1], arr[0]);

return arr;

}

Таблица 6.3 – Входные и выходные данные программы задание №3

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Первое второе третье четвертое | твертое второетретье третье Первое |

Анализ результатов:



Рисунок 6.3 – Результат выполнения программы задание №3

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,

Листинг программы:

const string Alphabet = @"абвгдеёжхийклмнопрстуфхцчшщъыьэюя";

Console.Write("Введите ФИО: ");

string[] fullName = Console.ReadLine().Split(' ');

int num = 0;

foreach (string word in fullName)

{

num += SumIndexInStr(word);

}

int code = ToDetermineCodeNumber(num);

Console.WriteLine($"{code}");

int SumIndexInStr(string str)

{

int num = 0;

string strLower = str.ToLower();

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

{

num += Alphabet.IndexOf(strLower[i]) + 1;

}

return num;

}

int ToDetermineCodeNumber(int num)

{

int code = 0;

while (num != 0)

{

code += num % 10;

num /= 10;

if (code > 9 && num == 0)

{

num = code;

code = 0;

}

}

return code;

}

Таблица 6.4 – Входные и выходные данные программы задание №4

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Александр Сергеевич Пушкин | 9 |

Анализ результатов:



Рисунок 6.4 – Результат выполнения программы задание №4