**7 Регулярные выражения**

Задание 1. Для данного задания все операции поиска и разбиения необходимо производить с помощью регулярных выражений. Найти в тексте слова, содержащие не более одного дефиса.

Листинг программы:

Console.WriteLine("Напишите текст: "); string text = Console.ReadLine();

Regex regex = new Regex(@"\b\w\*-\w\*\b");

Console.WriteLine();

Console.WriteLine("Слова содержащие не более одного дефиса: ");

MatchCollection matches = regex.Matches(text);

foreach (Match match in matches)

{Console.WriteLine(match.Value);}

Таблица 7.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Почему-то человечество усложняет жизнь невесомо--существенными проблемами | Слова содержащие не более одного дефиса:  Почему-то |

Источник: собственная разработка

Анализ результатов:

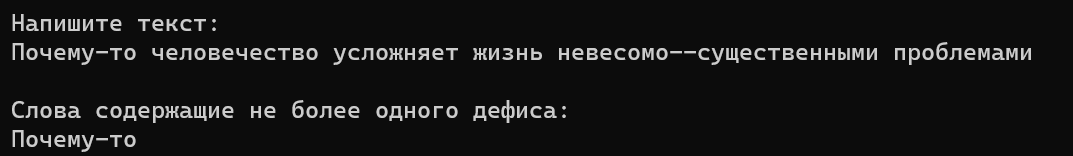


Рисунок 7.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Для данного задания все операции поиска и разбиения необходимо производить с помощью регулярных выражений. Задан текст. Определить, является ли он текстом на английском языке.

Листинг программы:

Console.WriteLine("Введите текст, а мы определим, на английском ли он: ");

string text = Console.ReadLine();

Regex regex = new Regex("[^a-zA-Z]");

string cleanText = regex.Replace(text, "");

if (!Regex.IsMatch(cleanText, @"^[a-zA-Z]+$"))

{ Console.WriteLine("Введенный текст не является на английском языке");}

Else

{Console.WriteLine("Введенный текст является на английском языке");}

Таблица 7.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| This pizza is very tasty | Введенный текст является на английском языке |

Источник: собственная разработка

Анализ результатов:

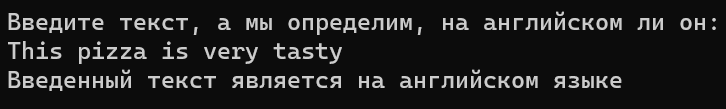


Рисунок 7.2 – Результат работы программы

Источник: собственная разработка

Задание 3. Для данного задания все операции поиска и разбиения необходимо производить с помощью регулярных выражений. Задан текст. Выбрать из него все e-mail адреса.

Листинг программы:

Console.Write("Введите предложение: "); string sentence = Console.ReadLine();

string sentence1 = sentence; string sentence2 = sentence;

string sentence3 = sentence; string sentence4 = sentence;

string[] words = sentence1.Split(' '); string temp = words[0];

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

sentence1 = string.Join(" ", words);

Console.WriteLine("1.Переставим первым и последним словами: "); Console.WriteLine(sentence1);

words = sentence2.Split(' ');

words[1] += words[2];

for (int i = 2; i < words.Length - 1; i++){words[i] = words[i + 1];}

Array.Resize(ref words, words.Length - 1);

sentence2 = string.Join(" ", words); Console.WriteLine("2.Объединим вторым и третьим словами: "); Console.WriteLine(sentence2);

words = sentence3.Split(' ');

char[] chars = words[2].ToCharArray();

Array.Reverse(chars); string reversed = new string(chars);

Console.WriteLine("3.Третье слово в обратном порядке: "); Console.WriteLine(sentence3);

words = sentence4.Split(' ');

words[0] = words[0].Substring(2);

sentence4 = string.Join(" ", words);

Console.WriteLine("4.Первое слово без первых двух букв: "); Console.WriteLine(sentence4);

Таблица 7.3 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Сегодня тарелку мишки разбил его брат, другой мишка | 1.Переставим первым и последним словами:  мишка тарелку мишки разбил его брат, другой Сегодня  2.Объединим вторым и третьим словами:  Сегодня тарелкумишки разбил его брат, другой мишка  3.Третье слово в обратном порядке:  Сегодня тарелку мишки разбил его брат, другой мишка  4.Первое слово без первых двух букв:  годня тарелку мишки разбил его брат, другой мишка |

Источник: собственная разработка

Анализ результатов:

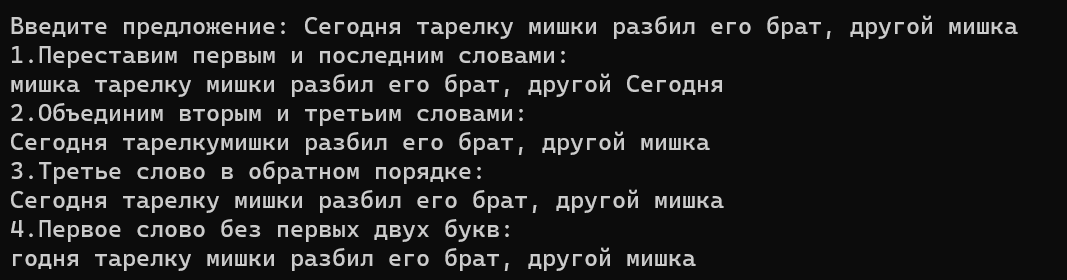


Рисунок 7.3 – Результат работы программы

Источник: собственная разработка