Министерство науки и высшего образования Российской Федерации Рязанский государственный радиотехнический университет

имени В.Ф. Уткина

Кафедра «Электронные вычислительные машины»

**Отчёт**

о лабораторной работе №4

«Регулярные выражения»

по дисциплине:

«Визуальное программирование»

Выполнили:

Студенты группы 140

Бригады №1

Волкова Е.В.

Егорова Е.М.

Поверил:

ст. преп. Бастрычкин А.С.

ст. преп. Хизриева Н. И.

Рязань 2024

**Цель работы**: изучить работу с регулярными выражениями на языке C#.

**Задание**:

1) Написать метод, который определяет количество входящих в заданную строку почтовых индексов (почтовый индекс состоит из 6 цифр).

2) Дана строка — предложение на русском языке. Поменять местами первую и последнюю буквы каждого слова.

3) Дана строка, содержащая помимо прочей информации номера телефонов в федеральном формате. Скрыть все цифры городской части номера кроме самой первой под символами x. Например, если в тексте имелся номер +7 (863) 297-51-11, то после преобразования он должен выглядеть как +7 (863) 2xx-xx-xx. Считать, что код города может содержать от трёх до пяти цифр, а городской номер — от 7 до 5 цифр соответственно.

4) Выяснить, какими могут быть российские автомобильные номера (с кодом региона), составить соответствующее регулярное выражение и написать метод, который находит в строке все автомобильные номера и возвращает их в виде последовательности.

5) Дана строка. Сохранить в новую строку все содержащиеся в ней IPv4-адреса в десятичной записи с точками через разделитель.

**Практическая часть.**

Код программы с описанием класса RegularExpressions:

using System.Text.RegularExpressions;

namespace VP\_Lab4

{

public class RegularExpressions

{

public static int CountPostalCodes(string input)

{

string pattern = @"\b\d{6}\b";

Regex rg = new Regex(pattern);

return rg.Matches(input).Count;

}

public static string SwapFirstAndLast(string input)

{

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

StringBuilder sb = new StringBuilder();

foreach (string word in words)

{

if (word.Length > 1)

{

char firstChar = word[0];

char lastChar = word[word.Length - 1];

sb.Append(lastChar);

sb.Append(word.Substring(1, word.Length - 2));

sb.Append(firstChar);

}

else

{

sb.Append(word);

}

sb.Append(" ");

}

return sb.ToString().Trim();

}

public static string HideCityNumbers(string input)

{

//string pattern = @"\+7 \(\d{3,5}\) \d{1,3}-(\d{2})-(\d{2})";

string hiddenPart = @"\d{1,3}-(\d{2})-(\d{2})";

Regex rg = new Regex(hiddenPart);

string cityPart = rg.Match(input).Groups[0].Value;

int cityLength = cityPart.IndexOf('-');

string hiddenCityPart = cityPart[0] + new string('x', cityLength - 1)+"-xx-xx";

string replacedInput = Regex.Replace(input, hiddenPart, hiddenCityPart);

return replacedInput;

}

public static IEnumerable<string> FindCarNumbers(string input)

{

Regex regex = new Regex(@"[АВЕКМНОРСТУХ]{1}\d{3}[АВЕКМНОРСТУХ]{2}\d{2,3}");

MatchCollection matches = regex.Matches(input);

foreach (Match match in matches)

{

yield return match.Value;

}

}

public static IEnumerable<string> ExtractIPv4Addresses(string input)

{

Regex regex = new Regex(@"\b(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\b");

MatchCollection matches = regex.Matches(input);

foreach (Match match in matches)

{

yield return match.Value;

}

}

}

}

Код программы с описанием интерфейса:

namespace VP\_Lab4

{

public class Program

{

static void Main()

{

RegularExpressions regular = new RegularExpressions();

Console.WriteLine("Задача 1");

string input1 = "123456 654321 987654 bn4546 4567899"; Console.WriteLine(input1.ToString());

Console.WriteLine($"Количество почтовых индексов: {RegularExpressions.CountPostalCodes(input1)}");

string input2 = "Регулярные выражения поддерживаются множеством классов .NET";

Console.WriteLine("\nЗадача 2"); Console.WriteLine(input2.ToString());

Console.WriteLine("Меняем местами первую и последнюю буквы каждого слова:");

Console.WriteLine(RegularExpressions.SwapFirstAndLast(input2));

string input3 = "Текст с номером +7 (863) 297-51-11 и еще каким-то номером +7 (495) 123-45-67";

Console.WriteLine("\nЗадача 3");

Console.WriteLine(RegularExpressions.HideCityNumbers(input3));

Console.WriteLine("\nЗадача 4");

string input4 = "А123ВС62, Х567РП67, Е999ЕЕ62, 777RUS77, 1234567";

Console.WriteLine(input4.ToString() + "\nРоссийские автомобильные номера: ");

IEnumerable<string> carNumbers = RegularExpressions.FindCarNumbers(input4);

foreach (string number in carNumbers)

{

Console.WriteLine(number);

}

Console.WriteLine("\nЗадача 5");

string input5 = "Текст с IP-адресами 192.168.0.1 и 10.0.0.1, а также 345.4.34.0";

Console.WriteLine(input5.ToString() + "\nIP-адреса: ");

IEnumerable<string> ipAddresses = RegularExpressions.ExtractIPv4Addresses(input5);

foreach (string ip in ipAddresses)

{

Console.WriteLine(ip);

}

}

}

}

Результат работы программы на рисунке 1.

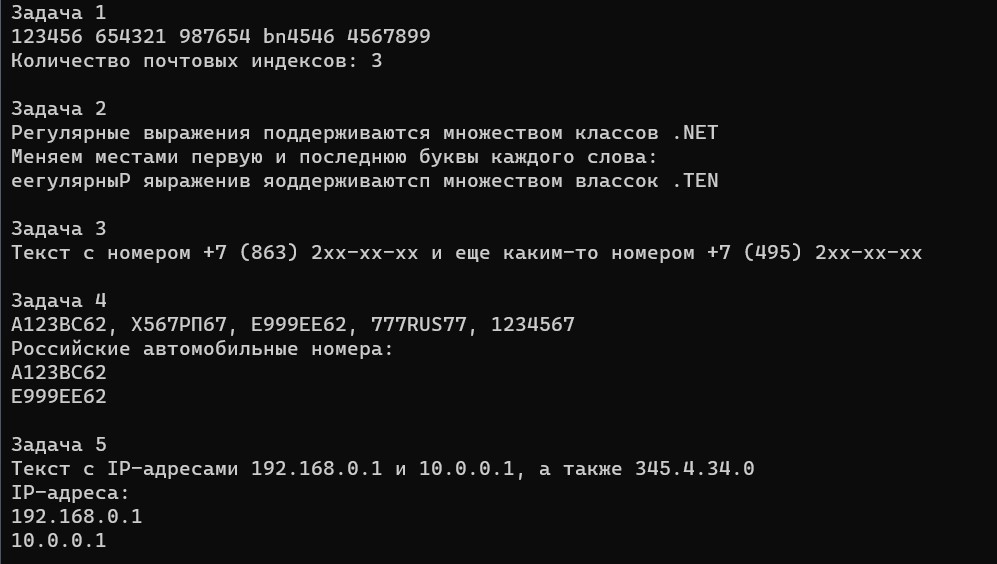


Рисунок 1 – Результат работы программы

**Заключение**

В ходе выполнения лабораторной работы была изучена работа с регулярными выражениями на языке C#.