Лабораторная работа №7

Условие:

1. В сообщении могут встречаться номера телефонов, записанные в формате xx-xx-xx, xxx-xxx или xxx-xx-xx. Вывести все номера телефонов, которые содержатся в сообщении;
2. 1. Шаблоны регулярных выражений для групп time, ip и site записаны в упрощенном виде. Преобразуйте их к такому виду, чтобы они соответствовали ограничениям, накладываемым на формат времени, ip-адреса и адреса web-сайта.

Решение:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

/\*

В сообщении могут встречаться номера телефонов,

записанные в формате xx-xx-xx, xxx-xxx или xxx-xx-xx.

Вывести все номера телефонов, которые содержатся в сообщении.

\*/

Console.WriteLine("Задание 1.");

Boolean Go = true;

Random rand = new Random();

while (Go)

{

back: string str1 = "";

str1 += rand.Next(1, 10);

int buf1 = rand.Next(2);

if (buf1 == 0)

{

for (int i = 0; i < 6; i++)

str1 += rand.Next(10);

Console.WriteLine("{0:000-00-00}", Convert.ToInt32(str1));

goto back;

}

for (int i = 0; i < 5; i++)

str1 += rand.Next(10);

int buf2 = rand.Next(2);

if (buf2 == 0)

Console.WriteLine("{0:00-00-00}", Convert.ToInt32(str1));

else

Console.WriteLine("{0:000-000}", Convert.ToInt32(str1));

int buf3 = rand.Next(40);

if (buf3 == 33)

Go = false;

}

Console.WriteLine();

/\*

1. Шаблоны регулярных выражений для групп

time, ip и site записаны в упрощенном виде.

Преобразуйте их к такому виду, чтобы они соответствовали ограничениям,

накладываемым на формат времени, ip-адреса и адреса web-сайта.

2. Используя дополнительную литературу и Интернет,

более подробно изучите работу с классом Group и коллекцией Groups класса Match.

\*/

Console.WriteLine("Задание 2.");

string text = @"04:55:34 223.345.12.158 www.aaa.ru

04:59:55 213.134.112.56 www.bbb.net

05:05:01 223.34.12.156 www.aaa.by";

Regex theReg = new Regex(@"(?<time>(\d|\:)+)\s" +

@"(?<ip>(\d|\.)+)\s" +

@"(?<site>\S+)");

MatchCollection theMatches = theReg.Matches(text);

foreach (Match theMatch in theMatches)

{

if (theMatch.Length != 0)

{

Console.WriteLine("\ntheMatch: {0}", theMatch.ToString()); //1

Console.WriteLine("time: {0}", theMatch.Groups["time"]); //2

Console.WriteLine("ip: {0}", theMatch.Groups["ip"]); //3

Console.WriteLine("site: {0}", theMatch.Groups["site"]); //4

}

}

Console.WriteLine();

Console.WriteLine("Правильные ограничения:");

Console.WriteLine();

Regex theReg1 = new Regex(@"(?<time1>\d{1,2}:\d{1,2}:\d{1,2})\s" +

@"(?<ip1>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})\s" +

@"(?<site1>\S{1,3}\.\S\*\.by|\S{1,3}\.\S\*\.net)");

MatchCollection thematches1 = theReg1.Matches(text);

foreach (Match sss in thematches1)

{

Console.WriteLine("\ntheMatch: {0}", sss.ToString()); //1

Console.WriteLine("time: {0}", sss.Groups["time1"]); //2

Console.WriteLine("ip: {0}", sss.Groups["ip1"]); //3

Console.WriteLine("site: {0}", sss.Groups["site1"]); //4

}

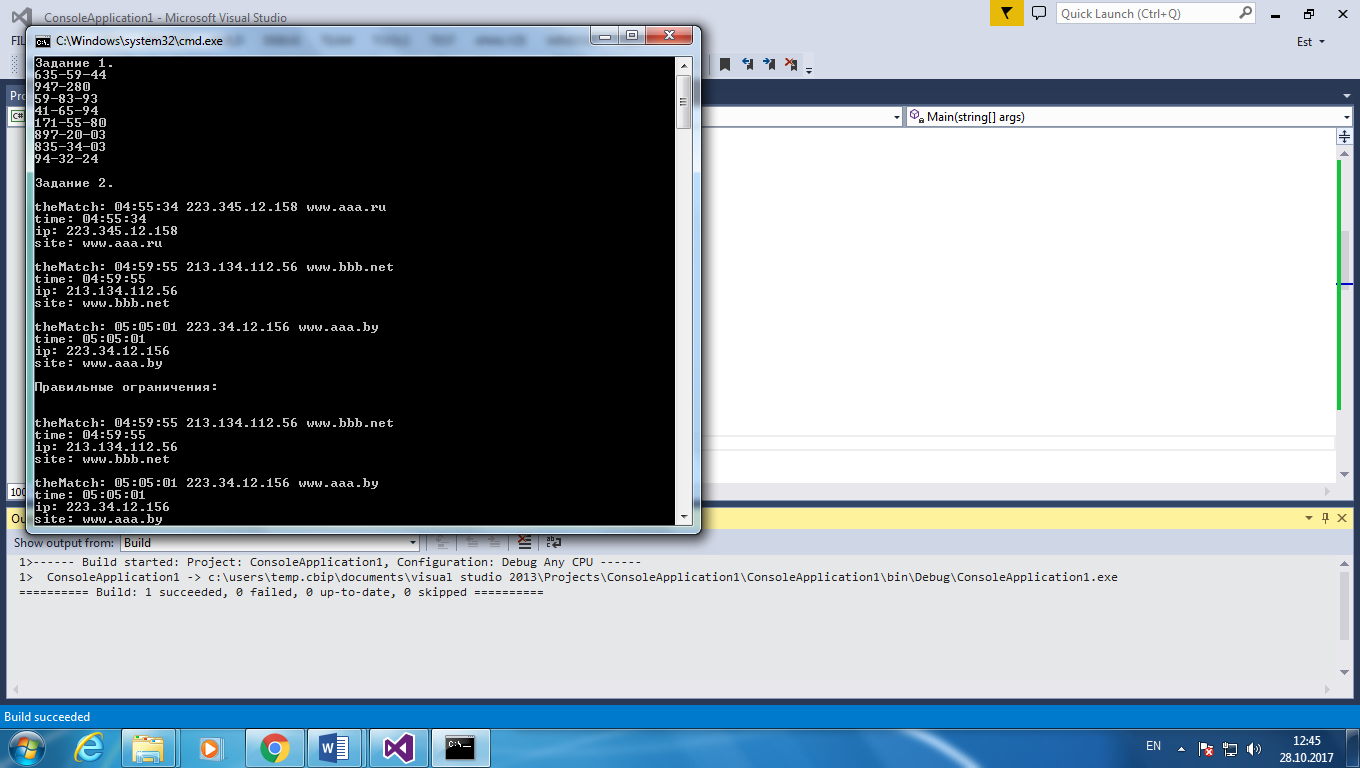
Console.ReadKey();

}

}

}

Тестирование:



**Ответы на контрольные вопросы:**

1. Используя регулярные выражения, вы указывается правила для множества возможных строк, которые вы хотите проверить; это множество может содержать английские фразы, или адреса электронной почты, или TeX команды, или все что угодно. С помощью РВ вы можете задавать вопросы, такие как «Соответствует ли эта строка шаблону?», или «Совпадает ли шаблон где-нибудь с этой строкой?». Вы можете также использовать регулярные выражения, чтобы изменить строку или разбить ее на части различными способами.
2. Обратные ссылки предоставляют удобный способ идентификации повторяющегося символа или подстроки в строке. Например, если входная строка содержит несколько экземпляров произвольной подстроки, то можно найти первое вхождение с помощью группы записи, а затем использовать обратную ссылку для поиска последующих вхождений подстроки.
3. Класс Regex представляет неизменяемое регулярное выражение.