|  |  |
| --- | --- |
|  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего профессионального образования  **«Московский технологический университет»**  **МИРЭА** | | |

Институт Информационных Технологий

Кафедра Корпоративных Информационных Систем

**ОТЧЕТ**

по Лабораторной Работе №6

на тему

«Регулярные выражения»

по дисциплине

«Технологии программирования»

Выполнил студент группы ИСБО-11-16 Шайхуллин С.В.

Принял Cтарший преподаватель Мирзоян Д.И.

Выполнено «\_\_\_» \_\_\_\_\_\_\_\_\_2017 г.

Зачтено «\_\_\_» \_\_\_\_\_\_\_\_\_ 2017г.

Москва, 2017

**Теоретическая часть**

Регулярные выражения являются быстрым, простым и относительно стандартным способом поиска и обработки текстовых данных. Это привело к большой популярности данной технологии, несмотря на ее недостатки. Большинство языков и сред программирования содержат ту или иную библиотеку поддержки регулярных выражений. В C# используется библиотека регулярных выражений, совместимая с Perl 5. Она включена в стандартный набор библиотек и имеет пространство имен System.Text.RegularExpressions.

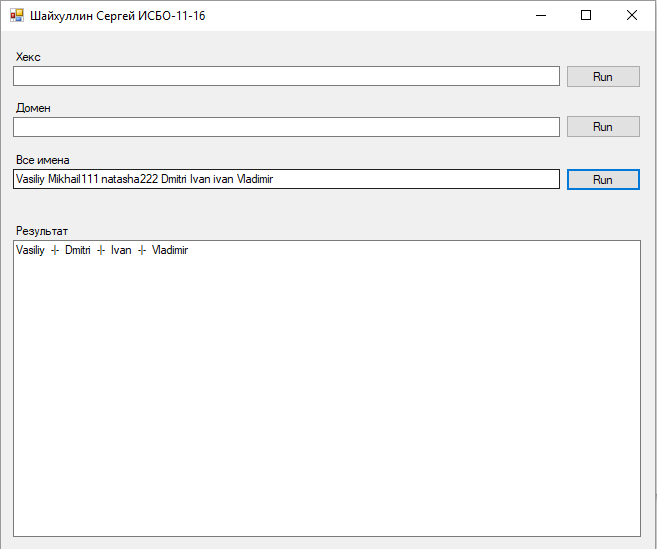
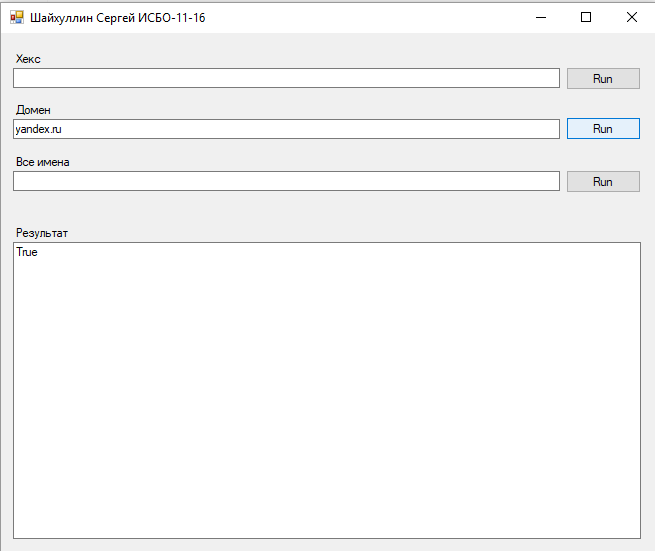
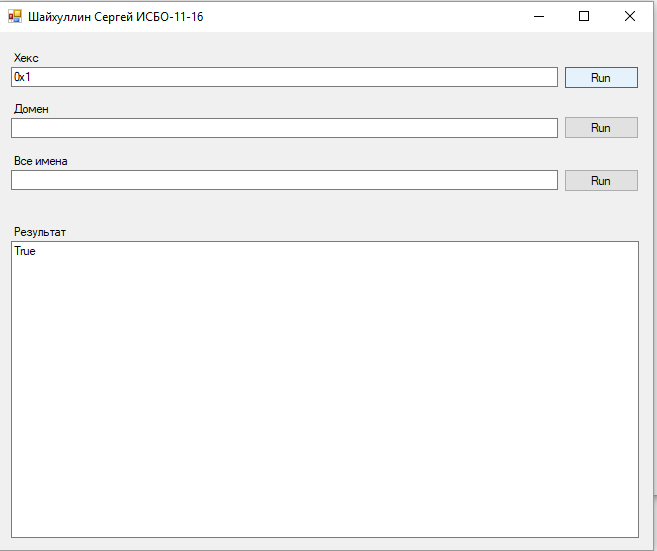
Для работы с регулярными выражениями служит класс Regex. Один экземпляр класса соответствует одному шаблону регулярных выражений. Это связано с необходимостью поддержания сложных структур данных для работы конечного автомата или формальной грамматики, соответствующих данному выражению. Эти структуры обеспечивают высокую производительность.

**Задание**

Вариант 1.

1. Проверить, что строка является корректным шестнадцатеричным числом.
2. Проверить, что строка является корректным доменным именем.
3. Выделить из строки все имена (начинаются с большой буквы, все остальные строчные, только на одном языке)

**Пример работы**



**Заключение**

В данной лабораторной работе мы научились писать не сложные регулярные выражения для фильтрации и валидации текста

**Исходный код**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab6

{

public partial class MainWindow : Form

{

public MainWindow()

{

InitializeComponent();

}

private void OnTask1(object sender, EventArgs e) =>

result.Text = new Regex("^0x[0-9a-fA-F]{1,8}$")

.IsMatch(task1.Text)

.ToString();

private void OnTask2(object sender, EventArgs e) =>

result.Text = new Regex(@"^([a-zA-Z0-9]([a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])?\.)+[a-zA-Z]{2,6}$")

.IsMatch(task2.Text)

.ToString();

private void OnTask3(object sender, EventArgs e) =>

result.Text = string.Join(" -|- ",

new Regex("\\b[A-Z][a-z]+\\b")

.Matches(task3.Text)

.OfType<Match>()

.Select(m => m.Value)

.ToArray());

}

}