**Московский государственный технический университет**

**им. Н.Э. Баумана**

Кафедра ИУ–5

**Отчет по лабораторной работе №6 по курсу «Базовые компоненты интернет технологий»**

Выполнил:

студент гр. ИУ5-32

Деркачев А.В.

\_\_\_\_\_\_\_\_

Проверил:

к.т.н., доцент

Гапанюк Ю.Е.

\_\_\_\_\_\_\_\_

**Москва – 2014 г**

1. **Цель выполнения лабораторной работы**

Разработать программу, использующую делегаты и реализующую работу с рефлексией .

1. **Листинг программы**

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace LR6FT

{ static class Program

{ /// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{ Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());}}}

del1.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LR6FT

{ public delegate string ab(int a, int b);}

pryamoug.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LR6FT

{ class pryamoug

{ public double shir;

public double dlin;

public pryamoug(double dlin, double shir)

{ this.dlin = dlin;

this.shir = shir; }

public override string ToString()

{ return base.ToString() + "\n" + "Длина прямоугольника: " + dlin + "\n" + "Ширина прямоугольника: " + shir + "\n" + "Площадь прямоугольника: " + shir\*dlin + "\n";}

public void print()

{ Console.WriteLine(this.ToString()); }}}

Reflect.cs

using System.Text;

using System.Threading.Tasks;

namespace LR6FT

{ class Reflect

{ public static string MethodReflectInfo<T>(T element) where T: class

{ Type t = typeof(T);

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

foreach (var x in t.GetConstructors())

{ constructors.Add(x.ToString()); }

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

foreach (var x in t.GetMethods())

{ metody.Add(x.ToString()); }

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

foreach (var x in t.GetFields())

{ polya.Add(x.ToString()); }

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

foreach (var x in t.GetProperties())

{ svva.Add(x.ToString()); }

string retstring = "Конструкторы: \n";

foreach (string c in constructors)

{ retstring += c + Environment.NewLine; }

retstring += "Методы: \n";

foreach (string c in metody)

{ retstring += c + Environment.NewLine; }

retstring += "Поля: \n";

foreach (string c in polya)

{ retstring += c + Environment.NewLine; }

retstring += "Свойства: \n";

foreach (string c in svva)

{ retstring += c + Environment.NewLine; }

return retstring;}}}

1. **Результаты работы программы**

