|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

**Отчёт**

***Лабораторная работа №6***

Выполнил студент: \_\_***Сафиуллин Антон Павлович\_\_\_\_\_\_\_\_\_\_***

*фамилия, имя, отчество*

Группа: \_\_\_\_***РТ5-31Б\_\_\_\_\_\_\_\_***

**Описание задания**

**Лабораторная работа №6**

**Часть 1. Разработать программу, использующую делегаты.**

(В качестве примера можно использовать проект «Delegates»).

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Определите делегат, принимающий несколько параметров различных типов и возвращающий значение произвольного типа.
3. Напишите метод, соответствующий данному делегату.
4. Напишите метод, принимающий разработанный Вами делегат, в качестве одного из входным параметров. Осуществите вызов метода, передавая в качестве параметра-делегата:
   * метод, разработанный в пункте 3;
   * лямбда-выражение.
5. Повторите пункт 4, используя вместо разработанного Вами делегата, обобщенный делегат Func< > или Action< >, соответствующий сигнатуре разработанного Вами делегата.

**Часть 2. Разработать программу, реализующую работу с рефлексией.**

(В качестве примера можно использовать проект «Reflection»).

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Создайте класс, содержащий конструкторы, свойства, методы.
3. С использованием рефлексии выведите информацию о конструкторах, свойствах, методах.
4. Создайте класс атрибута (унаследован от класса System.Attribute).
5. Назначьте атрибут некоторым свойствам классам. Выведите только те свойства, которым назначен атрибут.
6. Вызовите один из методов класса с использованием рефлексии.

**Текст Программы**

**Form1:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Лаба\_\_\_6

{

public partial class Form1 : Form

{

public delegate T TextBoxConverter<T0, T1, T2, T>(T0 p1, T1 p2, T2 str);

public delegate string connect(TextBoxConverter<int, int, string, string> p1, string p2, TextBoxConverter<int, int, string, string> p3);

public Form1()

{

InitializeComponent();

}

public static string TxTPlus(int p1, int p2, string str)

{

int p3 = p1 + p2;

return str + " " + p1.ToString() + "+" + p2.ToString() + "=" + p3.ToString();

}

public static string TxTMinus(int p1, int p2, string str)

{

int p3 = p1 - p2;

return str + " " + p1.ToString() + "-" + p2.ToString() + "=" + p3.ToString();

}

private void button1\_Click(object sender, EventArgs e)

{

string Text1 = Param2\_textBox.Text.Trim();

int Txt1 = Convert.ToInt32(Text1);

string Text2 = Param3\_textBox.Text.Trim();

int Txt2 = Convert.ToInt32(Text2);

string Text3 = Param1\_textBox.Text;

TextBoxConverter<int, int, string, string> strink= TxTPlus;

TextBoxConverter<int, int, string, string> strink1 = TxTPlus;

connect p1 = (TextBoxConverter<int, int, string, string> f1, string s, TextBoxConverter<int, int, string, string> f2) =>

{

f1 = TxTPlus;

f2 = IntPlus(f1, 2);

return f1(Txt1,Txt2,Text3) +s+ f2 (Txt1, Txt2, Text3);

};

this.listBox1.Items.Add(p1(strink, " Прибавим ещё 2 ", strink1));

}

public TextBoxConverter<int,int,string,string> IntPlus (TextBoxConverter<int,int,string,string> strink, int p3)

{

strink = (int p1, int p2, string str) =>

{

int t4 = p1 + p2 + p3;

return str + " " + p1.ToString() + "+" + p2.ToString()+ "+" + p3.ToString() + "=" + t4.ToString() ;

};

return strink;

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void Param1\_textBox\_TextChanged(object sender, EventArgs e)

{

}

private void Param2\_textBox\_TextChanged(object sender, EventArgs e)

{

}

private void Param3\_textBox\_TextChanged(object sender, EventArgs e)

{

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

Form f2 = new Form2();

f2.Show();

}

}

}

**Form2:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Reflection;

namespace Лаба\_\_\_6

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

this.ListInfo.Items.Add("Вывод информации о сборке:");

Assembly i = Assembly.GetExecutingAssembly();

this.ListInfo.Items.Add("Полное имя:" + i.FullName);

this.ListInfo.Items.Add("Исполняемый файл:" + i.Location);

this.ListInfo.Items.Add("");

this.ListInfo.Items.Add("");

Reflection.ForInspection obj = new Reflection.ForInspection();

Type File = obj.GetType();

//другой способ

//Type t = typeof(ForInspection);

this.ListInfo.Items.Add("\nИнформация о типе:");

this.ListInfo.Items.Add("Тип " + File.FullName + " унаследован от " + File.BaseType.FullName);

this.ListInfo.Items.Add("Пространство имен " + File.Namespace);

this.ListInfo.Items.Add("Находится в сборке " + File.AssemblyQualifiedName);

this.ListInfo.Items.Add("");

this.ListInfo.Items.Add("\nКонструкторы:");

foreach (var x in File.GetConstructors())

{

this.ListInfo.Items.Add(x);

}

this.ListInfo.Items.Add("");

this.ListInfo.Items.Add("\nМетоды:");

foreach (var x in File.GetMethods())

{

this.ListInfo.Items.Add(x);

}

this.ListInfo.Items.Add("");

this.ListInfo.Items.Add("\nСвойства:");

foreach (var x in File.GetProperties())

{

this.ListInfo.Items.Add(x);

}

this.ListInfo.Items.Add("");

this.ListInfo.Items.Add("\nПоля данных (public):");

foreach (var x in File.GetFields())

{

this.ListInfo.Items.Add(x);

}

this.ListInfo.Items.Add("");

this.ListInfo.Items.Add("\nForInspection реализует IComparable -> " + File.GetInterfaces().Contains(typeof(IComparable))

);

}

private void ListInfo\_SelectedIndexChanged(object sender, EventArgs e)

{

}

}

}

**Экранные формы с примерами выполнения программы**







