|  |  |
| --- | --- |
| 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)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 **Прикладная информатика**

**Отчет**

**по лабораторной работе № 7**

**Дисциплина:** Разработка приложений на языке C#

**Название работы:** Рефлексия



Студент гр. ИУ6-72Б **\_\_**01.10.2022**\_\_\_\_\_\_\_ \_**И.С. Марчук\_

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** А.М. Минитаева

(Подпись, дата) (И.О. Фамилия)

Москва, 2022

**Цель работы:** изучить основы работы с механизмами рефлексии в языке

программирования C#.

**Задание:**

Разработать программу, реализующую делегаты.

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.

Код программы:

using System;

using System.Reflection;

namespace \_6.\_2

{

class Program

{

static void Main(string[] args)

{

Type myType = typeof(ReflClass);//получаем тип

#region Информация о конструкторах

Console.WriteLine("\nИнформация о конструкторах");

ConstructorInfo[] cInfo = myType.GetConstructors();//заносим инфу в ConstructorInfo

foreach (ConstructorInfo o in cInfo)

Console.WriteLine(o);//выводим

#endregion

#region Информация о методах

Console.WriteLine("\nИнформация о методах");

MethodInfo[] cMeth = myType.GetMethods(BindingFlags.Instance | BindingFlags.Public | BindingFlags.NonPublic | BindingFlags.DeclaredOnly);//заносим инфу в MethodInfo, используя флаги, нам подходят: (Instance-Указывает, что члены экземпляров следует включить в поиск) (методы public), (NonPublic(частные)) и (только те, что находятся нашем классе)

foreach (MethodInfo o in cMeth)

Console.WriteLine(o);//выводим

#endregion

#region Информация о свойствах

Console.WriteLine("\nИнформация о свойствах");

PropertyInfo[] cProp = myType.GetProperties();//заносим инфу в PropertyInfo

foreach (PropertyInfo o in cProp)

Console.WriteLine(o);//выводим

#endregion

#region Информация только о свойствах с атрибутами

Console.WriteLine("\nИнформация только о свойствах с атрибутами");

foreach (PropertyInfo o in cProp)

{

object attributes = o.GetCustomAttribute(typeof(ReflClassAttribute), false);//извлекаем настраив. атрибут(который указали)

if (attributes != null)//если ничего не достали, значит его нет, а значит его не выводим

Console.WriteLine(o);//выводим

}

#endregion

#region Вызов метода

Console.WriteLine("\nВызов метода через рефлексию");

dynamic st = Activator.CreateInstance(myType);//создаем экз класса с помощью рефлексии

Console.WriteLine(st.Shet(true));//вызываем нужный метод

#endregion

}

}

class ReflClass

{

#region Конструкторы

public ReflClass()

{

X = 5;

Y = 6;

}

public ReflClass(int a, int b)

{

X = a;

Y = b;

}

#endregion

#region Свойства

[ReflClass]

public int X { get; set; }

public int S { get; set; }

[ReflClass]

public int Y { get; set; }

#endregion

#region Методы

public int Shet(bool f)

{

int M = f ? Sum() : Mult();

S = M;

return S;

}

private int Sum()

{

return X + Y;

}

private int Mult()

{

return X \* Y;

}

#endregion

}

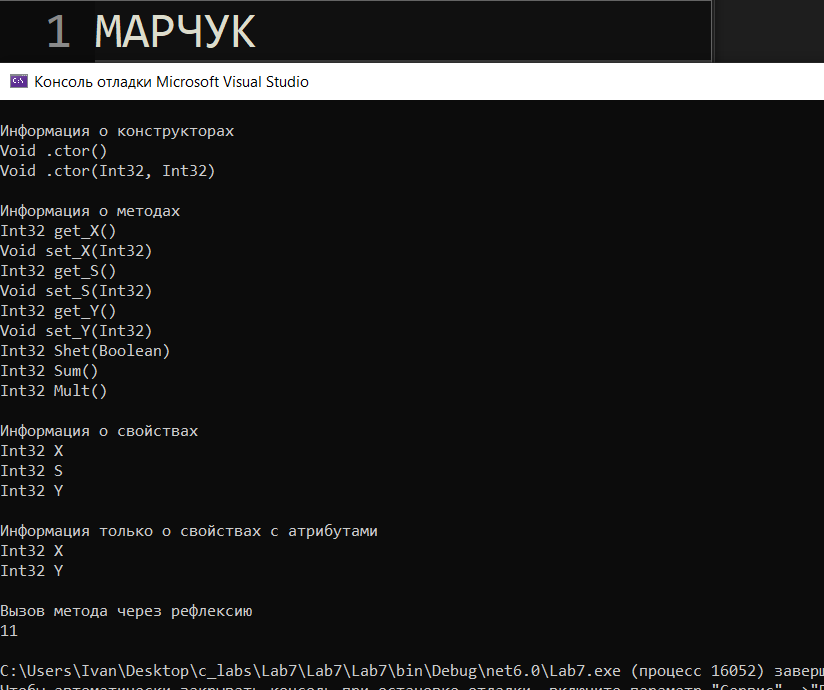
class ReflClassAttribute : System.Attribute//атрибут

{

}

}

Работа программы показана на рисунке 1.



*Рисунок 1 – Работа программы*

**Вывод:** в процессе выполнения лабораторной работы были изучены способы работы с рефлексией и типами данных. На примере класса были выведены все его методы, конструкторы и поля.