



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

О Т Ч Е Т
по лабораторной работе № 7

Дисциплина: Разработка приложений на языке C#

Название работы: Рефлексия

Студент гр. ИУ6-72Б

01.10.2022

(Подпись, дата)

И.С. Марчук

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

А.М. Минитаева

(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 в 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 настраи. 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.

1 МАРЧУК

Консоль отладки Microsoft Visual Studio

```
Информация о конструкторах
Void .ctor()
Void .ctor(Int32, Int32)

Информация о методах
Int32 get_X()
Void set_X(Int32)
Int32 get_S()
Void set_S(Int32)
Int32 get_Y()
Void set_Y(Int32)
Int32 Shet(Boolean)
Int32 Sum()
Int32 Mult()

Информация о свойствах
Int32 X
Int32 S
Int32 Y

Информация только о свойствах с атрибутами
Int32 X
Int32 Y

Вызов метода через рефлексию
11

C:\Users\Ivan\Desktop\c_labs\Lab7\Lab7\Lab7\bin\Debug\net6.0\Lab7.exe (процесс 16052) заверш
```

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.