

**Московский государственный технический
университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №6.1
«Разработка программы, реализующей работу с рефлексией»

Выполнил:
студент группы ИУ5-34Б
Сергеев Илья

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Подпись и дата:

Подпись и дата:

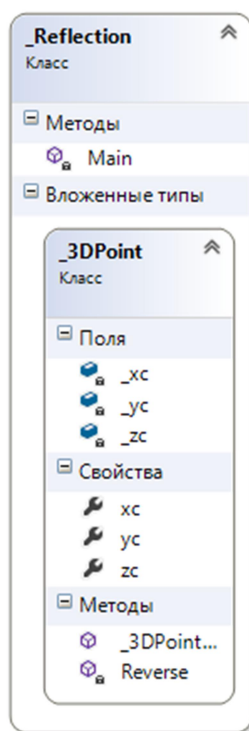
Москва, 2018 г.

Описание задания

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 классов. Выведите только те свойства, которым назначен атрибут.
6. Вызовите один из методов класса с использованием рефлексии.

Диаграмма классов

Диаграмма классов генерируется автоматически в среде Visual Studio:



Текст программы (листинг)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace reflection
{
    class _Reflection
    {
        class _3DPoint
        {
            //x-coordinate
            private double _xc;
            public double xc
            {
                get { return _xc; }
                set { _xc = value; }
            }
        }
    }
}
```

```

    }

    //y-coordinate
    private double _yc;
    public double yc
    {
        get { return _yc; }
        set { _yc = value; }
    }

    //z-coordinate
    private double _zc;
    public double zc
    {
        get { return _zc; }
        set { _zc = value; }
    }

    /// <summary>
    /// Default constructor
    /// </summary>
    public _3DPoint()
    {
        xc = 0;
        yc = 0;
        zc = 0;
    }

    /// <summary>
    /// Constructor for a point with equal coords "a"
    /// </summary>
    /// <param name="a"></param>
    public _3DPoint(double a)
    {
        xc = a;
        yc = a;
        zc = a;
    }

    /// <summary>
    /// Constructor for a point with different coords
    /// </summary>
    /// <param name="x"></param>
    /// <param name="y"></param>
    /// <param name="z"></param>
    public _3DPoint(double x, double y, double z)
    {
        xc = x;
        yc = y;
        zc = z;
    }

    /// <summary>
    /// Method of reversing point coords
    /// </summary>
    void Reverse()
    {
        xc = -xc;
        yc = -yc;
        zc = -zc;
    }
}

static void Main(string[] args)
{

```

```

        _3DPoint obj = new _3DPoint();
        Type t = obj.GetType();
        Console.WriteLine("\nИнформация о типе:");
        Console.WriteLine("Тип " + t.FullName + " унаследован от " +
t.BaseType.FullName);
        Console.WriteLine("Пространство имен " + t.Namespace);
        Console.WriteLine("Находится в сборке " + t.AssemblyQualifiedName);
        Console.WriteLine("\nКонструкторы:");
        foreach (var x in t.GetConstructors())
            Console.WriteLine(x);
        Console.WriteLine("\nМетоды:");
        foreach (var x in t.GetMethods())
            Console.WriteLine(x);
        Console.WriteLine("\nСвойства:");
        foreach (var x in t.GetProperties())
            Console.WriteLine(x);
        Console.WriteLine("\nПоля данных (public):");
        foreach (var x in t.GetFields())
            Console.WriteLine(x);
        Console.WriteLine("\nForInspection реализует IComparable -> " +
t.GetInterfaces().Contains(typeof(IComparable)) );

        Console.ReadKey();
    }
}
}

```

Экранные формы с примерами выполнения программы (скриншоты)

```

file:///F:/GitHub/Work6.1/reflection/bin/Debug/reflection.EXE

Информация о типе:
Тип reflection._Reflection+_3DPoint унаследован от System.Object
Пространство имен reflection
Находится в сборке reflection._Reflection+_3DPoint, reflection, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null

Конструкторы:
Void .ctor()
Void .ctor(Double)
Void .ctor(Double, Double, Double)

Методы:
Double get_xc()
Void set_xc(Double)
Double get_yc()
Void set_yc(Double)
Double get_zc()
Void set_zc(Double)
System.String ToString()
Boolean Equals(System.Object)
Int32 GetHashCode()
System.Type GetType()

Свойства:
Double xc
Double yc
Double zc

Поля данных (public):
ForInspection реализует IComparable -> False

```