

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

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №6

Выполнил:

студент группы ИУ5-31

Стрихар Павел

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю. Е.

Подпись и дата:

Москва, 2020 г.

Описание задания:

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.

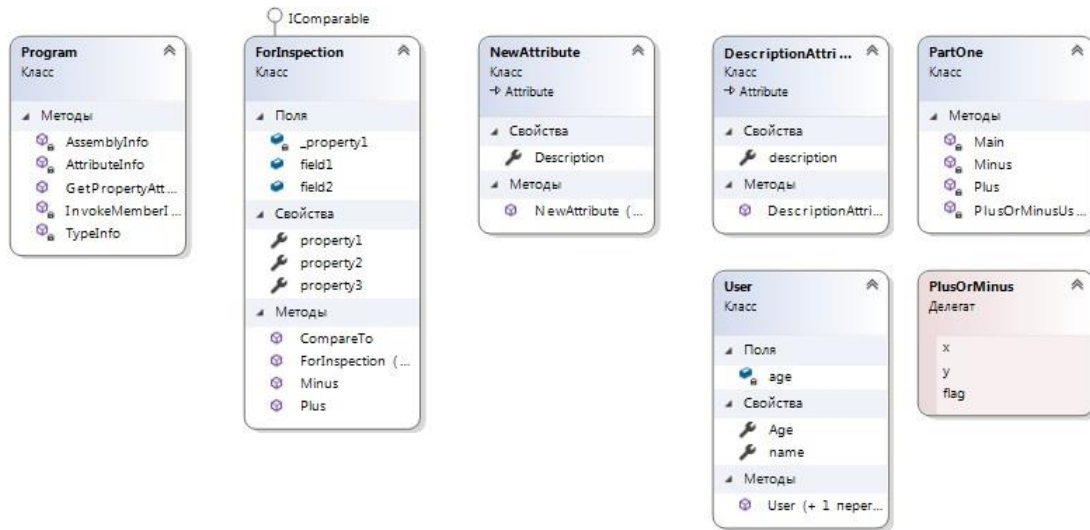
Часть 2. Разработать программу, реализующую работу с рефлексией.

(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.

Часть 1

Диаграмма классов:



Код программы:

Program.cs

```
using System;
using System.Collections.Generic;
using System.Net.Cache;
using System.Runtime.CompilerServices;
using System.Security.Cryptography.X509Certificates;
using System.Reflection;

namespace Lab6
{
    delegate object PlusOrMinus(int x, double y, bool flag);
    class PartOne
    {
        static object Plus(int x, double y, bool flag)
        {
            if (flag == true)
                return ((int)(x + y));
            else
                return (x + y);
        }
        static object Minus(int x, double y, bool flag)
        {
            if (flag == true)
                return ((int)(x - y));
            else
                return (x - y);
        }
        static void PlusOrMinusUse1(string str, int x, double y, bool flag, Func<int, double, bool, object> PlusOrMinusParam)
        {
            object result = PlusOrMinusParam(x, y, flag);
            Console.WriteLine(str + " " + result.ToString());
        }

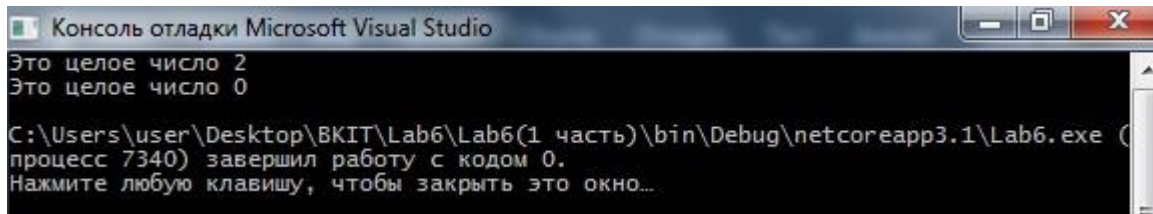
        static void Main(string[] args)
        {
            PlusOrMinus pm = new PlusOrMinus(Plus);
            PlusOrMinusUse1("Это целое число", 1, 1.5, true, Plus);
            PlusOrMinusUse1("Это целое число", 1, 1.5, true, (x, y, flag) =>
            {
                if (flag == true)
                    return ((int)(x - y));
                else
                    return (x - y);
            });
        }
    }

    class DescriptionAttribute : Attribute
    {
        public string description { get; set; }
        public DescriptionAttribute() { }
        public DescriptionAttribute(string _description) { description = _description; }
    }

    [Description(description = "18")]
    class User
    {
        public User() { }
        public User(string _name, int _age) { name = _name; age = _age; }
        public string name { get; set; }
        private int age;
        [Description("Возраст")]
    }
}
```

```
    public int Age
    {
        get
        {
            return age;
        }
        set
        {
            age = value;
        }
    }
}
}
```


Пример выполнения программы:

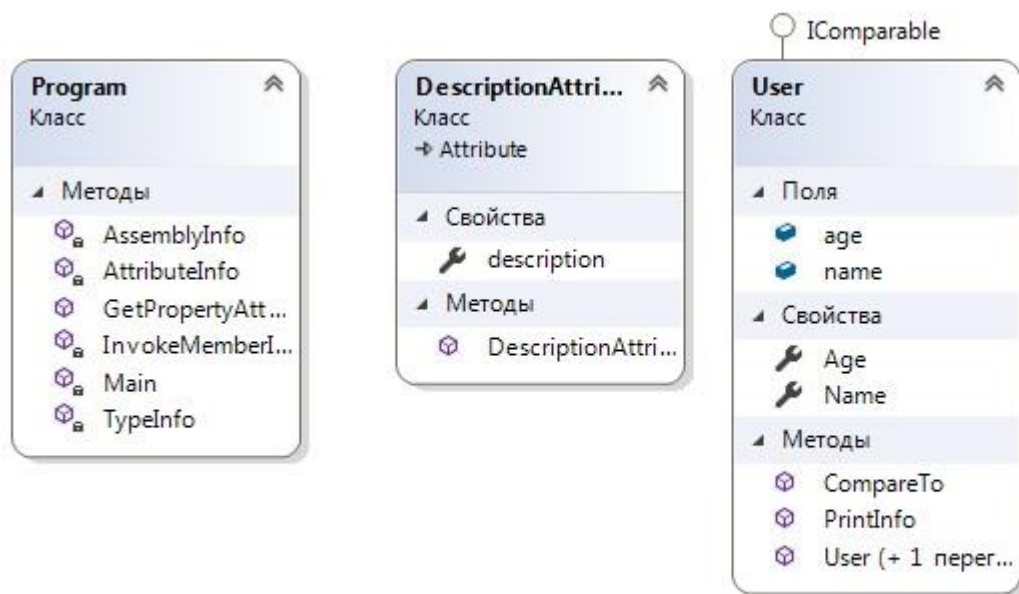


Консоль отладки Microsoft Visual Studio

```
Это целое число 2
Это целое число 0

C:\Users\user\Desktop\BKIT\Lab6\Lab6(1 часть)\bin\Debug\netcoreapp3.1\Lab6.exe (
процесс 7340) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...
```

Часть 2 Диаграмма классов:



Код программы:

Program.cs

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

namespace Lab6Part2
{
    class Program
    {
        /// <summary>
        /// Проверка, что у свойства есть атрибут заданного типа
        /// </summary>
        /// <returns>Значение атрибута</returns>
        public static bool GetPropertyAttribute(PropertyInfo checkType, Type attributeType,
        out object attribute)
        {
            bool Result = false;
```

```

        attribute = null;

        //Поиск атрибутов с заданным типом
        var isAttribute = checkType.GetCustomAttributes(attributeType, false);
        if (isAttribute.Length > 0)
        {
            Result = true;
            attribute = isAttribute[0];
        }

        return Result;
    }
    /// <summary>
    /// Получение информации о текущей сборке
    /// </summary>
    static void AssemblyInfo()
    {
        Console.WriteLine("Вывод информации о сборке:");
        Assembly i = Assembly.GetExecutingAssembly();
        Console.WriteLine("Полное имя:" + i.FullName);
        Console.WriteLine("Исполняемый файл:" + i.Location);
    }
    /// <summary>
    /// Получение информации о типе
    /// </summary>
    static void TypeInfo()
    {
        Type t = typeof(User);

        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))
);
    }

    /// <summary>
    /// Использование метода InvokeMember
    /// </summary>
    static void InvokeMemberInfo()
    {

```



```

        Type t = typeof(User);
        Console.WriteLine("\nВызов метода:");

        //Создание объекта
        User fi = new User("Pavel", 18);
        //User fi = (User)t.InvokeMember(null, BindingFlags.CreateInstance, null, null,
new object[] { });

        //Вызов метода
        t.InvokeMember("PrintInfo", BindingFlags.InvokeMethod, null, fi, new object[] {
});
    }

    /// <summary>
    /// Работа с атрибутами
    /// </summary>
    static void AttributeInfo()
    {
        Type t = typeof(User);
        Console.WriteLine("\nСвойства, помеченные атрибутом:");
        foreach (var x in t.GetProperties())
        {
            object attrObj;
            if (GetPropertyAttribute(x, typeof(DescriptionAttribute), out attrObj))
            {
                DescriptionAttribute attr = attrObj as DescriptionAttribute;
                Console.WriteLine(x.Name + " - " + attr.description);
            }
        }
    }
    static void Main(string[] args)
    {
        AssemblyInfo();
        TypeInfo();
        InvokeMemberInfo();
        AttributeInfo();
    }
}

```

Пример выполнения программы:

```
Консоль отладки Microsoft Visual Studio
Вывод информации о сборке:
Полное имя:Lab6Part2, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null
Исполняемый файл:C:\Users\user\Desktop\BKIT\Lab6\Lab6(2 часть)\bin\Debug\netcoreapp3.1\Lab6Part2.dll

Информация о типе:
Тип Lab6Part2.User унаследован от System.Object
Пространство имен Lab6Part2
Находится в сборке Lab6Part2.User, Lab6Part2, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null

Конструкторы:
Void .ctor()
Void .ctor(System.String, Int32)

Методы:
System.String get_Name()
Void set_Name(System.String)
Int32 get_Age()
Void set_Age(Int32)
Void PrintInfo()
Int32 CompareTo(System.Object)
System.Type GetType()
System.String ToString()
Boolean Equals(System.Object)
Int32 GetHashCode()

Свойства:
System.String Name
Int32 Age

Поля данных (public):
System.String name
Int32 age

ForInspection реализует IComparable -> True

Вызов метода:
Имя пользователя: Pavel
Возраст пользователя: 18

Свойства, помеченные атрибутом:
Age - Возраст

C:\Users\user\Desktop\BKIT\Lab6\Lab6(2 часть)\bin\Debug\netcoreapp3.1\Lab6Part2.exe (процесс 9088) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...
```

