

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

Кафедра
“Системы обработки информации и управления”
(ИУ – 5)

**Лабораторная работа по №6 по дисциплине «Базовые компоненты
интернет-технологий»**

Выполнил:
студент гр. ИУ5 - 31
Саадиев АС

“27” декабря 2017 г.

Москва – 2017 г.

Задание:

Часть 1. Разработать программу, использующую делегаты.

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

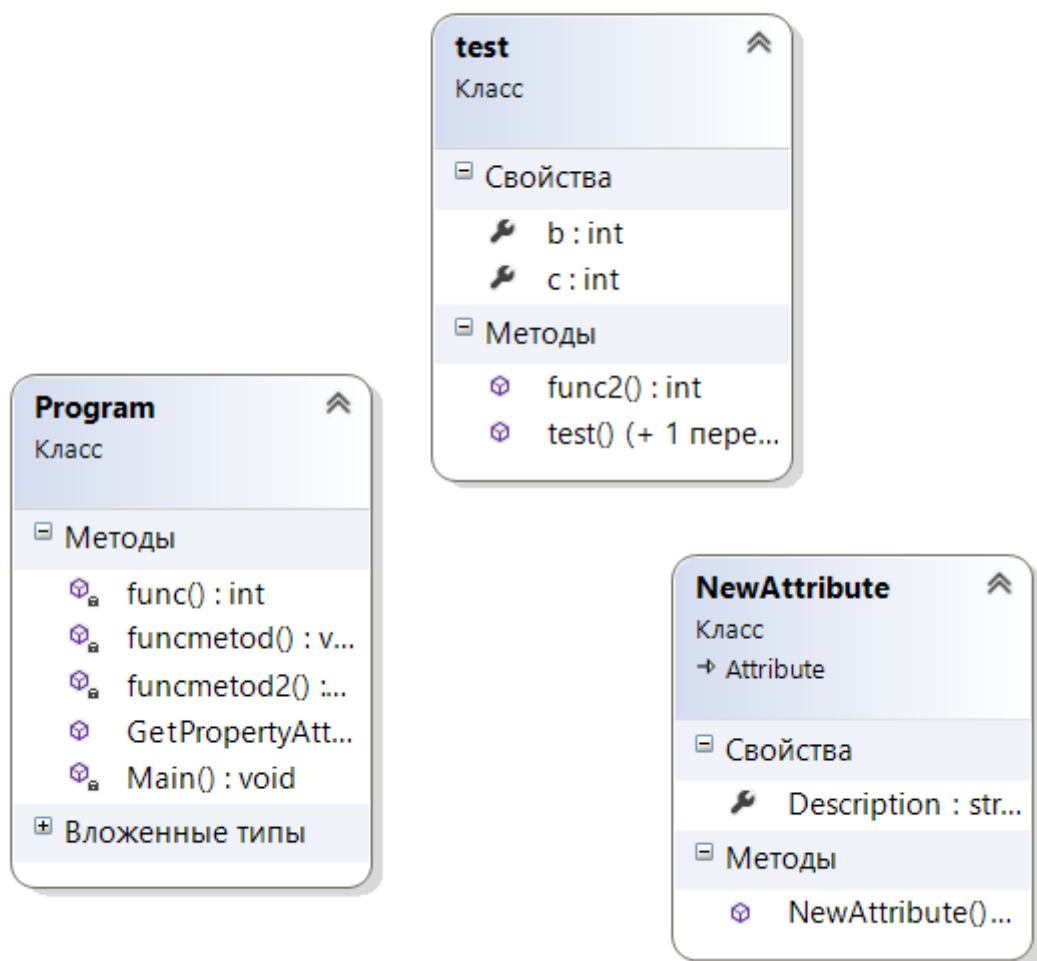
1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Определите делегат, принимающий несколько параметров различных типов и возвращающий значение произвольного типа.
3. Напишите метод, соответствующий данному делегату.
4. Напишите метод, принимающий разработанный Вами делегат, 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.

Диаграмма классов:



Код:

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

namespace Lab6
{
    class Program
    {
        delegate int func(int a, int b);

        static int mult(int a, int b)
        { return a * b; }

        static int div(int a, int b)
        { return a / b; }

        static void chooseMethod(int a, int b, Func<int, int, int> func)
        {
```

```

        Console.WriteLine("Result: " + func(a, b));
    }

    public class Gtribute : Attribute
    {
        public string desc { get; set; }
        public Gtribute() { }
        public Gtribute(string d)
        {
            desc = d;
        }
    }

    class Geek
    {
        [Gtribute("Usually equals 0")]
        public int power;
        public int intelligence;
        public int stamina;

        public Geek()
        {
            power = 0;
            intelligence = 0;
            stamina = 0;
        }

        public Geek(int p, int i, int s)
        {
            power = p;
            intelligence = i;
            stamina = s;
        }

        public void Roar()
        {
            Console.WriteLine("Roooooooooar");
        }

        [Gtribute("Very common")]
        public void Panic()
        {
            Console.WriteLine("OMG!!!");
        }
    }

    static void Main(string[] args)
    {
        int a = 10, b = 5;

        chooseMethod(a, b, mult);
        chooseMethod(a, b, div);

        chooseMethod(a, b, (int x, int y) => { return x * y; });

        Action<int, int> ac = (x, y) => { Console.WriteLine("{0} + {1} = {2}", x, y,
x + y); };
        ac(7, 3);

        Type t = typeof(Geek);

        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("\nПоля данных (public):");
    foreach (var x in t.GetFields())
    {
        Console.WriteLine(x);
    }

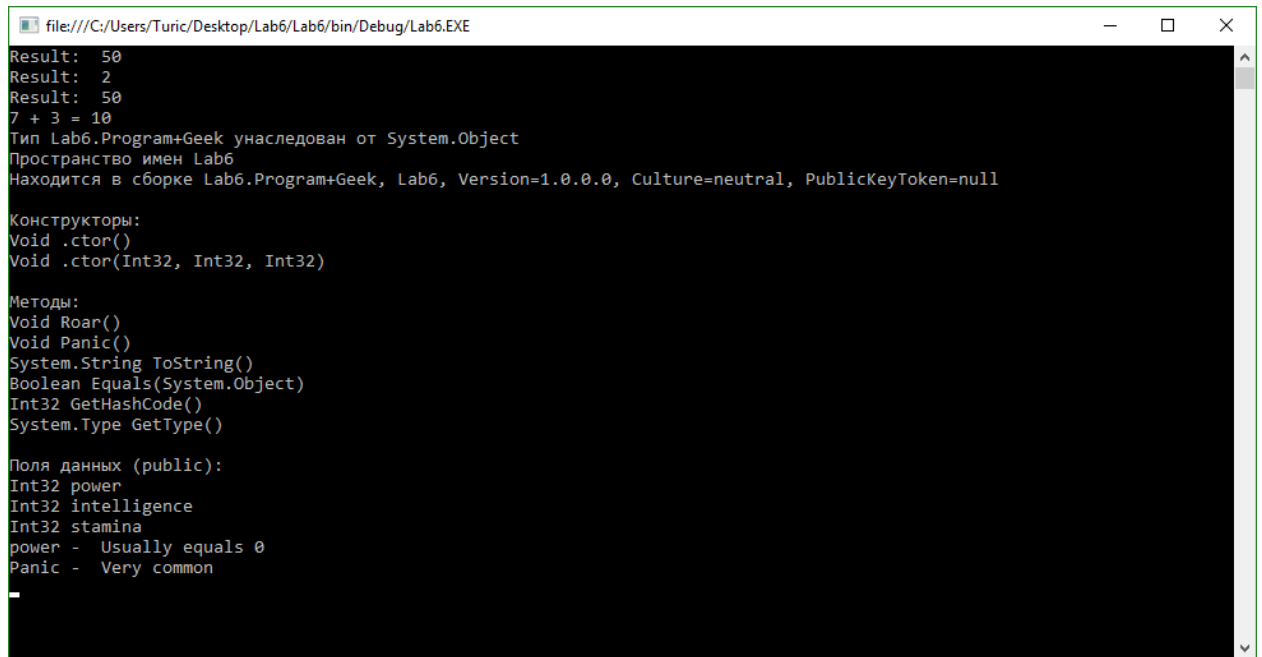
    foreach (var x in t.GetFields())
    {
        var at = x.GetCustomAttributes(typeof(Gtribute), false);
        if (at.Length > 0)
        {
            Gtribute g = at[0] as Gtribute;
            Console.WriteLine(x.Name + " - " + g.desc);
        }
    }

    foreach (var x in t.GetMethods())
    {
        var at = x.GetCustomAttributes(typeof(Gtribute), false);
        if (at.Length > 0)
        {
            Gtribute g = at[0] as Gtribute;
            Console.WriteLine(x.Name + " - " + g.desc);
        }
    }

    Console.ReadKey();
}
}
}

```

Скриншот:



```
file:///C:/Users/Turic/Desktop/Lab6/Lab6/bin/Debug/Lab6.EXE
Result: 50
Result: 2
Result: 50
7 + 3 = 10
Тип Lab6.Program+Geek унаследован от System.Object
Пространство имен Lab6
Находится в сборке Lab6.Program+Geek, Lab6, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null

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

Методы:
Void Roar()
Void Panic()
System.String ToString()
Boolean Equals(System.Object)
Int32 GetHashCode()
System.Type GetType()

Поля данных (public):
Int32 power
Int32 intelligence
Int32 stamina
power - Usually equals 0
Panic - Very common
-
```