2018

Отчет по лабораторной работе № 6

Базовые компоненты интернет технологий

Зубков андрей, иу5-34б

**Часть 1 (Делегаты)**

Описание задания

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 входных параметров. Осуществите вызов метода, передавая в качестве параметра-делегата:

* метод, разработанный в пункте 3;
* лямбда-выражение.

1. Повторите пункт 4, используя вместо разработанного Вами делегата, обобщенный делегат Func< > или Action< >, соответствующий сигнатуре разработанного Вами делегата.

Текст программы

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab6

{

class Program

{

delegate double myDelegate(int x, int y);

static double lengthToCoordinatesBegin(int x, int y)

{

return Math.Sqrt(x \* x + y \* y);

}

static void printLength(int x, int y, myDelegate myDelegateParam)

{

Console.WriteLine("Length = {0}", myDelegateParam(x, y));

}

static void printLengthFunc(int x, int y, Func<int, int, double> myFuncParam)

{

Console.WriteLine("Length = {0}", myFuncParam(x, y));

}

static void Main(string[] args)

{

int x = 4;

int y = 7;

myDelegate func = new myDelegate(lengthToCoordinatesBegin);

printLength(x, y, func);

printLength(x, y, (int first, int second) =>

{

return (first \* first + second \* second) / 2;

}

);

printLengthFunc(x, y, (int first, int second) =>

{

return (first + second) / 3.0;

}

);

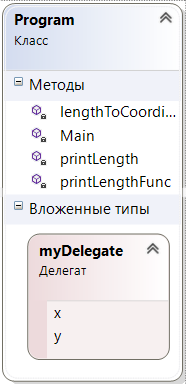
Console.ReadKey();

}

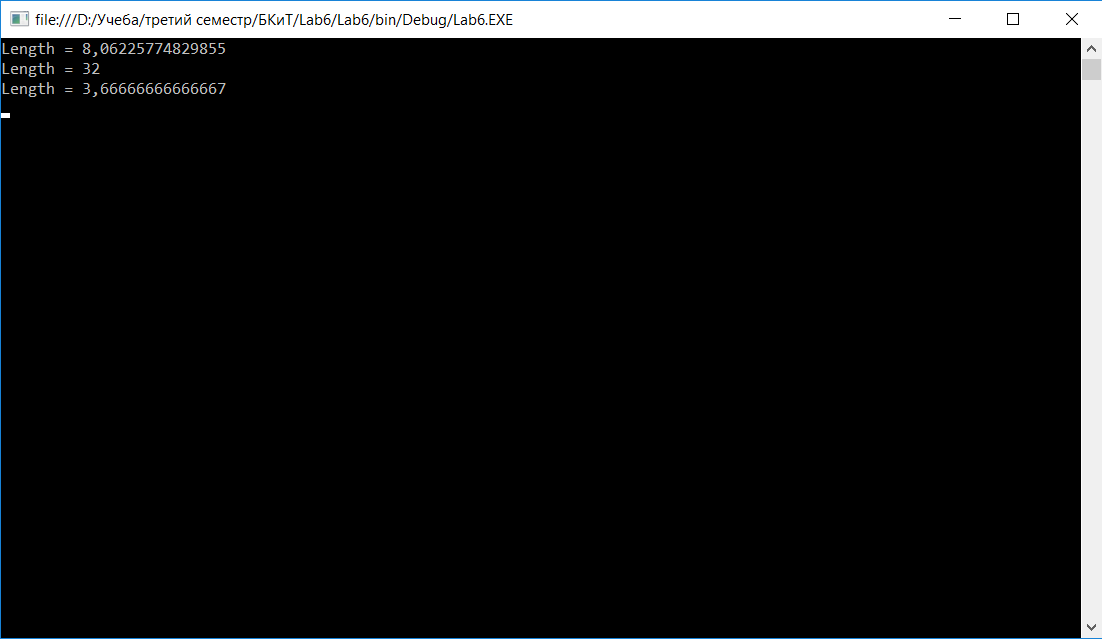
}

}

Диаграмма классов



Пример работы программы



**Часть 2 (Рефлексия)**

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

Текст программы

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Reflection;

namespace Lab6.\_2

{

class myClass

{

public myClass(int param)

{

param\_ = param;

}

private int param\_;

public int Name

{

get

{

return param\_;

}

set

{

param\_ = value;

}

}

public int multiplication()

{

return param\_ \* param\_;

}

};

class Program

{

static void Main(string[] args)

{

myClass tmp = new myClass(3);

Type t = tmp.GetType();

Console.WriteLine(“Constructors:”);

foreach (var x in t.GetConstructors())

{

Console.WriteLine(x);

}

Console.WriteLine(“\nMethods:”);

foreach (var x in t.GetMethods())

{

Console.WriteLine(x);

}

Console.WriteLine(“\nPropeties:”);

foreach (var x in t.GetProperties())

{

Console.WriteLine(x);

}

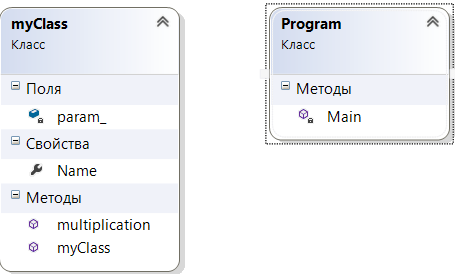
Console.ReadKey();

}

}

}

Диаграмма классов



Пример работы программы

