МГТУ им. Баумана  
Факультет «Информатика и системы управления»  
Кафедра «Автоматизированные системы обработки информации и управления»  
Дисциплина «Базовые компоненты интернет-технологий»

Отчёт по лабораторной работе №6



Описание задания:

Часть 1. Разработать программу, использующую делегаты.

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.

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.**

Текст программы:

Класс Program:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Delegates {

public delegate double Calc(int p1, int p2, double p3);

class Program {

public static double PlusMinus(int p1, int p2, double p3) {

return p1 + p2 - p3;

}

public static double MinusMinus(int p1, int p2, double p3) {

return p1 - p2 - p3;

}

public static double PlusDivide(int p1, int p2, double p3) {

return (p1 + p2) / p3;

}

public static void CalcMethod(int p1, int p2, double p3, Calc c) {

double result = c(p1, p2, p3);

Console.WriteLine(result);

}

public static void CalcFuncMethod(int p1, int p2, double p3, Func<int, int, double, double> f) {

double result = f(p1, p2, p3);

Console.WriteLine(result);

}

static void Main(string[] args) {

int i1 = 7, i2 = 3;

double i3 = 4;

Console.WriteLine("Вызовы методов, принимающих обыкновенный делегат.");

CalcMethod(i1, i2, i3, PlusMinus);

CalcMethod(i1, i2, i3, MinusMinus);

CalcMethod(i1, i2, i3, PlusDivide);

CalcMethod(i1, i2, i3, (int x, int y, double z) => {

double res = (x \* y) / z;

return res;

});

Console.WriteLine("Вызовы методов, принимающих обобщенный делегат типа Func.");

CalcFuncMethod(i1, i2, i3, PlusMinus);

CalcFuncMethod(i1, i2, i3, MinusMinus);

CalcFuncMethod(i1, i2, i3, PlusDivide);

CalcFuncMethod(i1, i2, i3, (int x, int y, double z) => {

double res = (x \* y) / z;

return res;

});

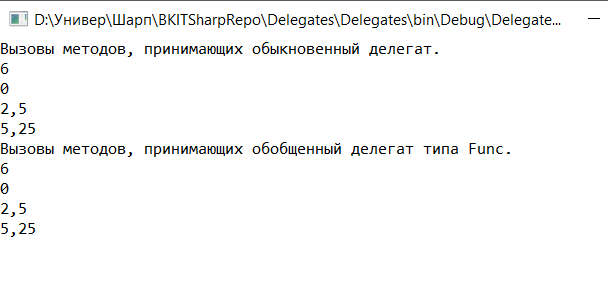
Console.ReadLine();

}

}

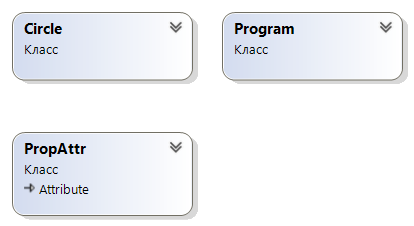
}

Экранные формы:



**Часть 2.**

Диаграмма классов:



Текст программы:

Класс Circle:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Reflection {

public class Circle {

public double Radius = 0;

[PropAttr("Диаметр")]

public double Diameter { get { return 2 \* Radius; } }

[PropAttr("Длина окружности")]

public double Circumference { get { return Diameter \* Math.PI; } }

public double Area { get { return Radius \* Radius \* Math.PI; } }

public override string ToString() {

return $"Радиус: {Radius}.\nДлина окружности: {Circumference}.\nПлощадь: {Area}.";

}

public Circle(double r) {

Radius = r;

}

}

}

Класс PropAttr:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Reflection {

public class PropAttr : Attribute {

public static string Description;

public PropAttr(string s) {

Description = s;

}

}

}

Класс Program:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Reflection {

class Program {

static void Main(string[] args) {

Circle obj = new Circle(2.4);

Type t = obj.GetType();

Console.WriteLine("Рефлексия.\n\nИнформация о типе: ");

Console.WriteLine($"Пространство имён {t.Namespace}\nИмя: {t.FullName}\nСборка: {t.AssemblyQualifiedName}");

Console.WriteLine("\nКонструкторы:");

foreach (var x in t.GetConstructors()) {

Console.WriteLine(x);

}

Console.WriteLine("\nПоля:");

foreach (var x in t.GetFields()) {

Console.WriteLine(x);

}

Console.WriteLine("\nСвойства:");

foreach (var x in t.GetProperties()) {

Console.WriteLine(x);

}

Console.WriteLine("\nМетоды:");

foreach (var x in t.GetMethods()) {

Console.WriteLine(x);

}

Console.WriteLine("\nСвойства с атрибутами:");

foreach (var x in t.GetProperties()) {

if (x.GetCustomAttributes(typeof(PropAttr), false).Length > 0) {

Console.WriteLine(x + " " + PropAttr.Description);

}

}

string method = "ToString";

Console.WriteLine($"\nВызов метода {method} с помощью рефлексии:");

object result = t.InvokeMember(method, System.Reflection.BindingFlags.InvokeMethod, null, obj, null);

Console.WriteLine(result);

Console.ReadLine();

}

}

}

Экранные формы:

