**Vector.cs (в 11)**

**Reflector.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Reflection;

using System.IO;

namespace ConsoleApp1

{

class reflector

{

public void GetInfo(string name)

{

Type type = Type.GetType(name, false, true);

StreamWriter sw = new StreamWriter("D:\\ООП\\Lab12\\Laba12.txt");

foreach (MemberInfo mi in type.GetMembers())

{

sw.WriteLine(mi.DeclaringType + " " + mi.MemberType + " " + mi.Name);

}

sw.Close();

}

public void PublicMethods(string name)

{

Type type = Type.GetType(name, false, true);

Console.WriteLine("Методы");

foreach (MethodInfo mi in type.GetMethods())

{

if (mi.IsPublic)

{

Console.WriteLine(mi.Name + " " + mi.ReturnType.Name);

}

}

}

public void PropertyInfo(string name)

{

Type type = Type.GetType(name, false, true);

Console.WriteLine("Поля");

foreach (FieldInfo mi in type.GetFields())

{

Console.WriteLine(mi.FieldType + " " + mi.Name);

}

Console.WriteLine("Свойства");

foreach(PropertyInfo prop in type.GetProperties())

{

Console.WriteLine(prop.Name + " " + prop.PropertyType);

}

}

public void Interfaces(string name)

{

Type type = Type.GetType(name, false, true);

Console.WriteLine("Реализованные интерфейсы");

foreach (Type mi in type.GetInterfaces())

{

Console.WriteLine(mi.Name);

}

}

public void GetMethodsByParamets(string name,Type param)

{

Type type = Type.GetType(name, false, true);

Console.WriteLine("Методы с заданным параметром");

foreach (MethodInfo mi in type.GetMethods())

{

if(mi.ReturnType==param)

{

Console.WriteLine(mi.Name+" "+mi.ReturnType);

}

}

}

public void CallMethods(string path)

{

StreamReader sr = new StreamReader(path);

string str=sr.ReadLine();

Assembly asm = Assembly.LoadFrom(str);

Type t = asm.GetType("ConsoleApp1.Program", true, true);

// создаем экземпляр класса Program

object obj = Activator.CreateInstance(t);

// получаем метод GetResult

MethodInfo method = t.GetMethod("Main", BindingFlags.DeclaredOnly

| BindingFlags.Instance | BindingFlags.NonPublic | BindingFlags.Static);

method.Invoke(obj, new object[] { new string[] { } });

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Reflection;

using System.IO;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Vector vect1 = new Vector(new int[] { 2, 3, 4, 14, 45, 2, 12, });

reflector refl = new reflector();

refl.GetInfo("ConsoleApp1.Vector");

refl.PublicMethods("ConsoleApp1.Vector");

refl.PropertyInfo("ConsoleApp1.Vector");

refl.Interfaces("ConsoleApp1.Vector");

refl.GetMethodsByParamets("ConsoleApp1.Vector", typeof(int));

refl.CallMethods("D:\\ООП\\Lab12\\ConsoleApp1\\ConsoleApp1\\bin\\Debug\\Laba12call.txt");

}

}

}