



Тренинг «Программирование Web приложений на ASP.NET MVC»

Саркисян Гаяне Феликсовна

Примерный перечень тем

1. Введение в C# и .NET Framework
2. Основные программные конструкции C#
3. Методы. Введение в unit-тестирование
4. Инкапсуляция. Наследование. Полиморфизм.
5. Обработка исключений
6. Принципы проектирования. Наследование vs Композиция vs Агрегация. Связный код. Шаблоны проектирования
7. Работа с файловой системой. Потоки и ввод-вывод данных
8. Управление ресурсами в .NET. Модель памяти
9. Делегаты и события
10. Обобщения и коллекции
11. Введение в LINQ
12. Введение в многопоточное программирование
13. XML и основные технологии .NET Framework XML. Сериализация.
14. Базы данных, SQL и ADO.NET. ADO.NET EF.
15. Введение в web
16. ASP.NET MVC

Инструментарий



Google Disk



Github



LINQPad



.NET Reflector



JetBrains dotPeek



NuGet

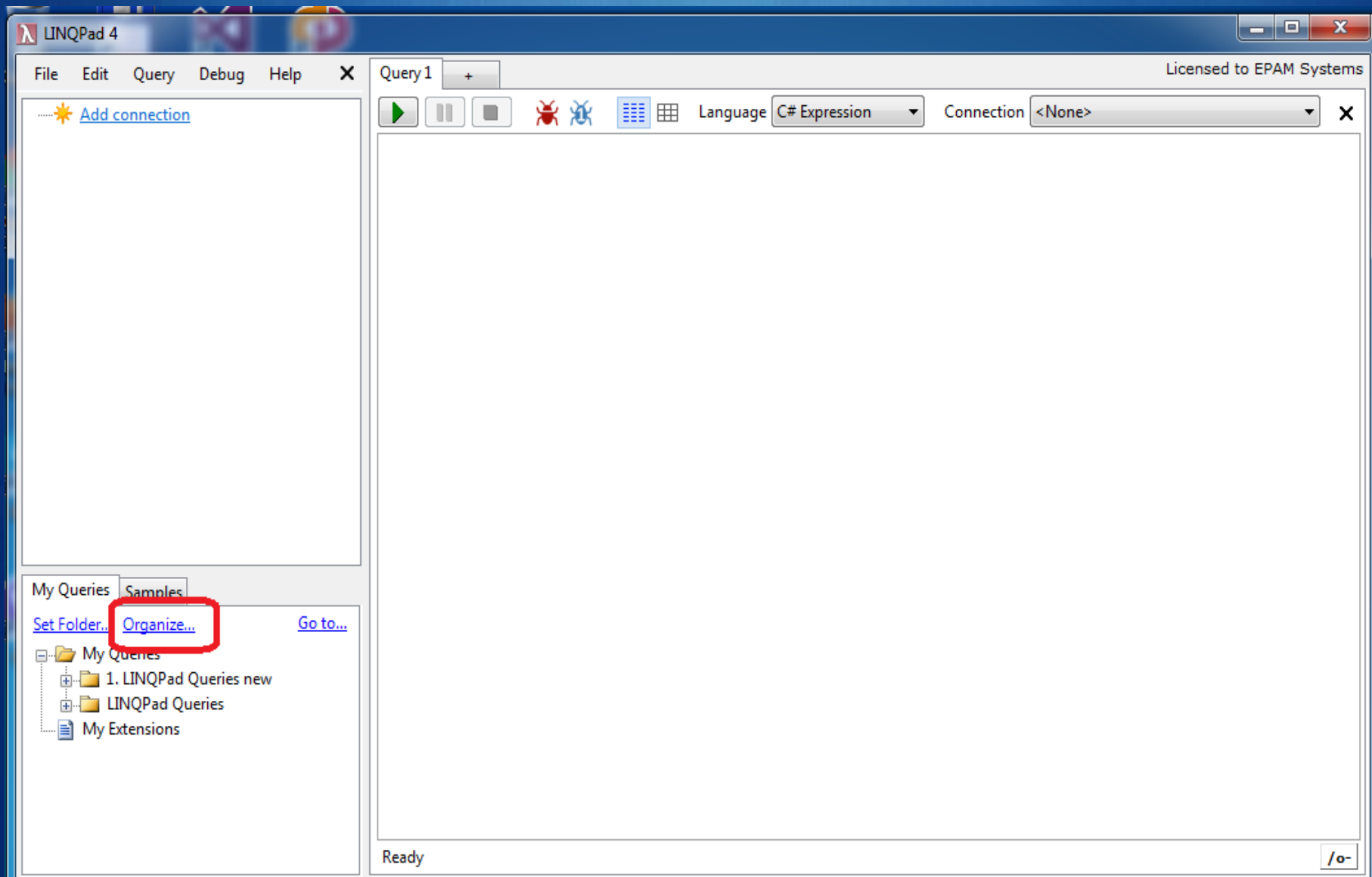


Fiddler

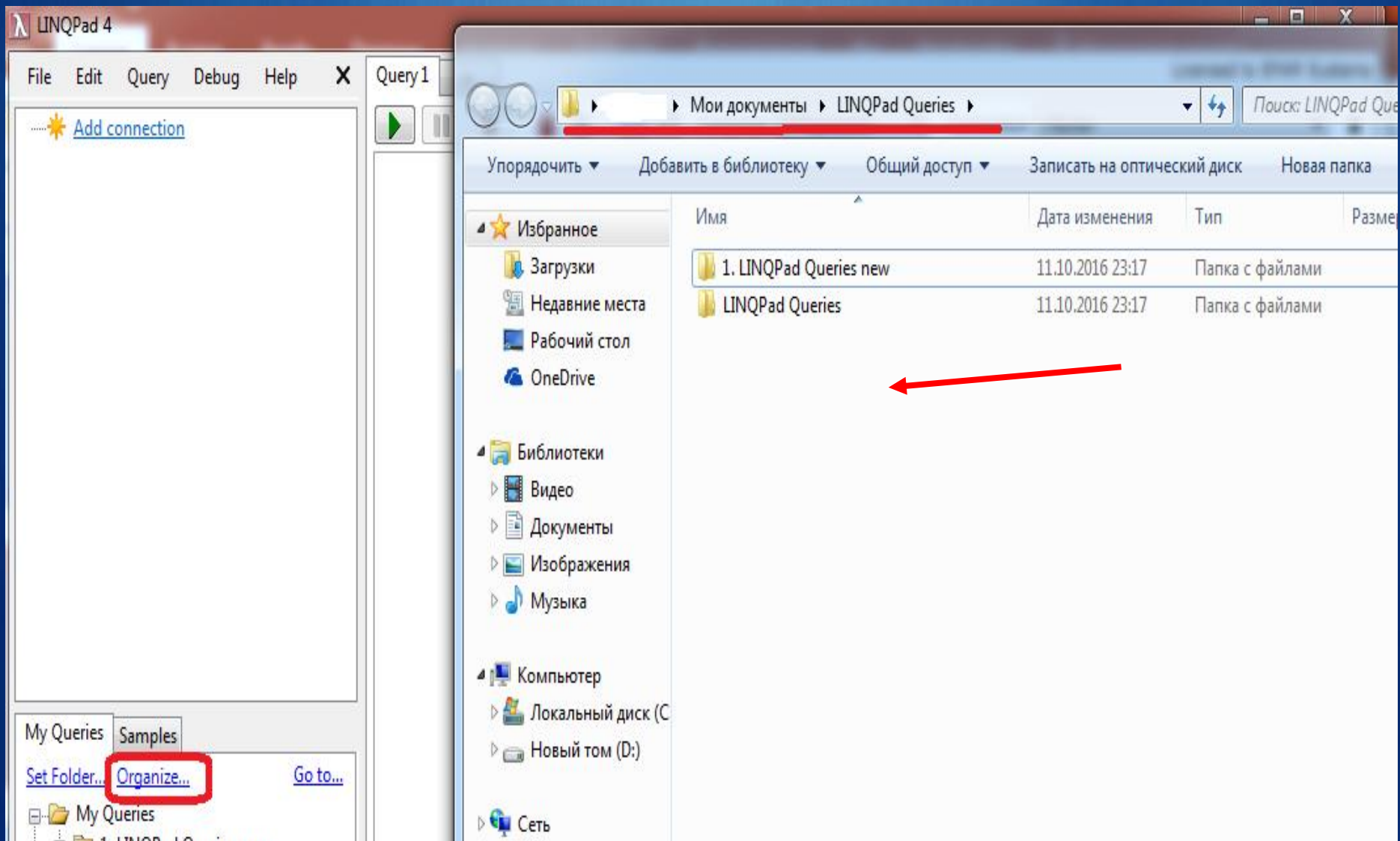


Example

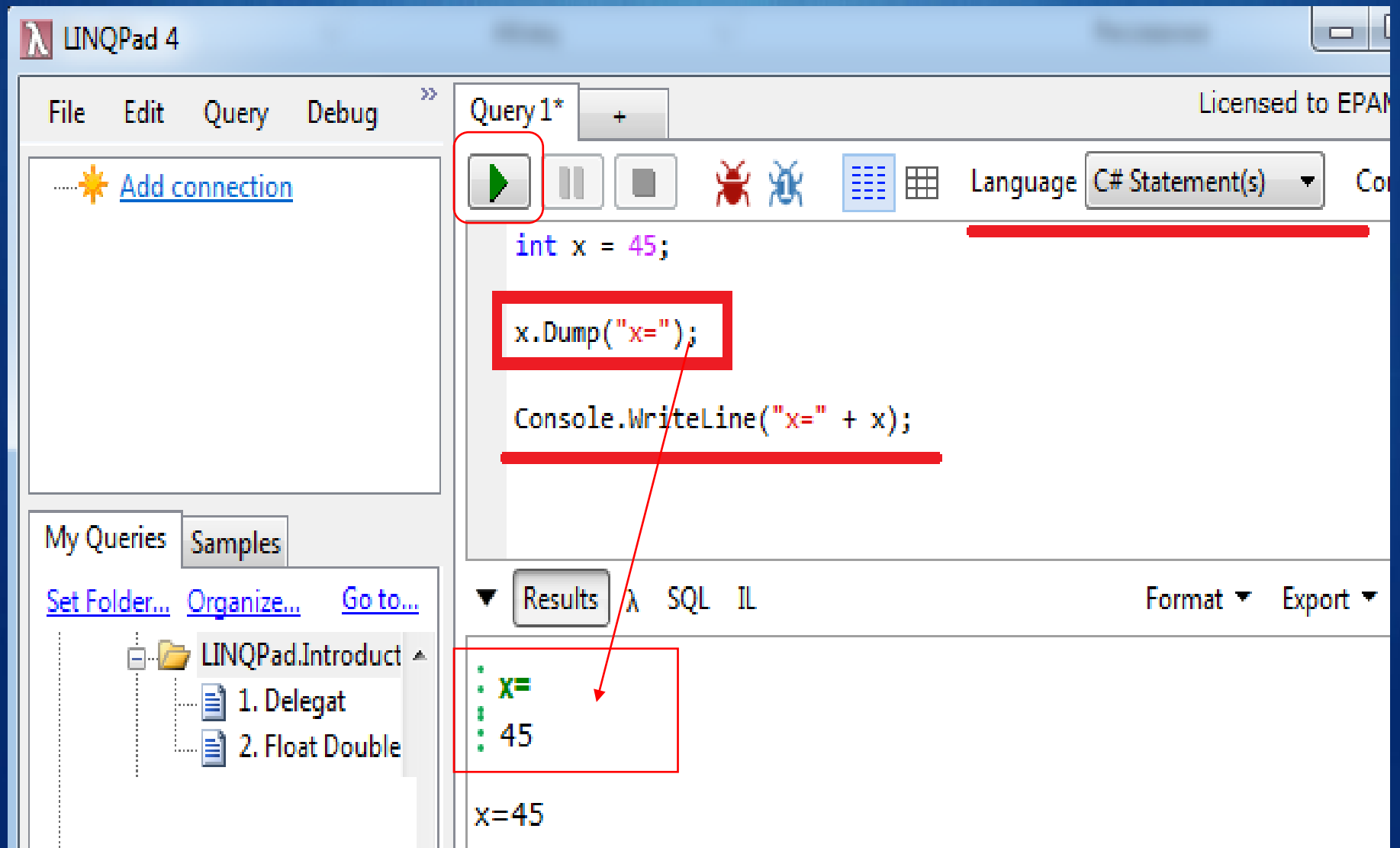
Инструментарий LINQPad



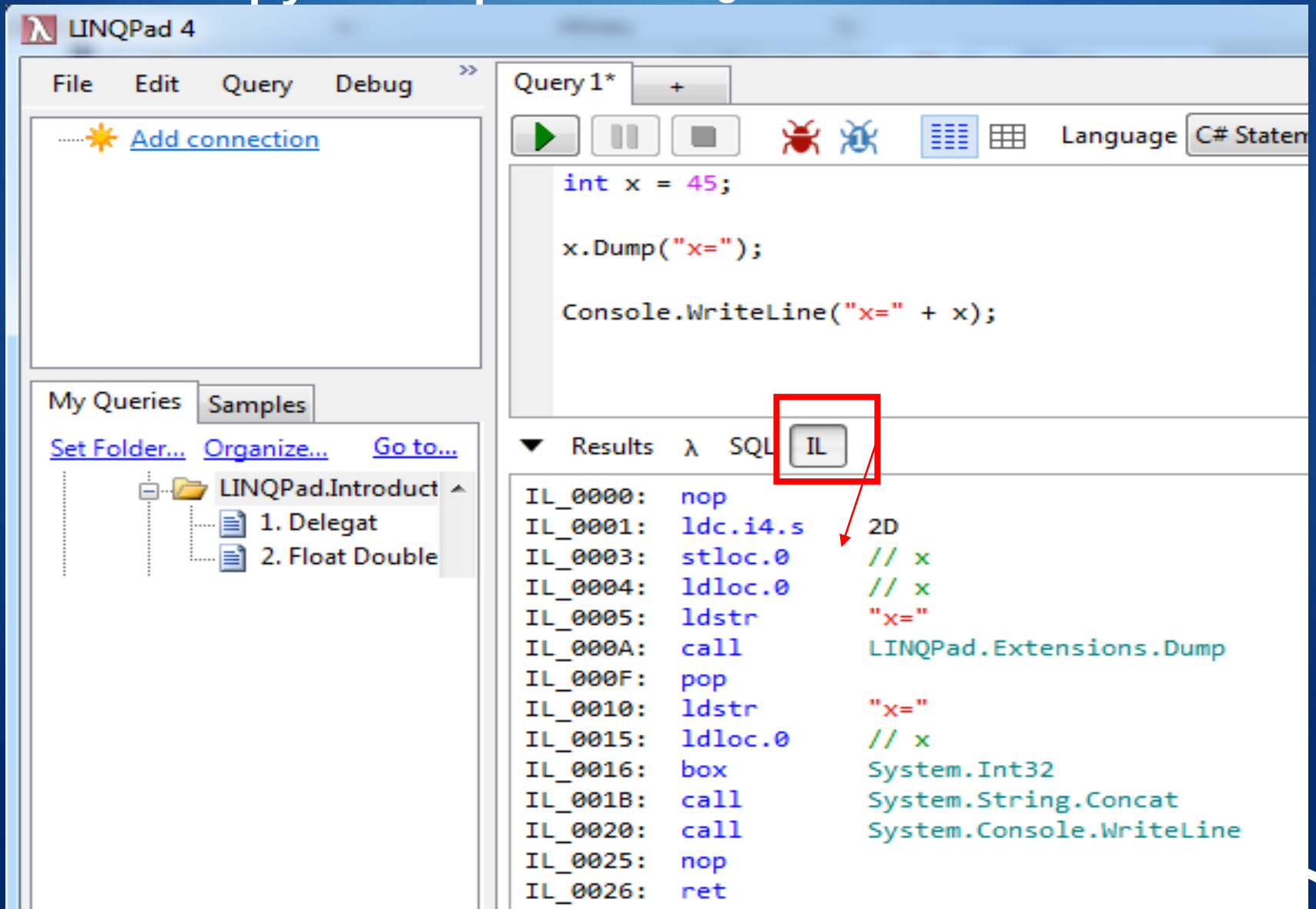
Инструментарий LINQPad



Инструментарий LINQPad



Инструментарий LINQPad



Инструментарий LINQPad

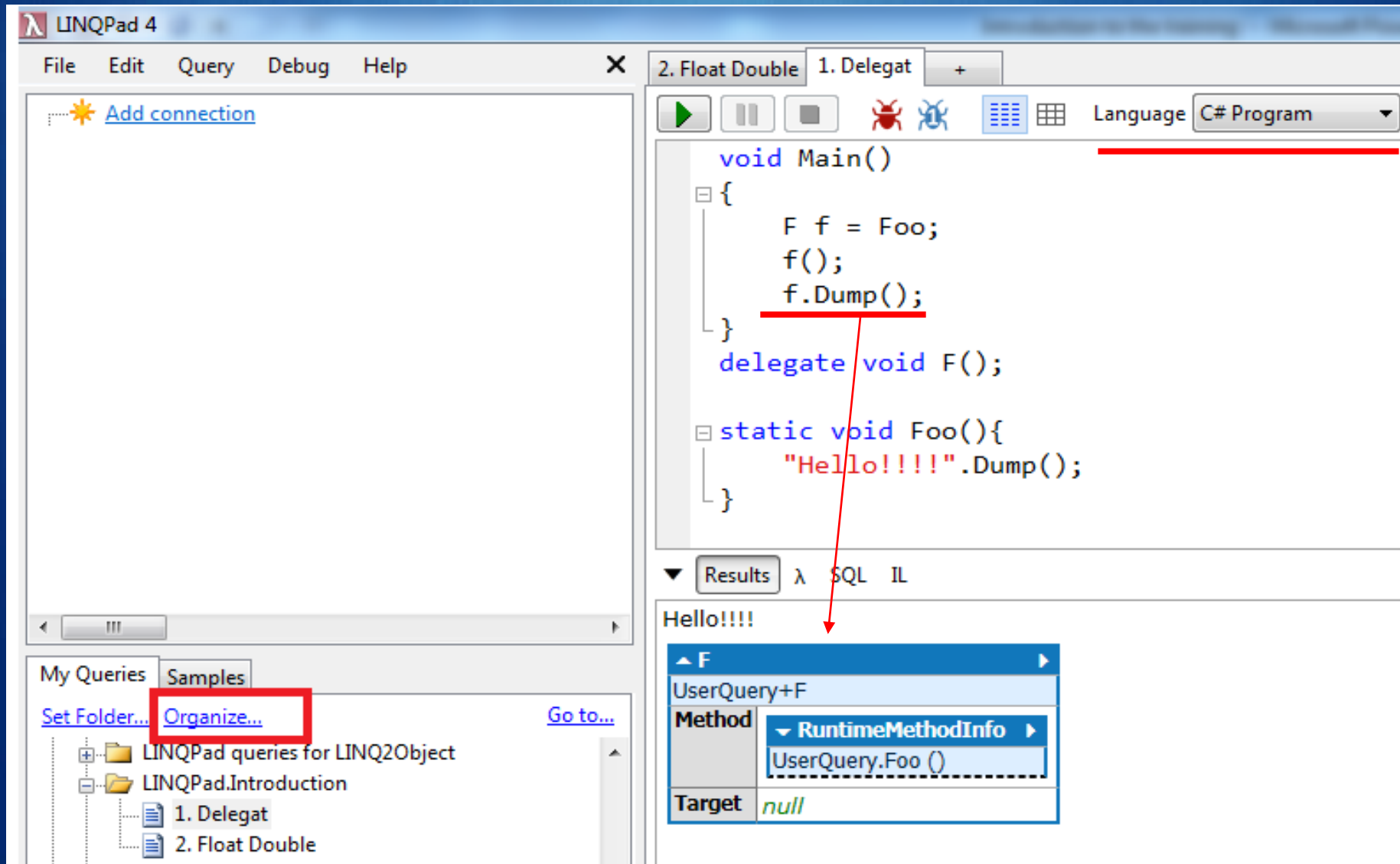
The screenshot displays the LINQPad 4 application window. The title bar reads "LINQPad 4". The menu bar includes "File", "Edit", "Query", and "Debug". On the left, there is a sidebar with a link to "Add connection" and a tree view under "My Queries" showing folders "1. LINQPad Queries new" and "LINQPad Queries", and a file "My Extensions". The main editor area shows a query named "Query 1*" with the following C# code:

```
int []arr = {1, 2, 3, 4, 5};  
arr.Dump("array");
```

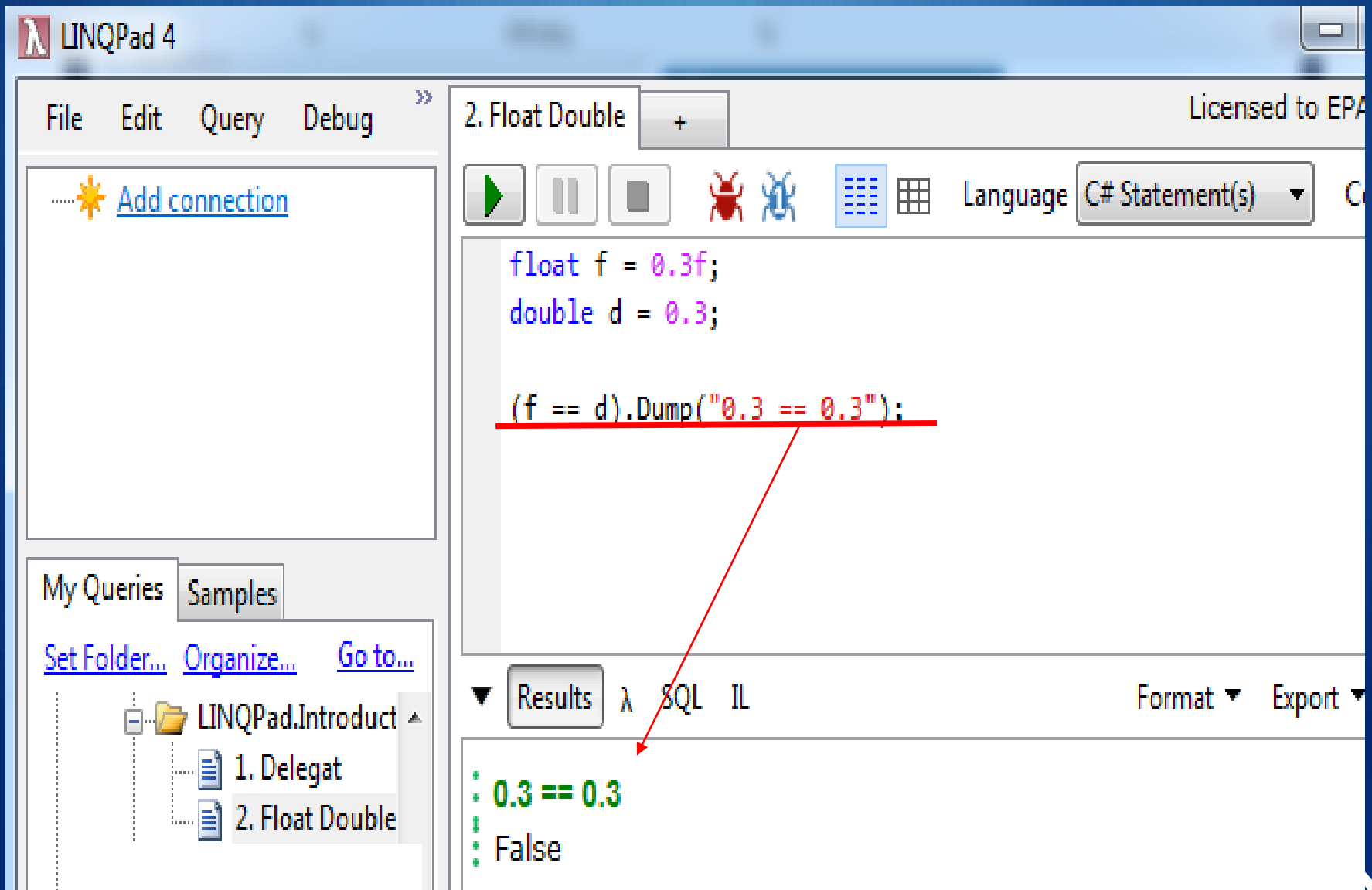
The code is executed, and the results are displayed in the "Results" tab. A red arrow points from the string "array" in the code to the label "array" in the results section. The results are shown as a table with 5 items:

Int32[] (5 items)
1
2
3
4
5

Инструментарий LINQPad



Инструментарий LINQPad



Example

Инструментарий LINQPad

The screenshot displays the LINQPad 4 application window. The top menu bar includes File, Edit, Query, and Debug. On the left, there is a 'My Queries' pane with a tree view showing a folder 'LINQPad.Introduct' containing two queries: '1. Delegat' and '2. Float Double'. The '2. Float Double' query is selected. The main editor area shows the following C# code:

```
float f = 0.3f;  
double d = 0.3;  
  
(f == d).Dump("0.3 == 0.3");
```

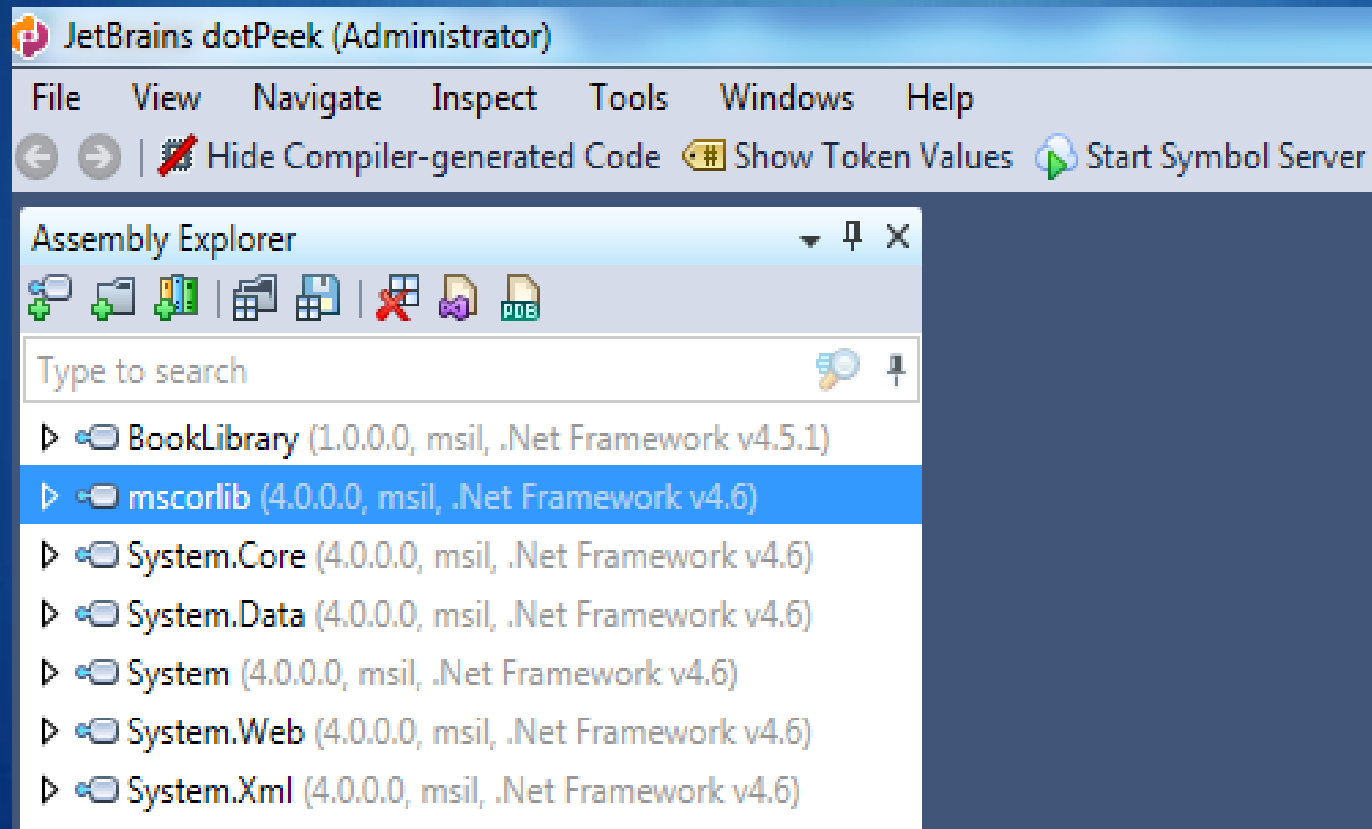
Below the code editor, the 'Results' tab is active, and the 'IL' (Intermediate Language) view is selected, which is highlighted with a red box. The IL code is as follows:

```
IL_0000: nop  
IL_0001: ldc.r4      9A 99 99 3E  
IL_0006: stloc.0     // f  
IL_0007: ldc.r8      33 33 33 33 33 33 D3 3F  
IL_0010: stloc.1     // d  
IL_0011: ldloc.0     // f  
IL_0012: conv.r8     // d  
IL_0013: ldloc.1     // d  
IL_0014: ceq  
IL_0016: ldstr       "0.3 == 0.3"  
IL_0018: call        LINQPad.Extensions.Dump  
IL_0020: pop  
IL_0021: ret
```

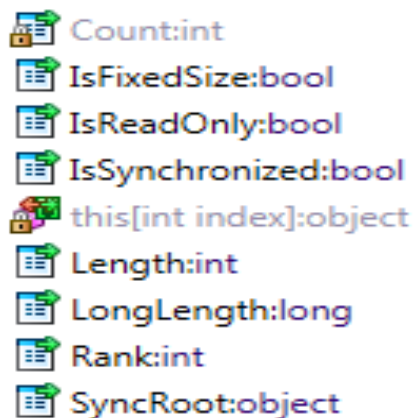
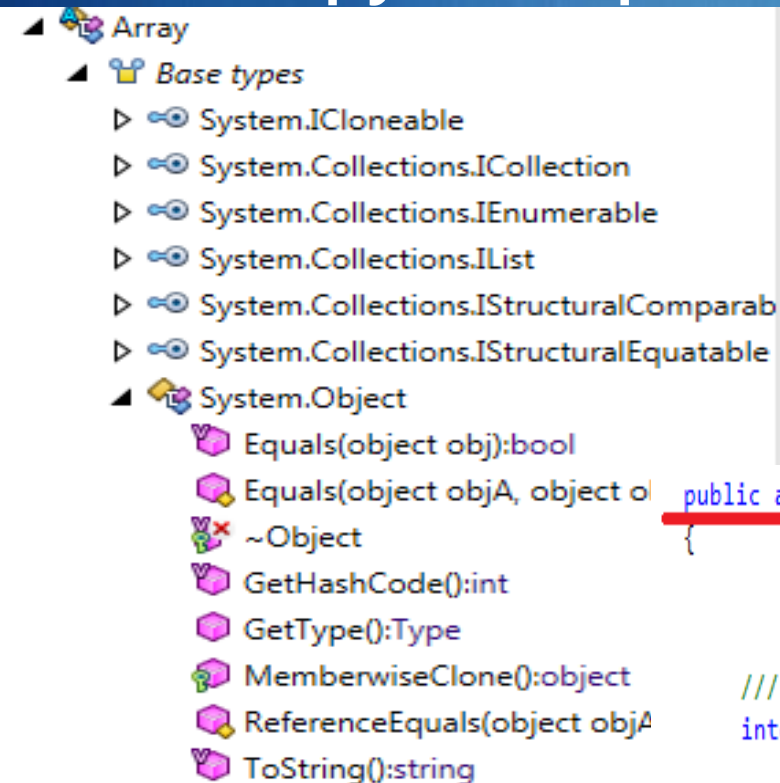


Example

Инструментарий dotPeek



Инструментарий dotPeek



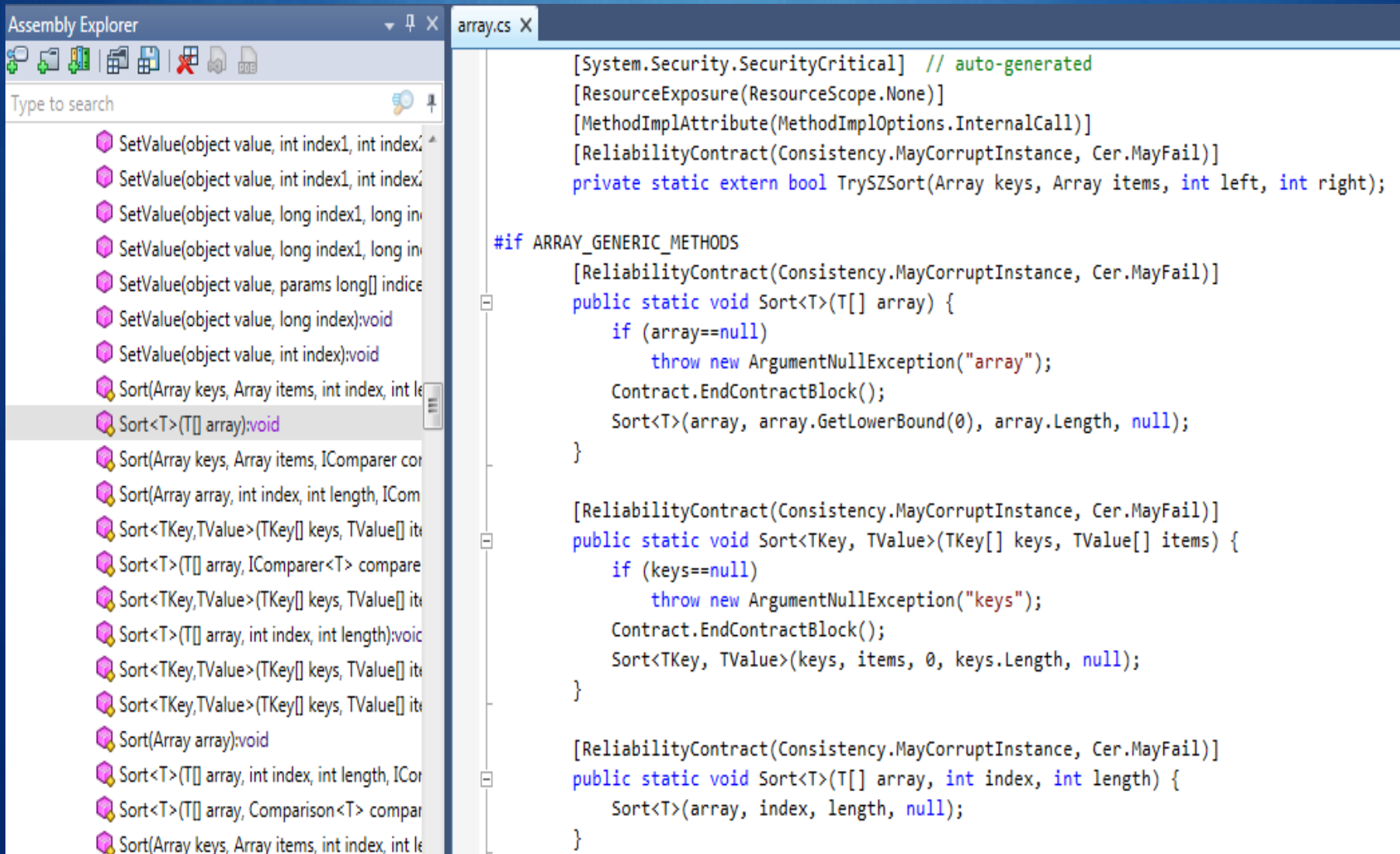
```
public abstract class Array : ICloneable, IList,
{
    IStructuralComparable,
    IStructuralEquatable

    /// <internalonly/>
    internal Array() {} // Don't call this. Internal for ArrayContracts type.

    public static ReadOnlyCollection<T> AsReadOnly<T>(T[] array) {
        if (array == null) {
            throw new ArgumentNullException("array");
        }
        Contract.Ensures(Contract.Result<ReadOnlyCollection<T>>() != null);

        // T[] implements IList<T>.
        return new ReadOnlyCollection<T>(array);
    }
}
```

Инструментарий dotPeek



Инструментарий dotPeek

JetBrains dotPeek (Administrator)

File View Navigate Inspect Tools Windows Help

Hide Compiler-generated Code Show Token Values Start Symbol Server

Assembly Explorer

Type to search

- ▲ MulticastDelegate
 - ▲ Base types
 - ▲ System.Delegate
 - ▲ Base types
 - System.ICloneable
 - System.Runtime.Serialization.ISerializable
 - System.Object
 - Delegate()
 - Delegate(Type target, string method)
 - Delegate(object target, string method)
 - AdjustTarget(object target, IntPtr methodPtr):IntPtr
 - Clone():object
 - CombineImpl(Delegate d):Delegate**
 - CompareUnmanagedFunctionPtrs(Delegate d1, Delegate d2):bool
 - CreateDelegate(Type type, object target, string method):Delegate
 - CreateDelegate(Type type, MethodInfo method):Delegate
 - Method:MethodInfo
 - Target:object
 - _methodBase:object
 - _methodPtr:IntPtr

multicastdelegate.cs delegate.cs X

```
newDelegate = source;
source = Remove(source, value);
}
while (newDelegate != source);

return newDelegate;
}

protected virtual Delegate CombineImpl(Delegate d)
{
    throw new MulticastNotSupportedException(Environment.GetResourceString("Mul

protected virtual Delegate RemoveImpl(Delegate d)
{
    return (d.Equals(this)) ? null : this;
}

public virtual Object Clone()
{
    return MemberwiseClone();
}

// V1 API.
public static Delegate CreateDelegate(Type type, Object target, String method)
```

Инструментарий dotPeek

```
namespace ClassLibrary
{
    ССЫЛОК 3
    public class Book
    {
        ССЫЛОК 0
        public int ID { get; set; }
        ССЫЛОК 2
        public string Title { get; set; }
        ССЫЛОК 2
        public string Author { get; set; }

        private const int Size = 1000;

        private readonly double PI = 3.14;

        private Car car = new Car();
        ССЫЛКА 1
        public Book(string title, string author)
        {
            this.Author = author;
            this.Title = title;
        }
        ССЫЛОК 0
        public override string ToString()
        {
            return string.Format("{0}, {1}, {2} ", Title, Author, Size + PI);
        }
    }
}
```

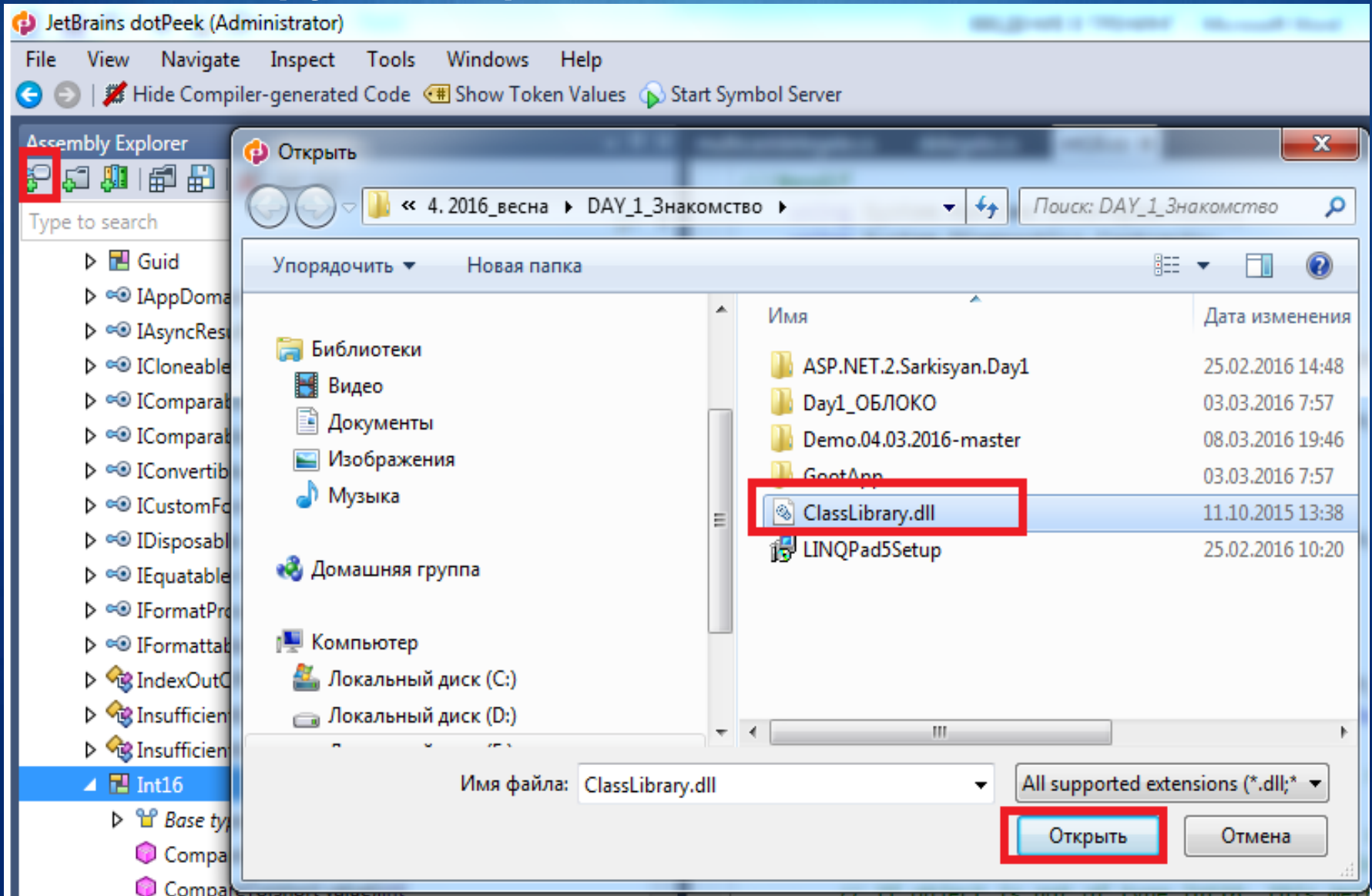
Решение "ASP.NET.2.Sarkisyan.Day1"

- ClassCar
 - Properties
 - References
 - Car.cs
- ClassLibrary
 - Properties
 - References
 - Book.cs
- ConsoleClassLibrary
 - Properties
 - References
 - App.config
 - Program.cs

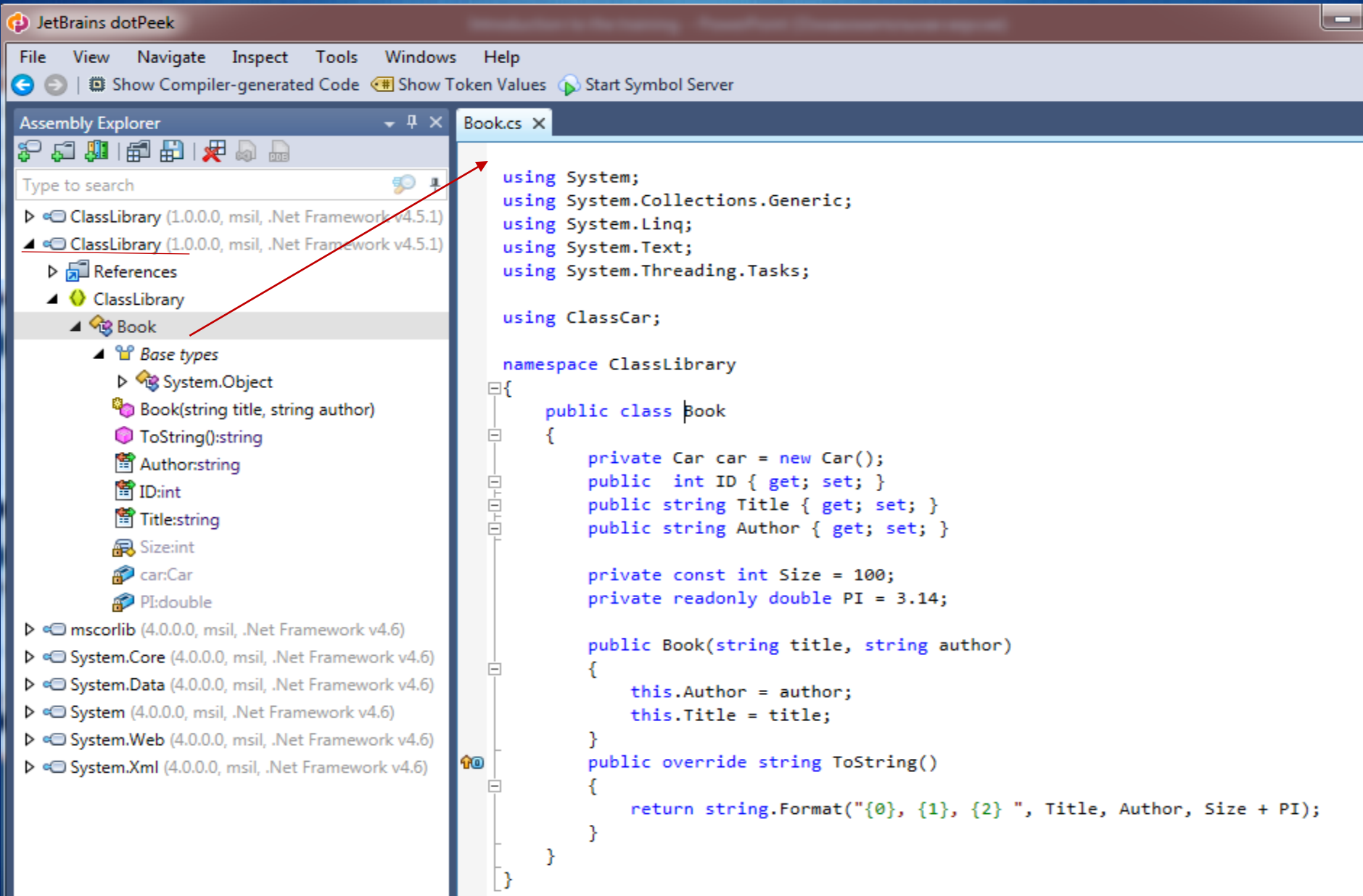


Example

Инструментарий dotPeek



Инструментарий dotPeek



Инструментарий *IL Dasm*

ClassLibrary
ClassLibrary.Book

- .class public auto ansi beforefieldinit
- <Author>k__BackingField : private string
- <ID>k__BackingField : private int32
- <Title>k__BackingField : private string
- PI : private initonly float64
- Size : private static literal int32
- car : private class [ClassCar]ClassCar.Car
- .ctor : void(string, string)
- ToString : string()
- get_Author : string()
- get_ID : int32()
- get_Title : string()
- set_Author : void(string)
- set_ID : void(int32)
- set_Title : void(string)
- Author : instance string()
- ID : instance int32()
- Title : instance string()

```
public int ID { get; set; }  
CCIMOK 2  
public string Title { get; set; }  
CCIMOK 2  
public string Author { get; set; }  
  
private const int Size = 1000;  
  
private readonly double PI = 3.14;  
  
private Car car = new Car();
```



Example

C#/.NET & ASP.NET MVC источники

For a beginner in .NET

- **Andrew Troelsen** (Эндрю Троелсен) Pro C# 5.0 and the .NET 4.5 Framework
- **Иэн Гриффитс** Программирование на C# 5.0

For the advanced .NET Developer

- **Jeffrey Richter** (Джеффри Рихтер) CLR via C#
- **Jon Skeet** (Джон Скит) C# in Depth
- **Joseph Albahari** (Джозеф Албахари), **Ben Albahari** (Бен Албахари) C# 5.0 in a Nutshell
- **Bart De Smet** (Барт де Смет) C# 5.0 Unleashed
- **Sasha Goldshtein** Pro .NET Performance: Optimize Your C# Applications
- **Jeffrey Palermo** (Джеффри Палермо) ASP.NET MVC 4 in Action
- **Adam Freeman** (Адам Фриман) Pro ASP.NET MVC 5
- **Mark Seemann** (Марк Сииман) Dependency Injection in .NET
- **Roy Oshero** The Art of Unit Testing with Examples in C#

<http://csharpindepth.com>

<http://habrahabr.ru>

<http://rsdn.org>

<http://stackoverflow.com>

<http://blogs.msdn.com/b/ruericlippert>

<http://msdn.microsoft.com>

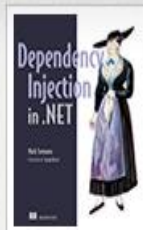
<http://www.entityframeworktutorial.net>

<http://www.asp.net/mvc>

<http://metanit.com>

<http://sergeyteplyakov.blogspot.com>

<http://smarly.net>



Внедрение зависимости в .NET

Марк Симан

Книга "Внедрение зависимостей в .NET" демонстрирует основные DI паттерны на обычном языке C#, поэтому вы в полной мере осознаете, как работает механизм внедрения зависимостей, кроме того она охватывает интеграцию с такими стандартными технологиями компании Microsoft, как ASP.NET MVC, и учит вас, как использовать такие DI фреймворки, как Structure Map, Castle Windsor и Unity.

Оглавление

Появление механизма внедрения зависимостей

1. "Дегустационное меню" механизма внедрения зависимостей
2. Комплексный пример
3. DI-контейнеры

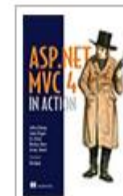
Каталог DI

4. DI паттерны
5. DI анти-паттерны
6. DI рефакторинг

DI самостоятельно

7. Построение объектов
8. Жизненный цикл объектов
9. Механизм перехвата

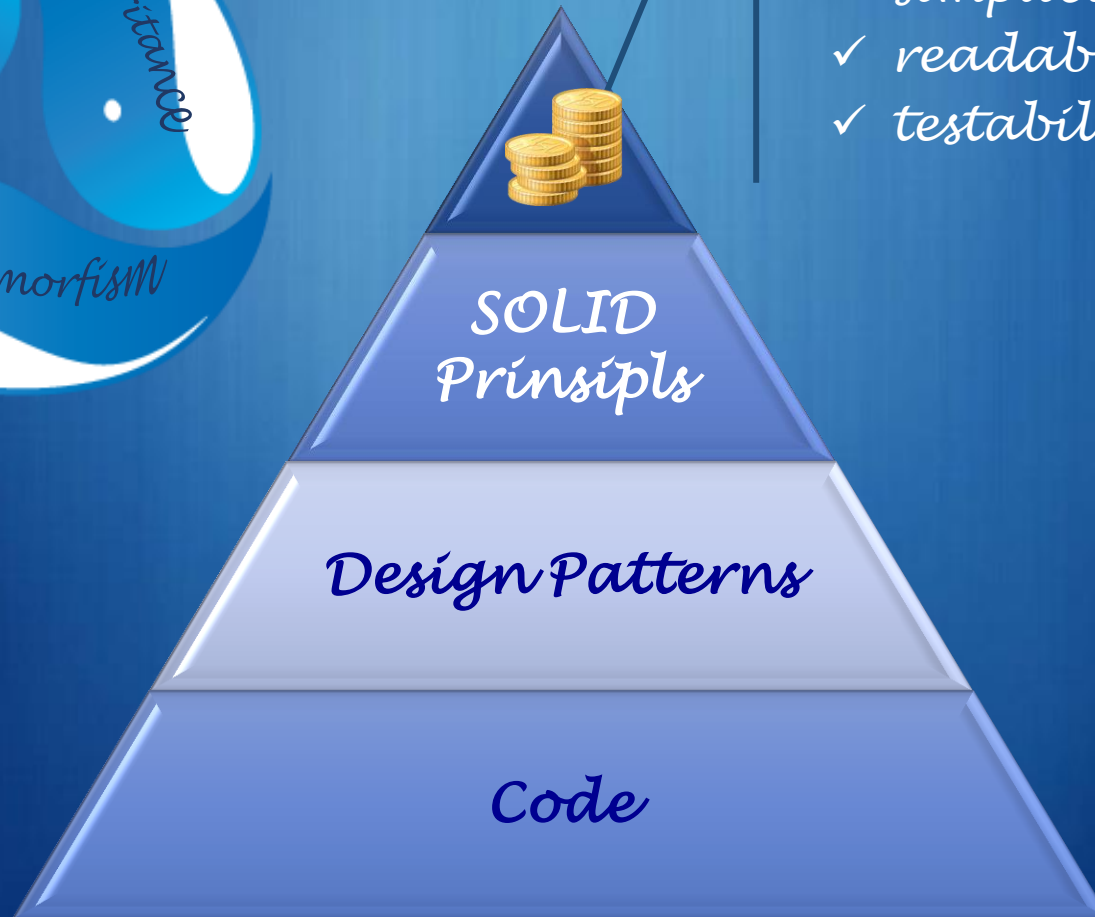
DI-контейнеры



Цели и задачи тренинга



ООП. Принципы и шаблоны проектирования



- ✓ *extensibility, agility*
- ✓ *maintainability*
- ✓ *simplicity*
- ✓ *readability, clarity*
- ✓ *testability*

Принцип объектно-ориентированного программирования

SRP: Single Responsibility Principle

Принцип единственной обязанности (Single Responsibility Principle SRP), который гласит,:

1. Не должно быть больше одной причины для изменения класса.
2. Каждый класс должен быть сфокусирован на своей области ответственности.

Цель принципа:

- **упростить внесение изменений** (Separation of Concerns и High Cohesion)
- **защититься от побочных эффектов при изменениях** (Don't Repeat Yourself и инкапсуляция)

Общие принципы проектирования

DRY: Don't Repeat Yourself

Достигается за счет того, что

- Отсутствием copy-paste;
- Повторного использования кода, reusable code.

SoC: Separation of Concerns

Separation of Concerns - *разделение системы на отдельные части.*

При этом эти части должны взаимодействовать и быть слабо связанными между собой.

Достигается за счет:

- *разделение на звенья (tiers)*
- *разделение на слои (layers)*
- *модульность*
- *разделение на классы*
- *инкапсуляция*

Low Coupling

Написать слабосвязанный код позволяет *зависимость от интерфейсов*.

«Программируйте не на основе классов, а на основе интерфейсов».

High Cohesion

High Cohesion – мера, определяющая связанность и сфокусированность обязанностей/ответственности элемента.

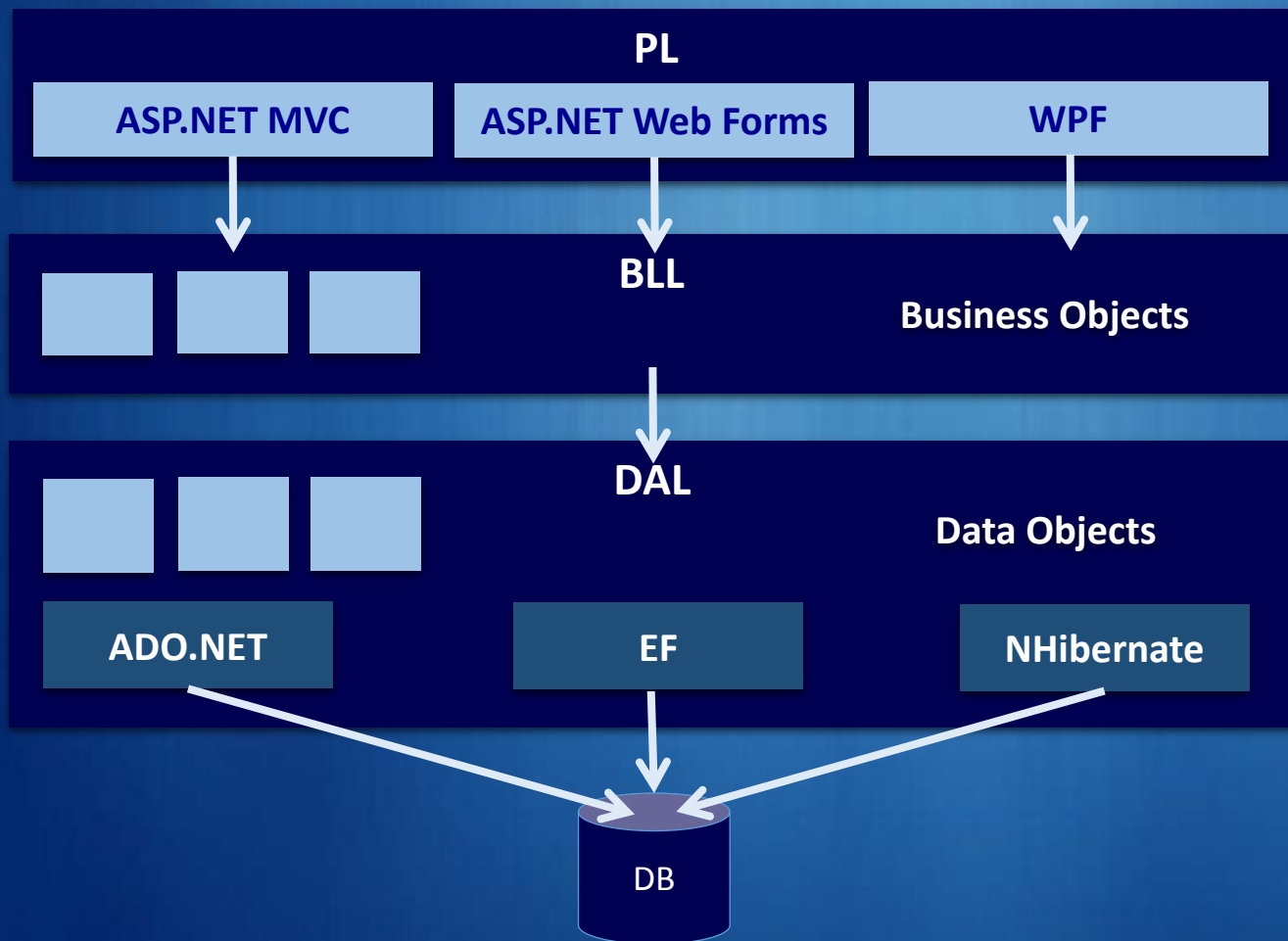
Достигается за счет:

- сужение обязанностей элемента
- разделение ответственности между несколькими элементами
- группирование похожей ответственности в одном элементе



Example

Структура проекта



Design Patterns

MVC

MVVM

Facade

Data Transfer Object

Strategy

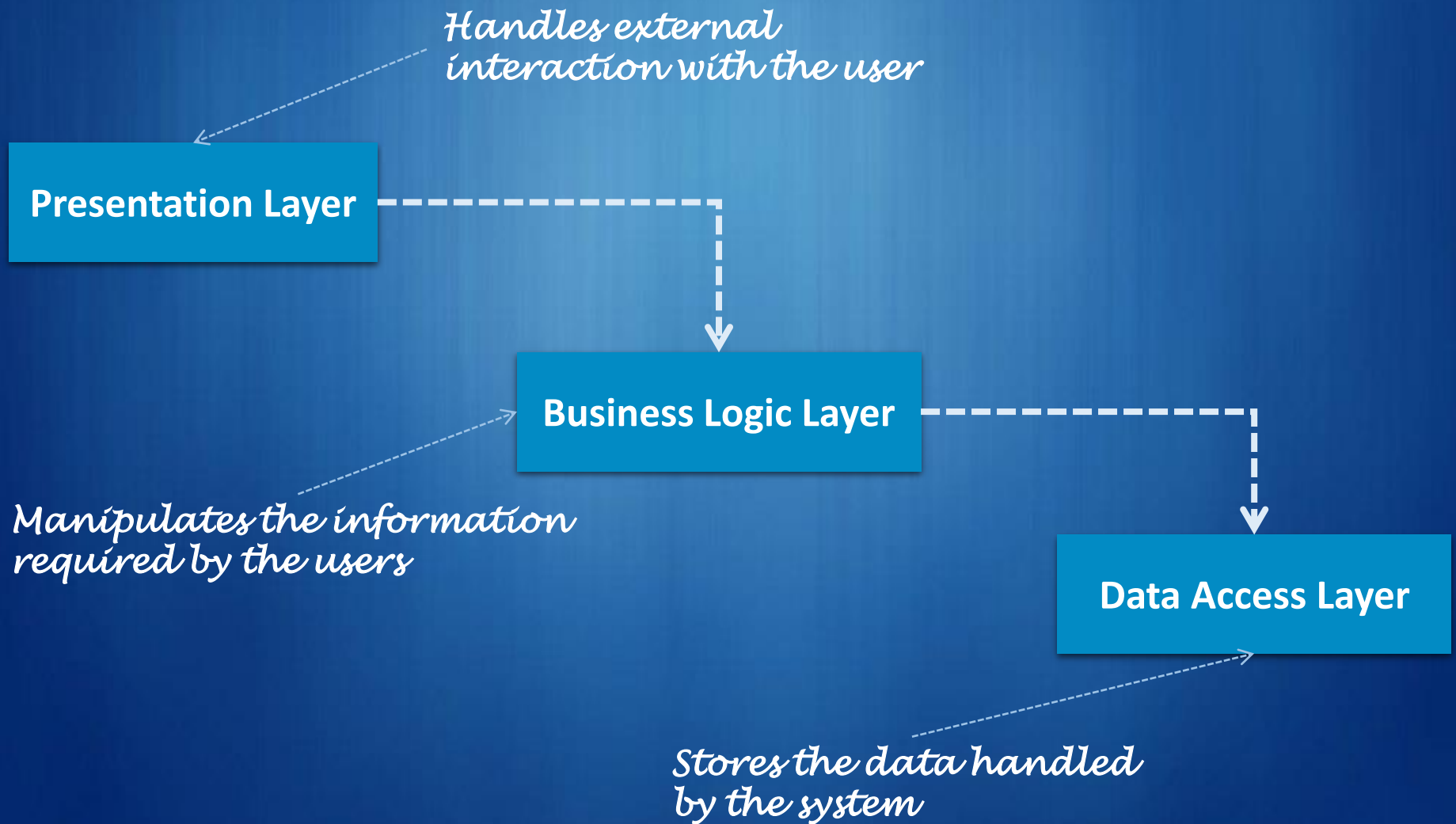
Lazy Load

Provider

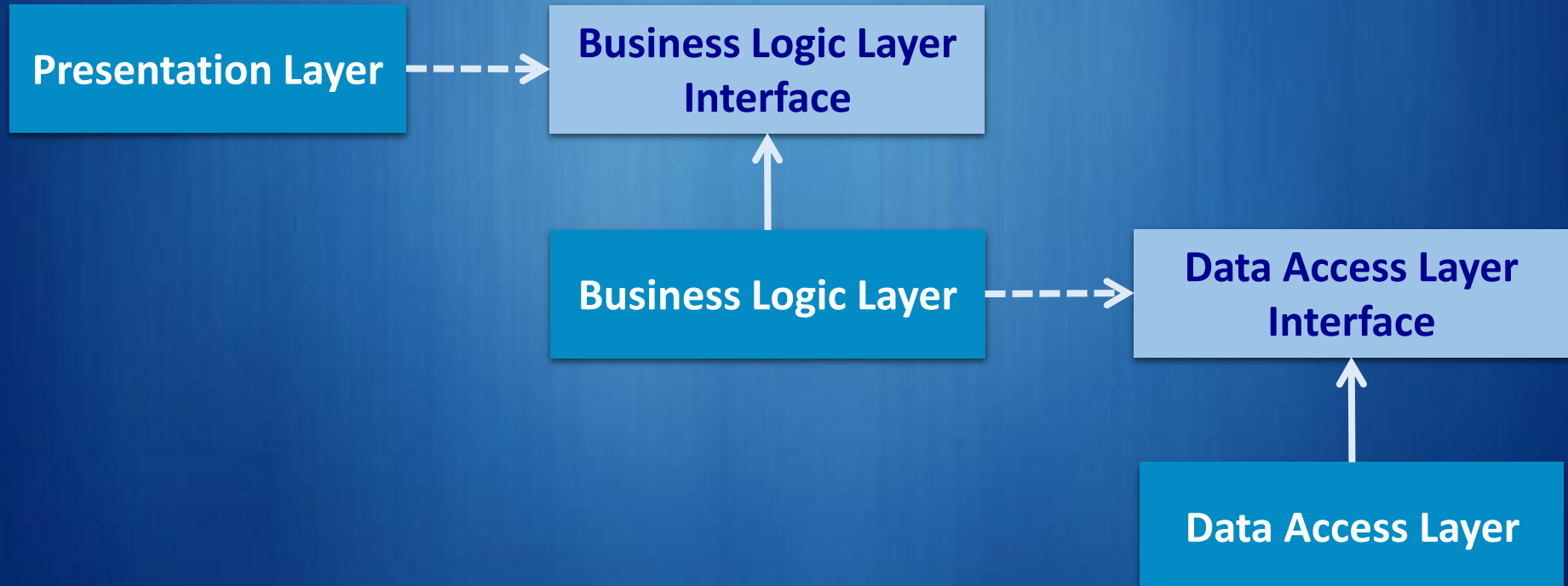
Repository

UnitOfWork

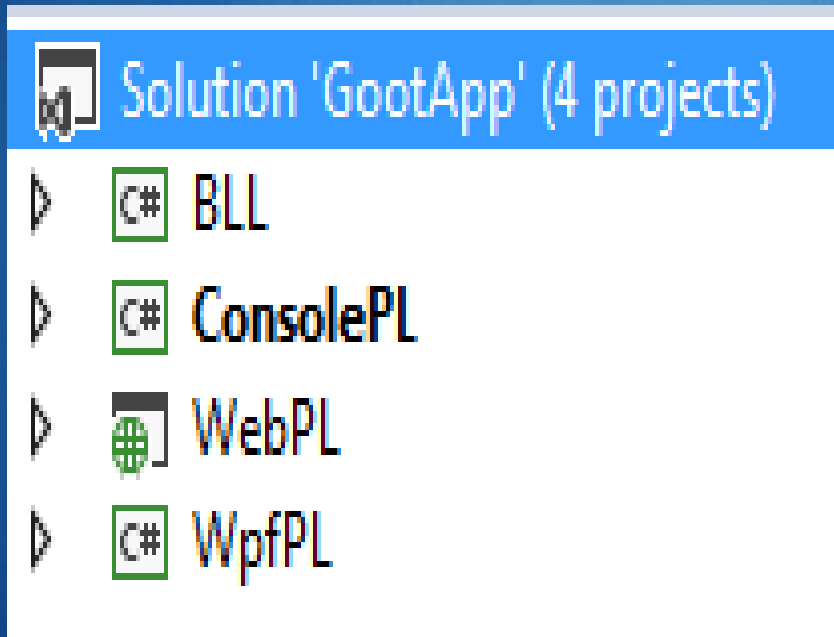
Структура проекта



Структура проекта



Example

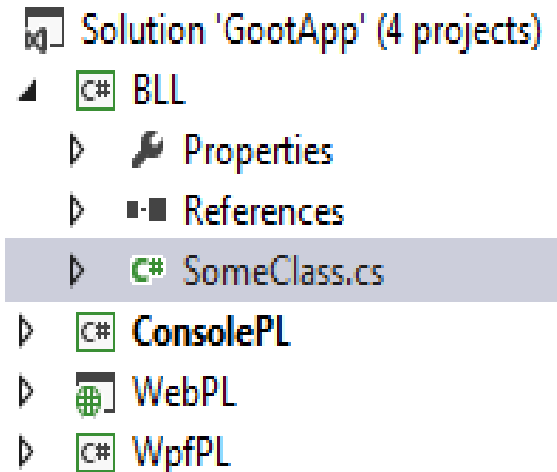


```

namespace BLL
{
    3 references
    public static class SomeClass
    {
        3 references
        public static string SomeMethod(string userName)
        {
            if (userName == null)
            {
                throw new ArgumentException("userName");
                //throw new ArgumentException(nameof(userName));
            }
            if (userName == string.Empty)
            {
                return "Hello, anonim!";
            }

            return string.Format("Hello, {0}!", userName);
            //return userName = $"Hello, {name}!";
        }
    }
}

```



Assembly Explorer

Type to search

File

Base types

System.Object

AppendAllLines(string path, IEnumerable<string> contents)

AppendAllLines(string path, IEnumerable<string> contents)

AppendAllText(string path, string contents)

AppendAllText(string path, string contents)

AppendText(string path, StreamWriter)

Copy(string sourceFileName, string destFile)

Copy(string sourceFileName, string destFile)

Create(string path, int bufferSize, FileOptions)

Create(string path, int bufferSize, FileOptions)

file.cs

```
sw.Write(contents);
}

[ResourceExposure(ResourceScope.Machine)]
[ResourceConsumption(ResourceScope.Machine)]
public static void AppendAllLines(String path, IEnumerable<String> contents)
{
    if (path == null)
        throw new ArgumentNullException("path");
    if (contents == null)
        throw new ArgumentNullException("contents");
    if (path.Length == 0)
        throw new ArgumentException(Environment.GetResourceString("Argument_EmptyPath"));
    Contract.EndContractBlock();

    InternalWriteAllLines(new StreamWriter(path, true, StreamWriter.UTF8NoBOM), contents);
}
```



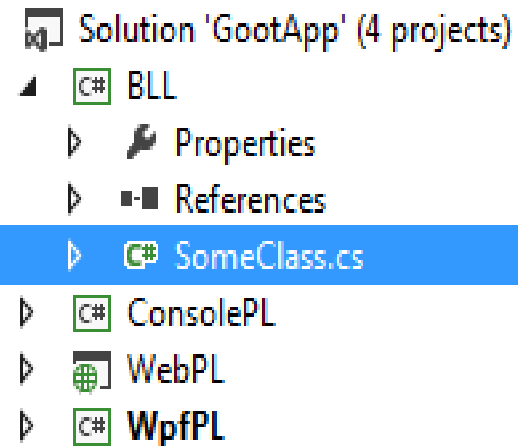
Example

```

namespace BLL
{
    3 references
    public static class SomeClass
    {
        3 references
        public static string SomeMethod(string userName)
        {
            if (userName == null)
            {
                throw new ArgumentNullException("userName");
                //throw new ArgumentNullException(nameof(userName));
            }
            if (userName == string.Empty)
            {
                return "Hello, anonim!";
            }

            return string.Format("Hello, {0}!", userName);
            //return $"Hello, {userName}!";
        }
    }
}

```



```
//using static System.Console;
//using static BLL.SomeClass;
namespace ConsolePL
{
```

0 references

```
class Program
```

```
{
```

0 references

```
static void Main(string[] args)
```

```
{
```

```
    Console.WriteLine("Enter your name:");
```

```
    string name = Console.ReadLine();
```

```
    Console.Clear();
```

```
    Console.WriteLine(SomeClass.SomeMethod(name));
```

```
    Console.ReadKey();
```

```
}
```

```
}
```

```
}
```

▲ C# ConsolePL

▸ Properties

▸ References

App.config

▸ C# Program.cs

```
//Write("Enter user name:");
```

```
//string userName = ReadLine();
```

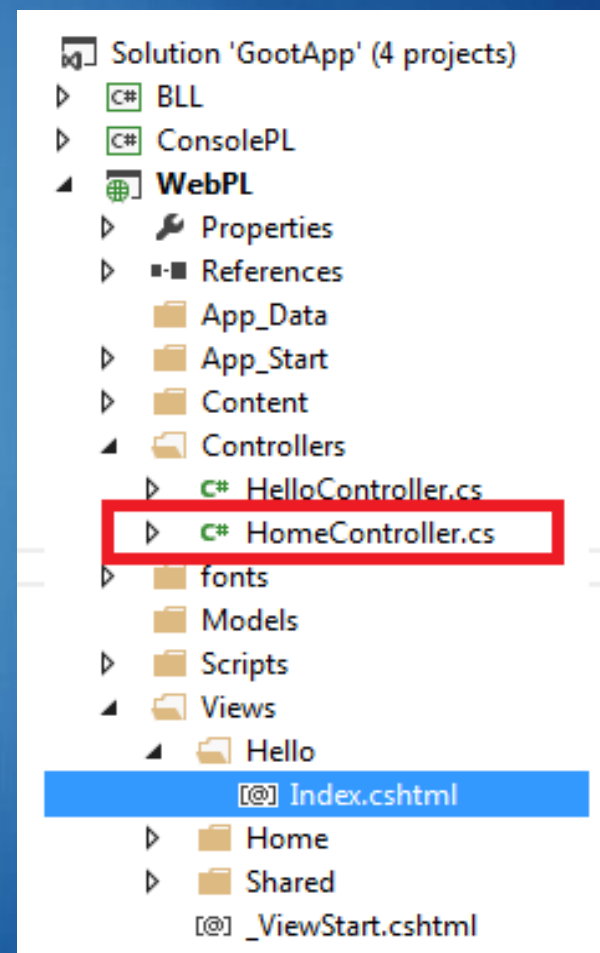
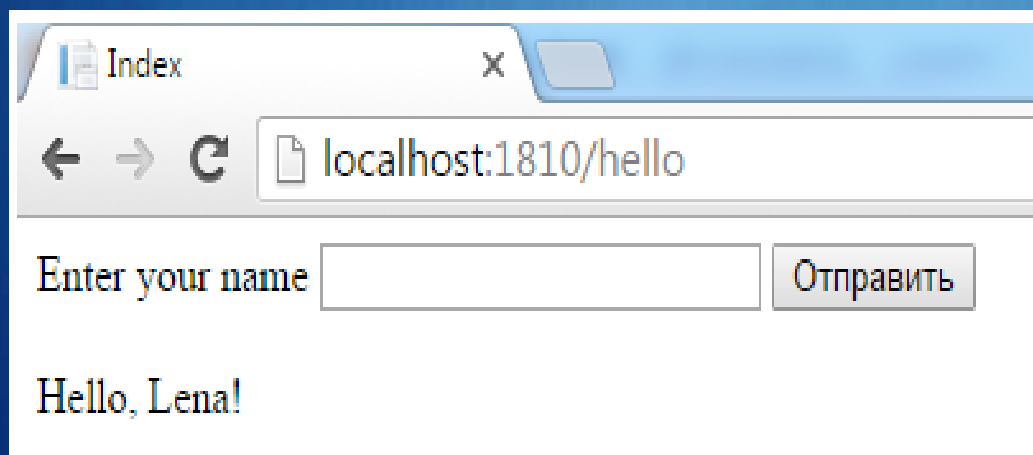
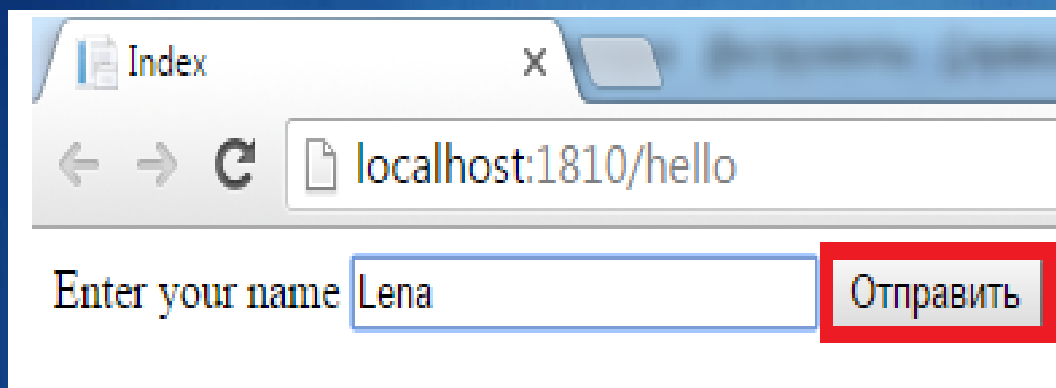
```
//Clear();
```

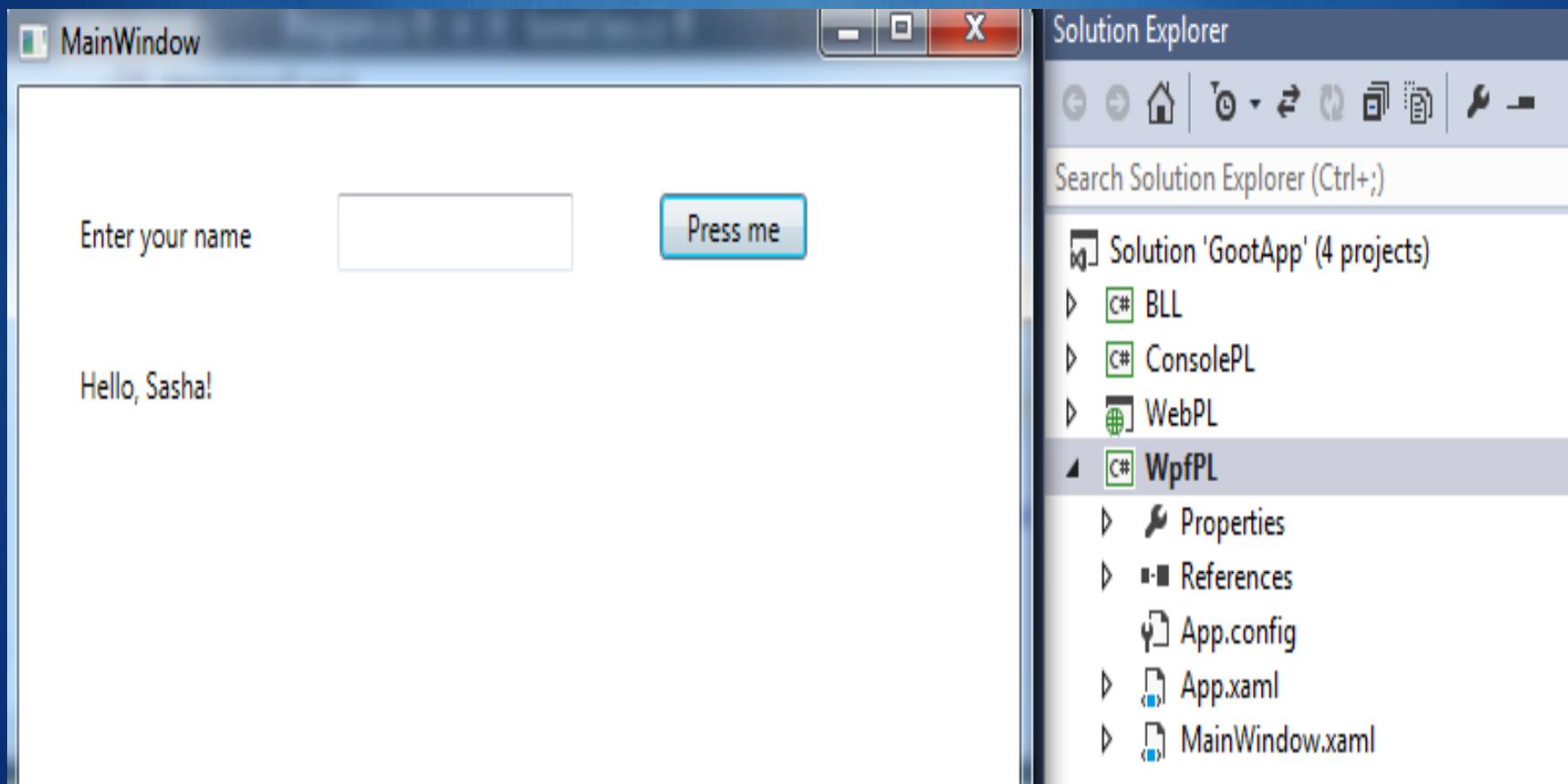
```
//WriteLine(GreetingMethod(userName));
```

```
//ReadKey();
```



Example





Example

Тренинги будут проходить
2 раза в неделю

ВТОРНИК 09 : 00 – 13 : 00

ЧЕТВЕРГ 09 : 00 – 13 : 00

Суббота Резервный день

Продолжительность тренинга

Start 09.03.17

Finish 23.05.17

Создать аккаунт на gmail.com

1. *фамилией и именем*
2. *номером телефона*
+375 XX XXX XX XX
(указать velcom, mts, life)
3. *электронный адрес и GitHub*





gayane.f.sarkisyan@gmail.com

+375 29 500 27 50

+375 44 777 27 50

gayane.f.sarkisyan@gmail.com

Инструментарий *Google Диск*

По названию ↑		Владелец	По дате изм...
Для просмотра			
 Литература		я	4 мар. 2017 г.
 Day 2		я	12:19
 Introduction to the training		я	6 мар. 2017 г.
 STUDENTS		я	4 мар. 2017 г.

Инструментарий *Google Диск*

Литература ▾

По названию ↑

Владелец

По дате изм...

1. C# & .NET Framework

2. Проектирование&Паттерны

3. ASP NET MVC

4. Параллельное программирование

Инструментарии

Джон Скит

Richter J. - CLR via C# - 2012-2013

Джозеф Албахари, Бен Албахари - C# 5.0. Справочник. Полное о...

Программирование на C# 5.0.zip

Albahari J., Albahari B. - C# 5.0 in a Nutshell, 5th Edition - 2012.pdf

Andrew Troelsen. Pro C# 5.0 and the .NET 4.5 Framework.djvu

Bart De Smet - C# 4.0 Unleashed - 2010.pdf

Programming C# 5.0 V413HAV.pdf

Инструментарий *Google Диск*

Литература ▼

По названию ↑

- 1. C# & .NET Framework
- 2. Проектирование&Паттерны
- 3. ASP NET MVC
- 4. Параллельное программирование с п
- Инструментарии

PDF Копия "Копия "Кормен, Лейзерсон, Ривест - Алгоритмы, пос..."

PDF Копия "Копия "Thomas H. Cormen, Charles E. Leiserson, Ronald L. ..."

PDF Тепляков С. Паттерны проектирования на платформе .NET.pdf

PDF Фримен - Паттерны проектирования - 2011.pdf

Чистый код. Роберт Мартин. 2010.djvu

PDF Lakman_G_-_Karyera_programmista_Kak_ustroits.pdf

PDF Pro .NET Performance. Sasha Goldshtein.pdf



Инструментарий *Google Диск*



Инструментарий *Google Диск*

Литература ▾

По названию ↑	Владелец	По дате изм...
1. C# & .NET Framework	я	12:48
2. Проектирование&Паттерны		12:48
3. ASP NET MVC		12:48
4. Параллельное программирование		12:48
Инструментарии	я	12:51

 ildasm.exe
 LINDPad_new_5.zip

Инструментарий *Google Диск*

Introduction to the training

По названию ↓



LINQPad.Introduction.zip



Introduction to the training .pdf



GootApp.zip



Day1_Tasks.docx



ASP.NET.2.Sarkisyan.Day1.zip

Каждый Task

Deadline – 10:00, 12.03.2017.

Ознакомится с материалом

- Джеффри Рихтер. CLR via C#. Часть I. Основы CLR.
- Что нужно знать про арифметику с плавающей запятой (<http://habrahabr.ru/post/112953/>)

Задание 1.

.....











Задание 2.

.....

Ознакомится с материалом *по книге C# 5.0 IN A NUTSHELL. Chapter 6.*

Задание 3.

Инструментарий *Google Диск*

STUDENTS ▼		
По названию ↑	Владелец	По дате изм...
 Бурдейко Тимофей	я	4 мар. 2017 г.
 Веремейчик Илья	я	4 мар. 2017 г.
 Войнич Анжелика	я	4 мар. 2017 г.
 Володько Дмитрий	я	4 мар. 2017 г.
 Захарова Александра	я	4 мар. 2017 г.
 Ковбан Евгений	я	4 мар. 2017 г.
 Куриленко Александр	я	4 мар. 2017 г.
 Ломако Алексей	я	4 мар. 2017 г.
 Менгель Эдгар	я	4 мар. 2017 г.
 Пашковский Сергей	я	4 мар. 2017 г.

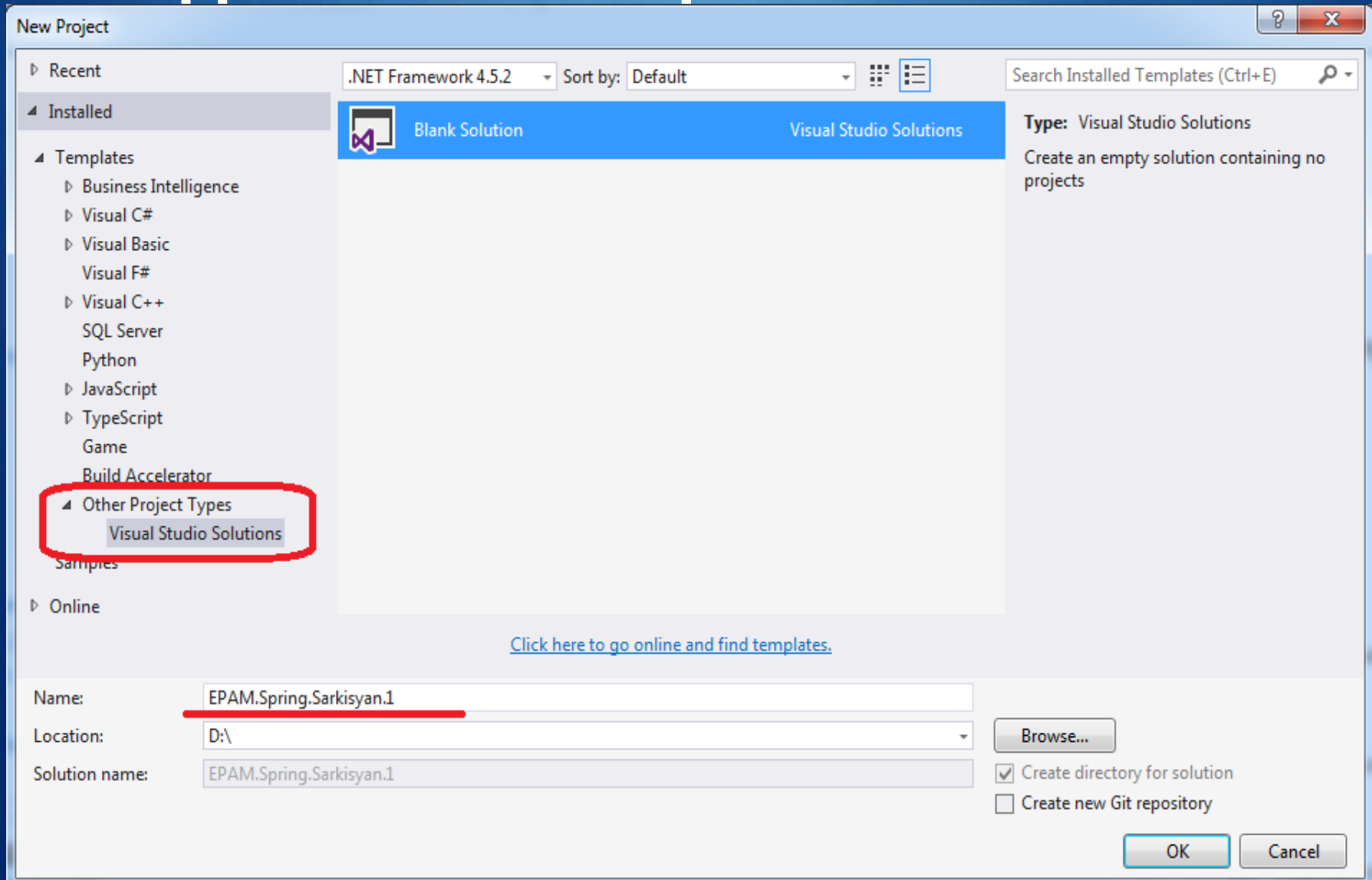
Требование к выложенному проекту

EPAM.Spring.Фамилия.№

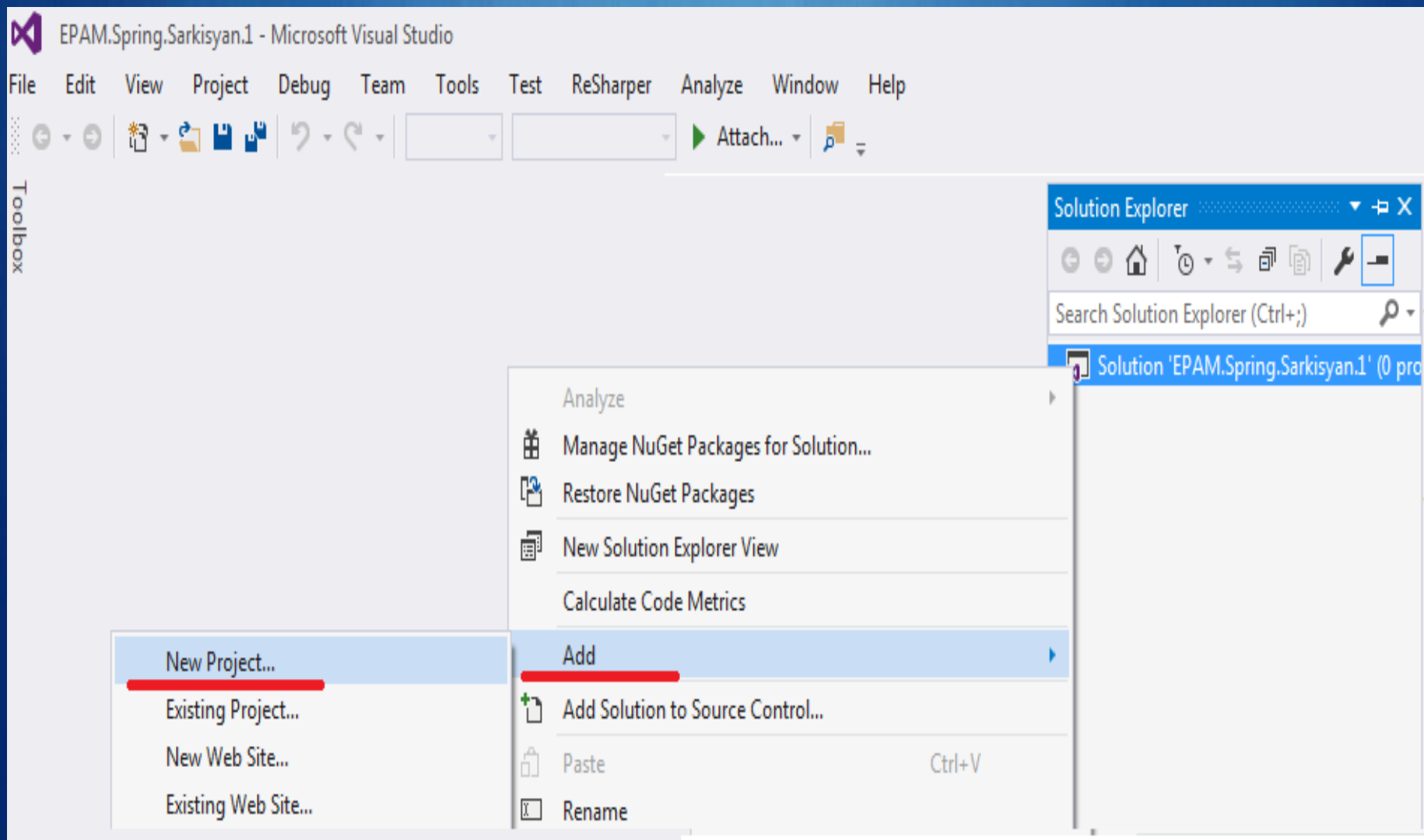


EPAM.Spring.Sarkisyan.1 zip

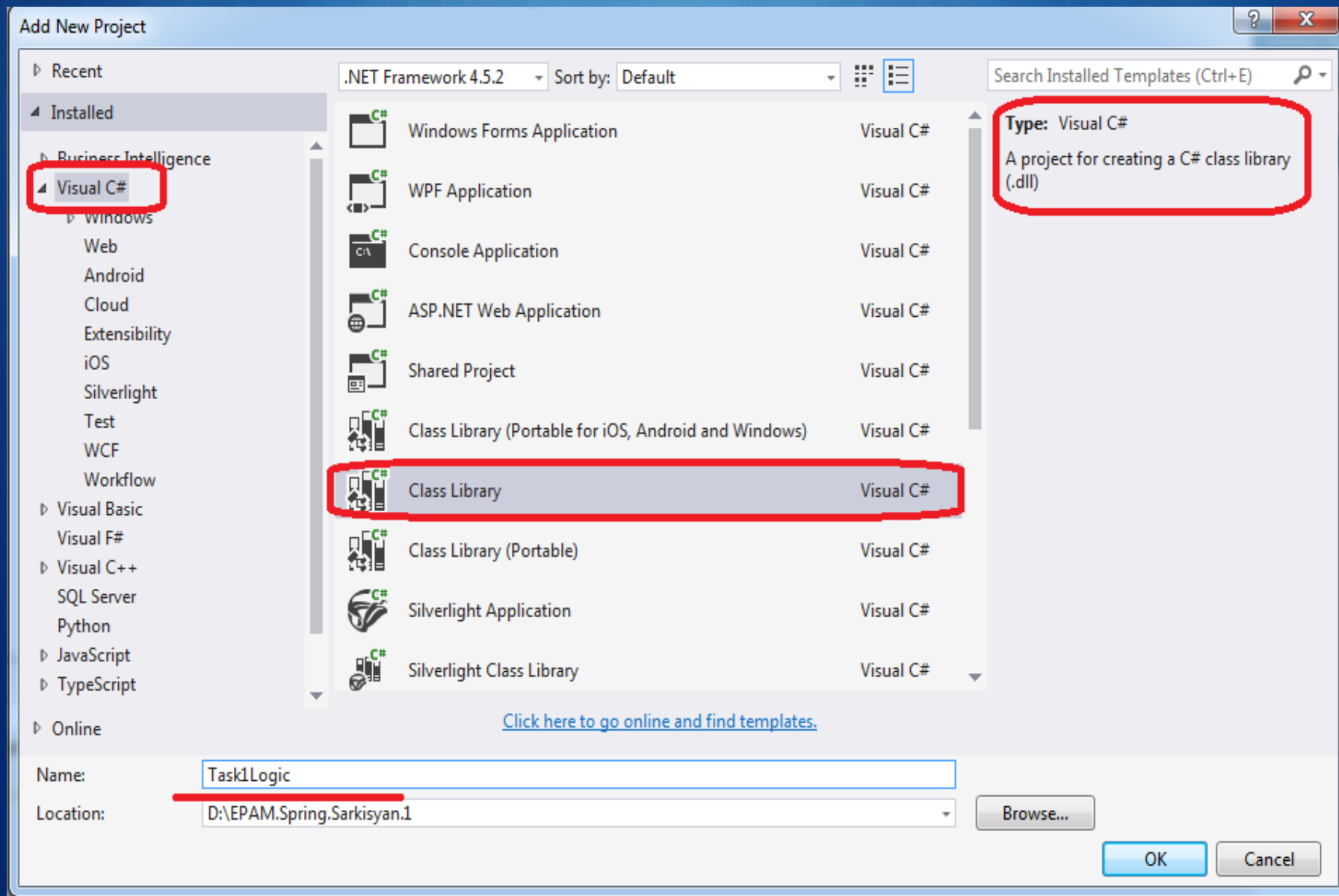
Все задания должны находиться в одном бланке решений.



Для решения каждого задания создаем приложение типа Library



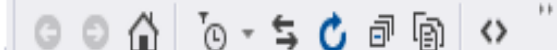
Для решения каждого задания создаем приложение типа Library



Toolbox Class1.cs Task1Logic Task1Logic.Class1

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Task1Logic
8 {
9     0 references
10     public class Class1
11     {
12     }
13 }
```

Solution Explorer



Search Solution Explorer (Ctrl+;)

Solution 'EPAM.Spring.Sarkisyan.1' (1 project)

Task1Logic

Properties

References

Class1.cs

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Task1Logic
8 {
9     0 references
10     public class Class1
11     {
12     }
13 }
```



MaximumElementOfArray.cs* X

C# Task1Logic

Task1Logic.MaximumElementOfArray

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Task1Logic
8 {
9     public class MaximumElementOfArray
10     {
11     }
12 }
13
```

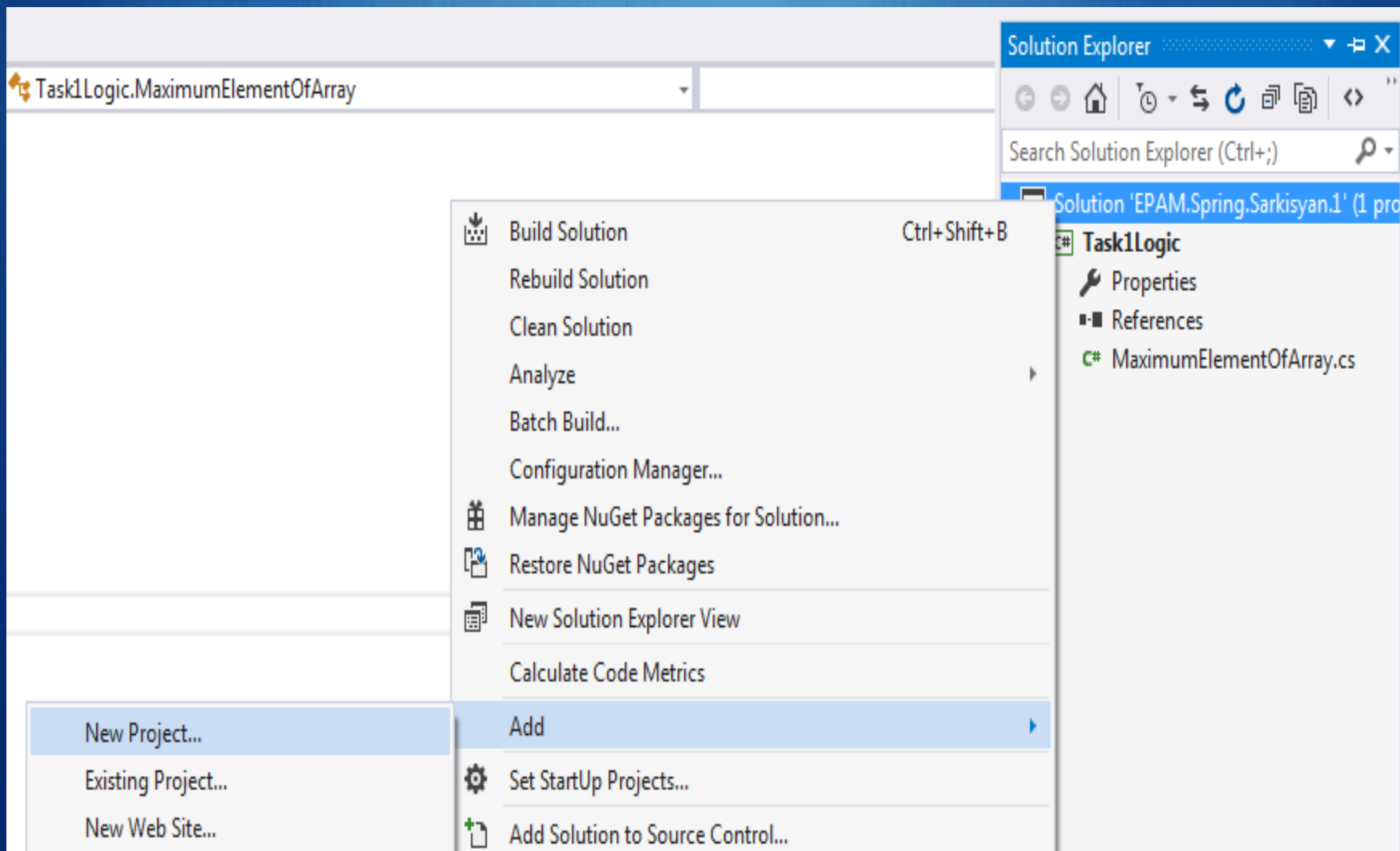
Solution Explorer X



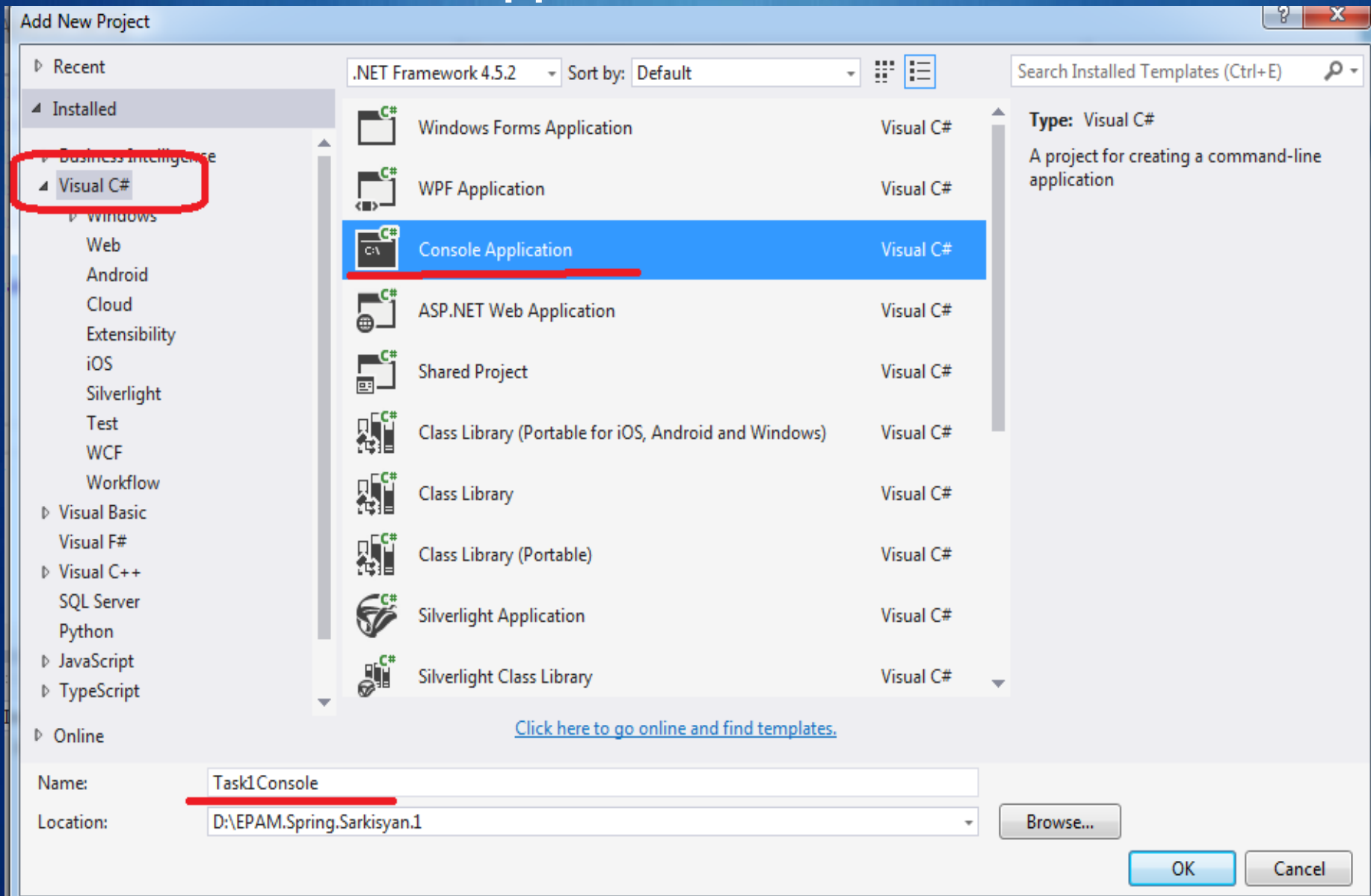
Search Solution Explorer (Ctrl+;) 🔍

- Solution 'EPAM.Spring.Sarkisyan.1' (1 project)
- Task1Logic
 - Properties
 - References
 - MaximumElementOfArray.cs

Для представления решения каждого задания создаем Console Application



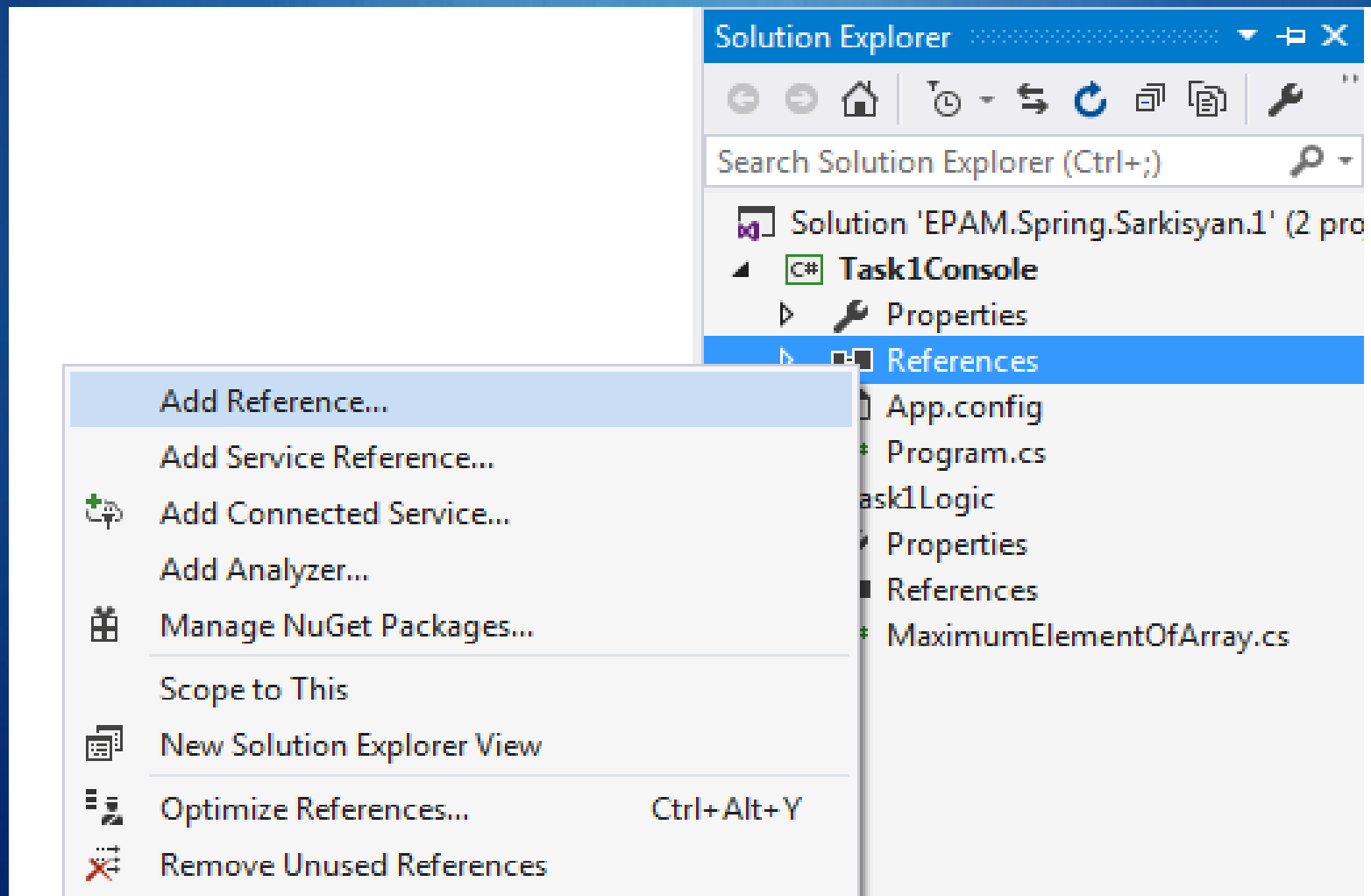
Для представления решения каждого задания создаем Console Application



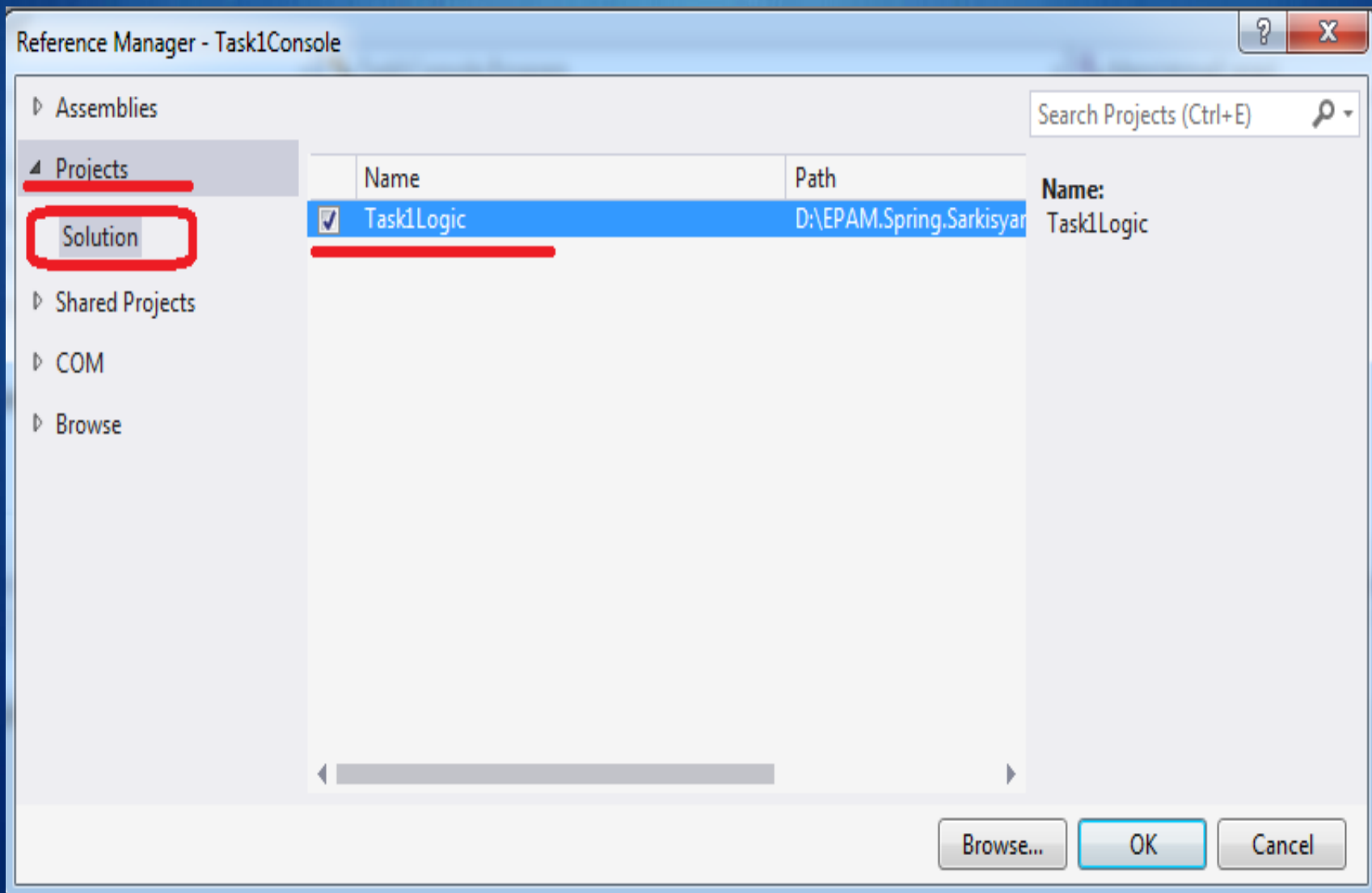


```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Task1Console
8 {
9     0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15         }
16 }
```

Связать Console с Library



Связать Console с Library



Связать Console с Library

EPAM.Spring.Sarkisyan.1 - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test ReSharper Analyze Window Help

Debug Any CPU Start

Program.cs X MaximumElementOfArray.cs*

C# Task1Console Task1Console.Program

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Task1Console
8 {
9     0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15         }
16     }
```

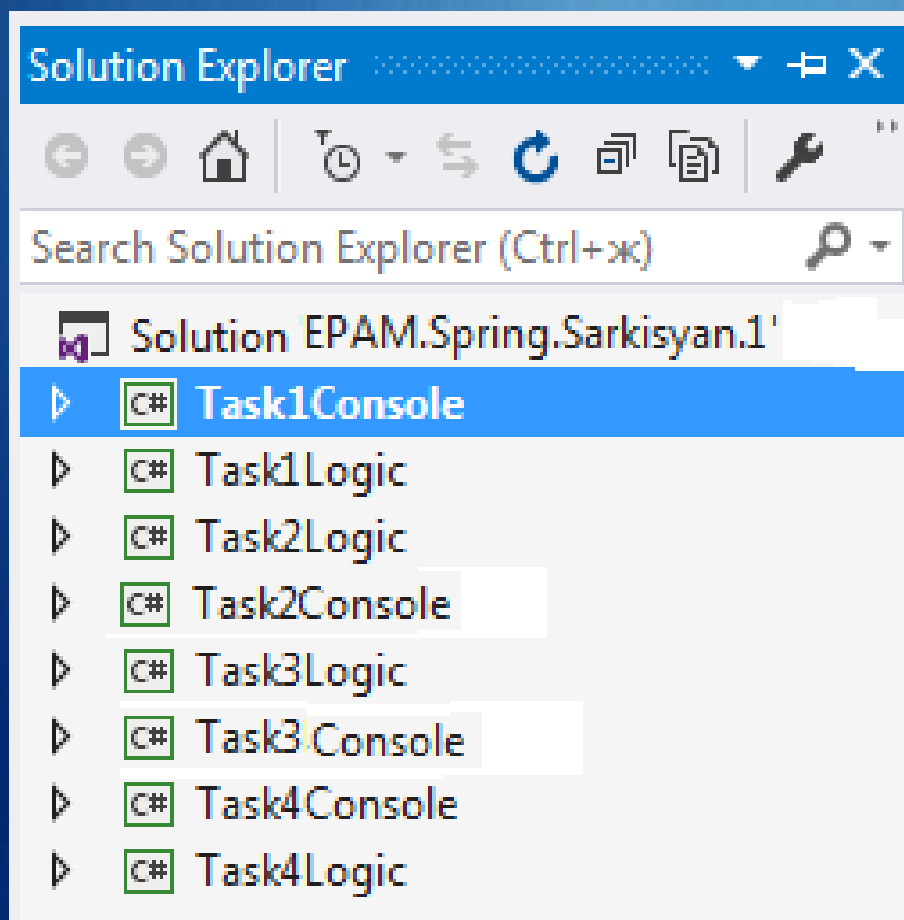
Solution Explorer

Search Solution Explorer (Ctrl+;)

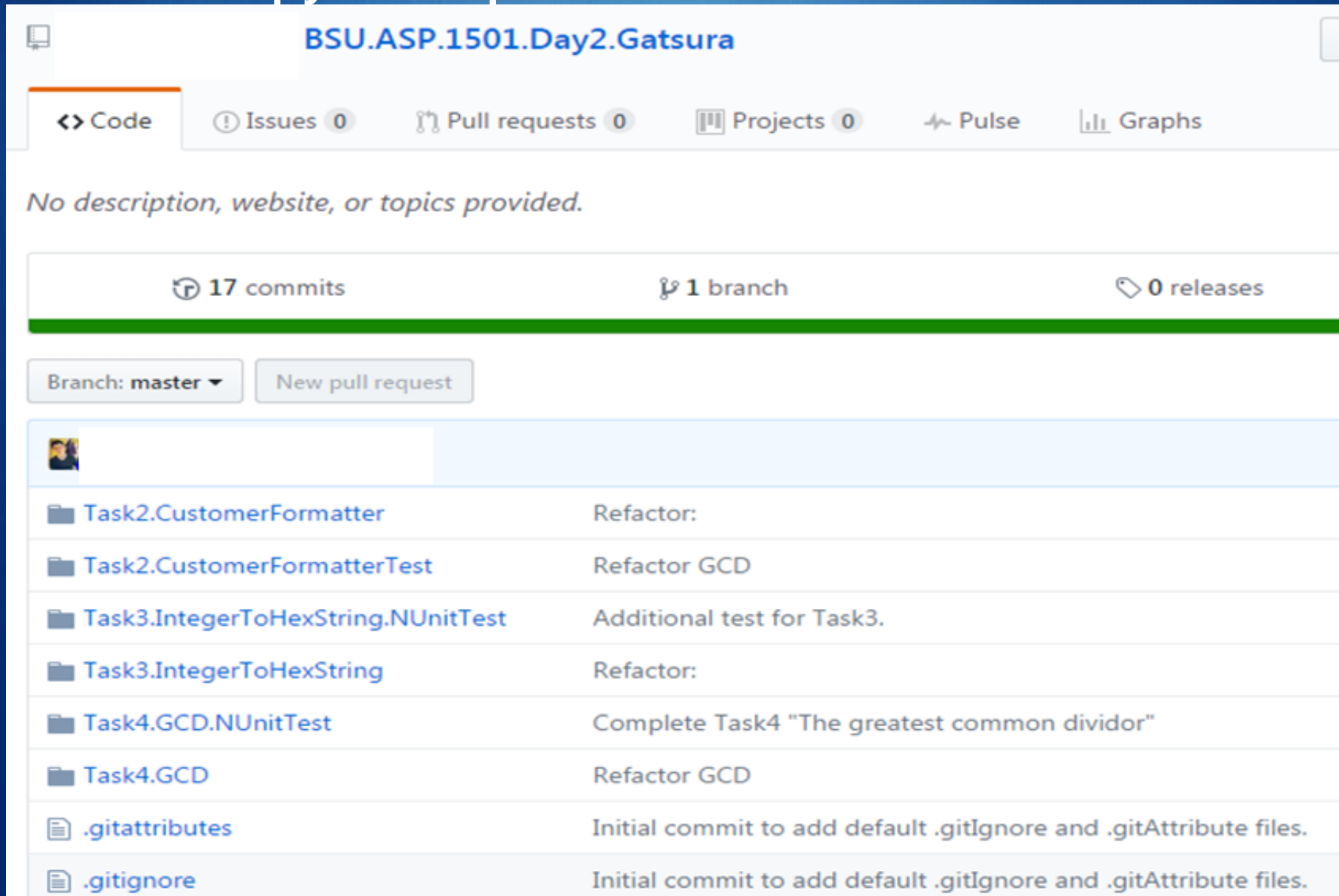
Solution 'EPAM.Spring.Sarkisyan.1' (2 projects)

- C# Task1Console
 - Properties
 - References
 - Analyzers
 - Microsoft.CSharp
 - System
 - System.Core
 - System.Data
 - System.Data.DataSetExtensions
 - System.Net.Http
 - System.Xml
 - System.Xml.Linq
 - Task1Logic
 - App.config
 - C# Program.cs

Все задания должны находится
в одном бланке решения



Инструментарий *GitHub*



The screenshot shows the GitHub interface for a repository named "BSU.ASP.1501.Day2.Gatsura". The repository has 17 commits, 1 branch, and 0 releases. The "Code" tab is selected, showing a list of files and folders. The repository description is "No description, website, or topics provided."

Repository: BSU.ASP.1501.Day2.Gatsura

Code Issues 0 Pull requests 0 Projects 0 Pulse Graphs

No description, website, or topics provided.

17 commits 1 branch 0 releases

Branch: master New pull request

Task2.CustomerFormatter	Refactor:
Task2.CustomerFormatterTest	Refactor GCD
Task3.IntegerToHexString.NUnitTest	Additional test for Task3.
Task3.IntegerToHexString	Refactor:
Task4.GCD.NUnitTest	Complete Task4 "The greatest common divisor"
Task4.GCD	Refactor GCD
.gitattributes	Initial commit to add default .gitIgnore and .gitAttribute files.
.gitignore	Initial commit to add default .gitIgnore and .gitAttribute files.

Требования к наименованиям

Стиль **Pascal case** применяется к **методам и свойствам** — каждое слово в имени метода начинается с верхнего регистра без символа нижнего подчеркивания. Глаголы.

```
public override string ToString()
{
    return string.Format("Author: {0} Title: {1}", Author, Title);
}
```

```
public double GetArea()
{
    throw new NotImplementedException();
}
```

Требования к наименованиям

Стиль *Camel case* (для **локальных переменных и полей типа**) – первое слово в нижнем регистре, а все остальные начинаются с буквы верхнего регистра. Существительные.

3 references

```
public class Book
{
    private int id;

    private string authorName;
}
```

```
public static double Sqrt(double number, int rank = 2, double precision = 0.001)
{
    int sign = GetSign(ref number, rank);

    double previousX = 1;
    double presentX;
    double tempPrecision = precision;
```

Требования к наименованиям

Upper case (для имен констант) все слова содержат буквы верхнего регистра.

Hungarian case — в начале в нижнем регистре сокращенно тип идентификатора а далее все слова начинаются с верхнего регистра. Только для интерфейсов и в generic тип.

<TKey, TValue>

IFormatProvider, ICustomFormatter

Группировать в регионы:

1. Поля и константы

Fiels and Constants

```
#region Fiels and Constants  
  
private int id;  
  
private string autorName;  
  
#endregion
```

2. Конструкторы

```
#region Constructor  
Oreferences  
public Book(string Author)  
{  
    this.AuthorName = Author;  
}  
#endregion
```

Группировать в регионы:

3. Свойства

```
#region Property
```

```
3 references
```

```
public int ID {  
    get  
    {  
        return id;  
    }  
    private set { }  
}
```

```
1 reference
```

```
public string AuthorName { get; set; }
```

```
#endregion
```

Группировать в регионы:

4. Методы

```
#region Public Methods

2 references
public UserEntity GetUserEntity(int id)
{
    return userRepository.GetById(id).ToBllUser();
}

2 references
public void DeleteUser(UserEntity user)
{
    userRepository.Delete(user.ToDalUser());
    unitOfWork.Commit();
}

#endregion

#region Private Methods

#endregion
```

5. События

Документирование приложений. XML комментарии

**Создание документированного компонента
включает несколько шагов:**

1. Создание компонента
2. Создание XML файла
3. Тестирование компонента в сценарии развертывания



Example

Документирование приложений.

XML комментарии

1. Создание компонента :

```
/// <summary>
/// Class describing a triangle on three sides
/// </summary>
2 references
public class TriangleNew
{
    /// <summary>
    /// side a
    /// </summary>
    private double a;
}
```


Документирование приложений.

XML комментарии

Тег	Назначение
<summary>	Предоставляет краткое описание. Для более подробного описания используются теги <remarks>.
<remarks>	Содержит подробное описание. Этот тег может содержать вложенные разделы (пункты), списки и другие типы тегов.
<example>	Предоставляет пример того, как метод, свойство или другой член библиотеки должен быть использован. Этот тег часто связано с использованием вложенных тегов <code>.
<code>	Указывает, что прилагаемый текст является кодом приложения.
<returns>	Документирует возвращаемое значение и тип метода.
<exception>	Документирует класс исключения (синтаксис проверяется компилятором)
<param>	Помечает параметр метода (синтаксис проверяется компилятором)
<value>	Описывает свойство

<https://msdn.microsoft.com/ru-ru/library/b2s063f7.aspx>

Документирование приложений.

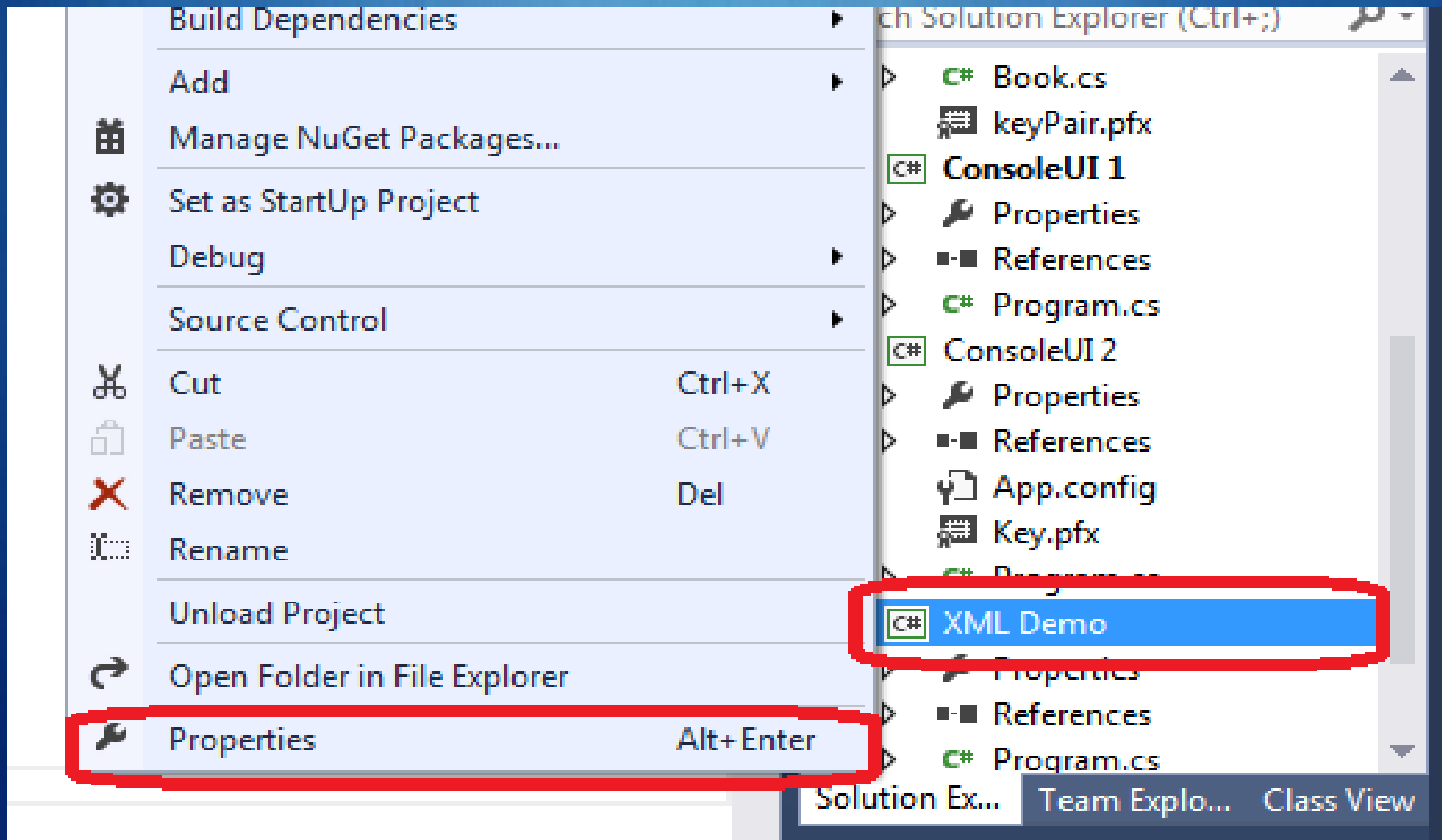
```
/// <summary>
/// constructor
/// </summary>
/// <param name="a">side a</param>
/// <param name="b">side b</param>
/// <param name="c">side c</param>
1 reference
public TriangleNew(double a, double b, double c)
{
    throw new NotImplementedException();
}

/// <summary>
/// Determination of the area of a triangle using the formula of Heron
/// </summary>
/// <returns>area of a triangle</returns>

1 reference
public double GetArea()
{
    throw new NotImplementedException();
}
```

Документирование приложений. XML комментарии

2. Создание XML файла



Документирование приложений. XML комментарии

The screenshot shows the 'Build' tab of the 'Build Properties' dialog in Visual Studio. The 'Application' section is selected in the left sidebar. The 'Configuration' is set to 'Active (Debug)' and the 'Platform' is set to 'Active (x86)'. The 'Output' section is expanded, showing the 'Output path' as 'bin\Debug\' and the 'XML documentation file' as 'bin\Debug\XML Demo.XML'. The 'Generate serialization assembly' is set to 'Auto'. The 'Advanced...' button is visible at the bottom right.

Application

Build

Build Events

Debug

Resources

Services

Settings

Reference Paths

Signing

Security

Publish

Configuration: Active (Debug) Platform: Active (x86)

Output

Output path: bin\Debug\ Browse...

☒ XML documentation file: bin\Debug\XML Demo.XML

☐ Register for COM interop






Generate serialization assembly: Auto

Advanced...

Specific warnings:

Документирование приложений.

XML комментарии

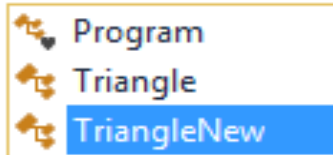
▼ \ASP.NET.1.Kravchuk.Day1\XML Demo\bin\Debug*.*					
Имя	Тип	Размер	Дата		
↑ [..]		<Папка>	01.03.2015 20:07		
 XML Demo	exe	5 120	01.03.2015 20:07		
 XML Demo	pdb	30 208	01.03.2015 20:07		
 XML Demo	XML	1 567	01.03.2015 20:07		
 XML Demo.vshost	exe	24 224	01.03.2015 20:07		
 XML Demo.vshos..	manifest	490	17.03.2010 22:39		

Документирование приложений.

XML комментарии

3. Тестирование компонента в сценарии развертывания

```
var t = new XML_Demo.|
```

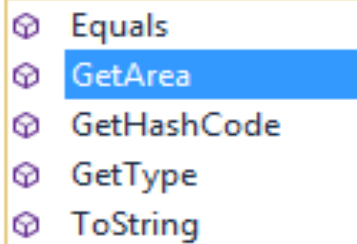


class XML_Demo.TriangleNew
Class describing a triangle on three sides

```
var t = new XML_Demo.TriangleNew()
```

TriangleNew.TriangleNew(double a, double b, double c)
constructor
a: side a

```
t.|
```



double TriangleNew.GetArea()
Determination of the area of a triangle using the formula of Heron

+375 29 500 27 50

+375 44 777 27 50

gayane.f.sarkisyan@gmail.com

Спасибо за внимание!