



OTUS

ОНЛАЙН ОБРАЗОВАНИЕ

Онлайн-образование



Меня хорошо видно && слышно?

Ставьте +, если все хорошо
Напишите в чат, если есть проблемы

Проверить, идет ли запись!





Code style от Microsoft

Панкрашов Дмитрий

Маршрут вебинара

Зачем нужно придерживаться
стиля?




Coding convention



DRY\DIY, YAGNI, KISS



Разбор примера

The background of the entire image is an aerial photograph of a city, likely New York City, showing a dense cluster of skyscrapers. A semi-transparent blue overlay covers the entire image. Overlaid on this blue background is a network of thin, light blue lines connecting various points, creating a geometric, web-like pattern that suggests digital connectivity or a network structure.

**Для чего существует
соглашение написания кода?**

Вводная

Работаем на крупный маркетплейс (Ozon, Wildberries).

Каждую минуту\секунду кто-то что-то покупает.

Можно вычислить прибыль в единицу времени -
тыс.руб\мин тыс.руб\сек

Вводная

Разработчик Вася задеплоил новые фичи.

Некоторые клиенты не могут оплатить заказ.

Уходят покупать в другой магазин.

Наш маркетплейс теряет прибыль.

Ужасающие кадры

```
1010
1011
1012     if(debug)Easy.WriteToLOG("will save ",sPath);
1013     try { empl.Person.Save(); } catch (Exception ex) { Easy.WriteToLOG("error save : "+ex.Message ,sPath); };
1014     if(debug)Easy.WriteToLOG(" was save ",sPath);
1015
1016
1017     if(worker.CurrentStatus == "Работа") { empl.Status = centrvd.NNK.Employee.Status.Active; }
1018     else { empl.Status = centrvd.NNK.Employee.Status.Closed; };
1019
1020     if(debug)Easy.WriteToLOG(" 2 ",sPath);
1021
1022     if(empl.Department != null)
1023     {
1024         if(empl.Department.Id1Ccentrvd != worker.IdDivision)
1025         { if(debug)Easy.WriteToLOG("2 1 ",sPath);
1026           var depart = centrvd.NNK.Departments.GetAll(x => Equals(x.Id1Ccentrvd, worker.IdDivision)).FirstOrDefault();
1027           empl.Department = depart;
1028         }
1029     }
1030     else
1031     { if(debug)Easy.WriteToLOG(" 22 ",sPath);
1032       var depart = centrvd.NNK.Departments.GetAll(x => Equals(x.Id1Ccentrvd, worker.IdDivision)).FirstOrDefault();
1033       empl.Department = depart;
1034     };
1035     if(debug)Easy.WriteToLOG(" 3 ",sPath);
1036
1037     if(empl.JobTitle != null)
1038     {
1039         if(empl.JobTitle.Id1CDemo475 != worker.IdPosition)
1040         { if(debug)Easy.WriteToLOG("3 1 ",sPath);
1041           var xmlTitle = staffInfo.Positions.Position.Find(x => Equals(x.Id, worker.IdPosition));
1042           xmlTitle.Name = TrimJobTitleName(xmlTitle.Name);
1043           var jobtitle = centrvd.NNK.JobTitles.GetAll(x => Equals(x.Name, xmlTitle.Name)).FirstOrDefault();
1044           empl.JobTitle = jobtitle;
1045         }
1046     }
1047     else
1048     { if(debug)Easy.WriteToLOG(" 32 ",sPath);
1049       var xmlTitle = staffInfo.Positions.Position.Find(x => Equals(x.Id, worker.IdPosition));
1050       xmlTitle.Name = TrimJobTitleName(xmlTitle.Name);
1051       var jobtitle = centrvd.NNK.JobTitles.GetAll(x => Equals(x.Name, xmlTitle.Name)).FirstOrDefault();
```

Форматирование курилщика

```
1010
1011
1012     if(debug)Easy.WriteToLOG("will save ",sPath);
1013     try { empl.Person.Save(); } catch (Exception ex) { Easy.WriteToLOG("error save : "+ex.Message ,sPath); };
1014     if(debug)Easy.WriteToLOG(" was save ",sPath);
1015
1016
1017     if(worker.CurrentStatus == "Работа") { empl.Status = centrvd.NNK.Employee.Status.Active; }
1018     else { empl.Status = centrvd.NNK.Employee.Status.Closed; };
1019
1020     if(debug)Easy.WriteToLOG(" 2 ",sPath);
1021
1022     if(empl.Department != null)
1023     {
1024         if(empl.Department.Id1Ccentrvd != worker.IdDivision)
1025         { if(debug)Easy.WriteToLOG("2 1 ",sPath);
1026           var depart = centrvd.NNK.Departments.GetAll(x => Equals(x.Id1Ccentrvd, worker.IdDivision)).FirstOrDefault();
1027           empl.Department = depart;
1028         }
1029     }
1030     else
1031     { if(debug)Easy.WriteToLOG(" 22 ",sPath);
1032       var depart = centrvd.NNK.Departments.GetAll(x => Equals(x.Id1Ccentrvd, worker.IdDivision)).FirstOrDefault();
1033       empl.Department = depart;
1034     };
1035     if(debug)Easy.WriteToLOG(" 3 ",sPath);
1036
1037     if(empl.JobTitle != null)
1038     {
1039         if(empl.JobTitle.Id1CDemo475 != worker.IdPosition)
1040         { if(debug)Easy.WriteToLOG("3 1 ",sPath);
1041           var xmlTitle = staffInfo.Positions.Position.Find(x => Equals(x.Id, worker.IdPosition));
1042           xmlTitle.Name = TrimJobTitleName(xmlTitle.Name);
1043           var jobtitle = centrvd.NNK.JobTitles.GetAll(x => Equals(x.Name, xmlTitle.Name)).FirstOrDefault();
1044           empl.JobTitle = jobtitle;
1045         }
1046     }
1047     else
1048     { if(debug)Easy.WriteToLOG(" 32 ",sPath);
1049       var xmlTitle = staffInfo.Positions.Position.Find(x => Equals(x.Id, worker.IdPosition));
1050       xmlTitle.Name = TrimJobTitleName(xmlTitle.Name);
1051       var jobtitle = centrvd.NNK.JobTitles.GetAll(x => Equals(x.Name, xmlTitle.Name)).FirstOrDefault();
```


Форматирование здорового человека

Kestrel HTTP Server

```
216 // Ungraceful shutdown
217 public void Dispose()
218 {
219     var cancelledTokenSource = new CancellationTokenSource();
220     cancelledTokenSource.Cancel();
221     StopAsync(cancelledTokenSource.Token).GetAwaiter().GetResult();
222 }
223
224 private void ValidateOptions()
225 {
226     Options.ConfigurationLoader?.Load();
227
228     if (Options.Limits.MaxRequestBufferSize.HasValue &&
229         Options.Limits.MaxRequestBufferSize < Options.Limits.MaxRequestLineSize)
230     {
231         throw new InvalidOperationException(
232             CoreStrings.FormatMaxRequestBufferSmallerThanRequestLineBuffer(Options.Limits.MaxRequestBufferSize.Value, Options.Limits.MaxRequestLineSize));
233     }
234
235     if (Options.Limits.MaxRequestBufferSize.HasValue &&
236         Options.Limits.MaxRequestBufferSize < Options.Limits.MaxRequestHeadersTotalSize)
237     {
238         throw new InvalidOperationException(
239             CoreStrings.FormatMaxRequestBufferSmallerThanRequestHeaderBuffer(Options.Limits.MaxRequestBufferSize.Value, Options.Limits.MaxRequestHeadersTotalSize));
240     }
241 }
```

The background of the slide is an aerial photograph of a city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer. On the left side of this layer, there is a network diagram consisting of white dots connected by thin white lines, forming a web-like structure. The text "Coding convention" is centered in the middle of the slide in a white, sans-serif font.

Coding convention

Форматирование

- **Отступы**
- **Одно выражение в строке**
- **Одно объявление в строке**
- **Разделять объявления методов и полей**
- **Использовать () для большей очевидности**

Комментирование

- **Лучший код тот - который документирует сам себя**
- **Не пытаться комментировать вообще вот
прямо всё**

Комментирование

- **Комментарии пишутся на отдельной строке**
- **Не после кода**
- **Между // и комментарием ставится пробел**
- **XML comments**

XML Комментарии

- Начинаются с ///
- Используют теги
- Например `<summary>Описание</summary>`

Комментирование

```
/// <summary>
/// Gets or sets the maximum size of the request buffer.
/// Defaults to 1,048,576 bytes (1 MB).
/// </summary>
/// <remarks>
/// When set to null, the size of the request buffer is unlimited.
/// </remarks>
public long? MaxRequestBufferSize
{
    get => _maxRequestBufferSize;
    set
    {
        if (value.HasValue && value.Value <= 0)
        {
            throw new ArgumentOutOfRangeException(nameof(value), CoreStrings.PositiveNumberOrNullRequired);
        }
        _maxRequestBufferSize = value;
    }
}
```


Именование

- **Pascal Case**

Первая буква слова - заглавная.

Пример: DataService, Enumerable

- **Camel Case**

Как Pascal, но самая первая буква - маленькая.

Пример: countRecords, firstName.

- **Snake Case**

В C# не используется.

Примеры: first_name, count_records

Именование

Pascal case:

- class
- record
- struct
- interface
- public members

Camel case:

- private\internal поля
(_fieldName)
- static private\static internal
- method parameters

Переменные

- Имя - осмысленное
- Отражающее назначение переменной
- Не слишком длинное

- i, j, k
- birthday
- countRecords



- peremennaya_cikla
- birthdayOfPersonGi..
- c



Переменные - типизация

- **var** - когда тип ясен по правой части
- Конкретный тип для неочевидных случаев

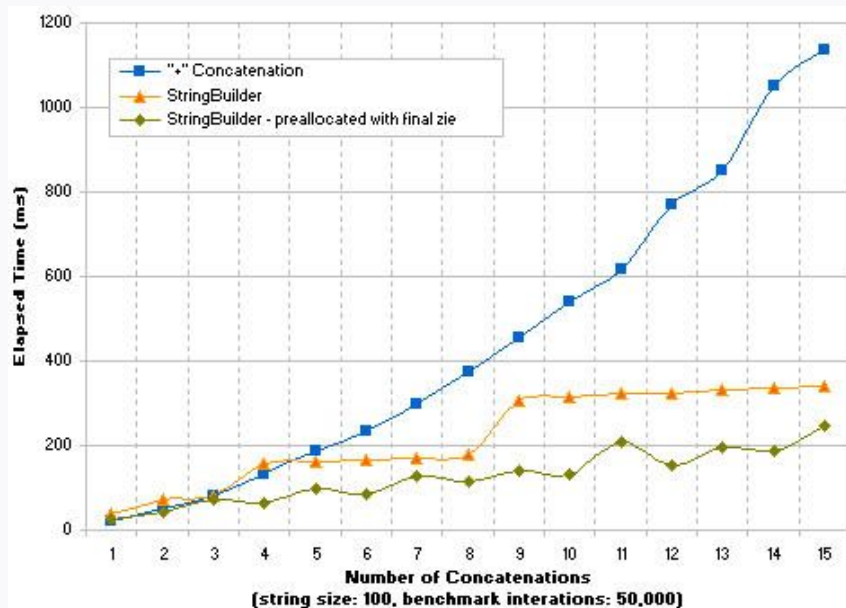
```
var var1 = "This is clearly a string.";
var var2 = 27;
```

```
int var3 = Convert.ToInt32(Console.ReadLine());
int var4 = ExampleClass.ResultSoFar();
```

```
var inputInt = Console.ReadLine();
Console.WriteLine(inputInt);
```

Строки

- Интерполяция для небольших объемов
- Для большого объема - StringBuilder



The background of the image is an aerial photograph of a city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer that features a white network pattern of interconnected dots and lines. The text is centered within this blue layer.

DRY\|DIE | KISS | YAGNI
boilerplate

DRY - Don't repeat yourself

Принцип призывает не повторяться при написании кода.

Все что вы пишете в проекте, должно быть определено только один раз.

DIE - Duplication is evil

Этот принцип заключается в том, что нужно избегать повторений одного и того же кода.

DRY \ DIE

```
private static void DoSomething(string name, string description, int age, string address =  
"Stockholm, Sweden", string format = "{0} is {1}, lives in {2}, age {3}")  
{  
    Console.WriteLine(format, name, description, address, age);  
}
```


KISS - keep it short simple *

Делайте вещи проще.

Порой наиболее правильное решение – это наиболее простая реализация задачи, в которой нет ничего лишнего.

YAGNI — You ain't gonna need it

Вам это не понадобится.

Все что не предусмотрено техническим заданием проекта, не должно быть в нем.

Чрезмерное усложнение кодовой базы из-за:

- Желания применить шаблон\подход\технологию
- Ориентации на переиспользование кода (пишем свой фреймворк)

Задача - написать калькулятор

Очевидное решение:

1. Определить операции (+, -, /, *)
2. Разбить строку на части:
 - a. Операнд
 - b. операция
3. Выполнить проверки
4. Выполнить выражение

Задача - написать калькулятор

Решение на все случаи жизни:

1. А только ли десятичные числа складываем?
2. А если по модулю \backslash в кольце?
3. А можем ли мы складывать
иррациональные числа?
4. Щас напишем!

Задача - написать калькулятор

```
interface ICalculatableEntity {
```

```
interface INumber {
```

```
interface IOperator {
```

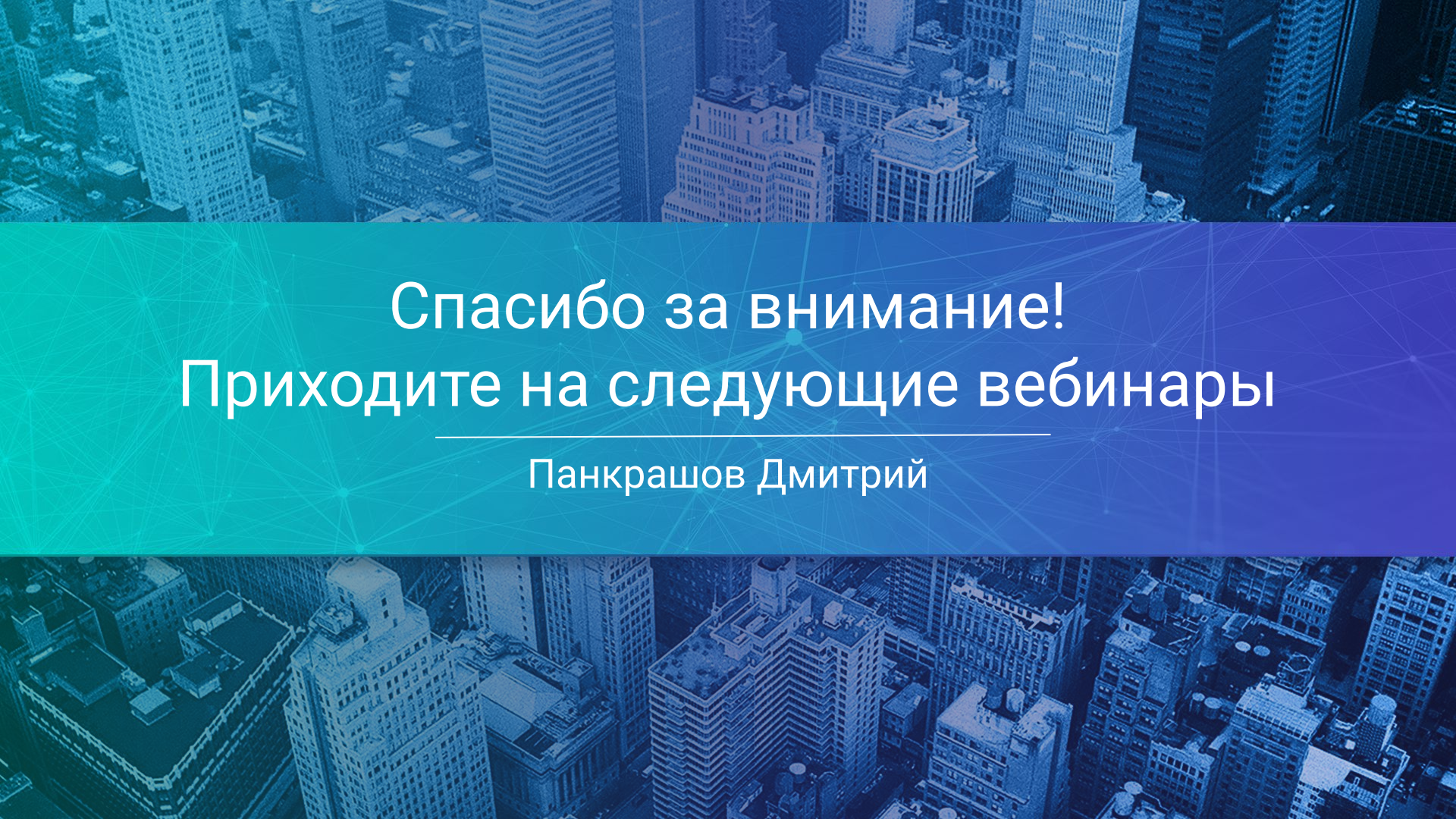
```
interface IExpression {
```

```
interface IResult {
```

Много кода - малый time to market

The image features a high-angle, aerial view of a dense urban skyline, likely New York City, with numerous skyscrapers and buildings. The entire image is overlaid with a semi-transparent blue layer. A network of thin, light blue lines connects various points across the blue area, creating a digital or technological aesthetic. The word "ПРАКТИКА" is centered in the middle of the image in a bold, white, sans-serif font.

ПРАКТИКА



Спасибо за внимание!
Приходите на следующие вебинары

Панкрашов Дмитрий