

Именуване на идентификатори

Именуване на променливи, методи,
параметри, свойства, константи,
класове, интерфейси и др.



Учителски екип

Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>



Съдържание

1. Общи съвети за именуване
2. Именуване на класове, типове и приложения
3. Именуване на методи и техните параметри
4. Именуване на променливи и константи
5. Други съвети за именуване



Общи съвети за именуване

- Винаги ползвайте **английски**
 - Английският е единственият език, използван от **ВСИЧКИ** софтуерни разработки
- Избягвайте **съкращенията**
 - Пример: **scrpCnt** vs. **scriptsCount**
- Избягвайте трудни за произнасяне имена
 - Пример: **dtbgRegExPtrn** vs. **dateTimeBulgarianRegExPattern**



Използвайте смислени имена

- Винаги предпочитайте **смислени** имена
 - Имена, които биха дали отговор на въпросите:
 - Какво прави този клас? Каква е целта на тази променлива? За какво се използва тази променлива / клас?
 - Примери:
 - **FactorialCalculator, studentsCount, Math.PI, configFileName, CreateReport**
 - Лоши примери:
 - **k, k2, k3, junk, f33, KJJ, button1, variable, temp, tmp, temp_var, something, someValue**



Имената трябва да са смислени в контекста

- Дали дадено име е **смислено** или не зависи от неговия **контекст** (неговия ограждащ тип)
- Примери за смислени имена:
 - **Generate()** в класа **LabyrinthGenerator**
 - **Find(string fileName)** в класа **FileFinder**
 - **Deposit(decimal amount)** в класа **Account**
- Примери за безсмислени имена:
 - **Generate()** в класа **Program**
 - **Find(string name)** в класа **Program**



Уж смислени имена

- Неопитни разработчици често използват "уж" смислени имена, които всъщност нямат смисъл
 - Лоши примери:
 - **Topic6Exercise12, LoopsExercise12, Problem7, OOPLecture_LastExercise**
 - Да, **Topic6Exercise12** показва, че това е решението на задача 12, но за какво е?
 - Сума на числа или игра Тетрис?
 - По-добри примери:
 - **MaximalNumbersSubsequence**



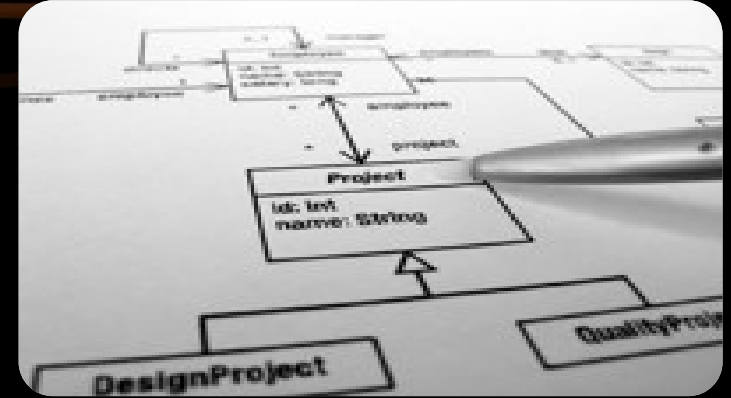
Именуване на класове и типове

- Именуване на типове (класове, структури, изброими типове и т.н.)
 - Използвайте **PascalCase** нотацията
 - Примери:
 - **RecursiveFactorialCalculator, TreeSet, XmlDocument, IEnumerable, Color, TreeNode, InvalidTransactionException, MainForm**
 - Лоши примери:
 - **recursiveFactorialCalculator, recursive_factorial_calculator, RECURSIVE_FACTORIAL_CALCULATOR**



Именуване на класове и структури

- Използвайте следните формати:
 - [Съществително]
 - [Прилагателно] + [Съществително]
- Примери:
 - **Student, FileSystem, BinaryTreeNode, Constants, MathUtils, CheckBox, Calendar**
- Лоши примери :
 - **Move, FindUsers, Fast, ExtremelyFast, Optimize, Check, FastFindInDatabase**



Именуване на интерфейси

- Допустими са следните формати:
 - 'I' + [Глагол] + 'able'
 - 'I' + [Съществително], 'I' + [Прилагателно] + [Съществително]
- Примери:
 - **IEnumerable, IFormattable, IDataReader, IList, IHttpModule, ICommandExecutor**
- Лоши примери :
 - **List, iFindUsers, IFast, IMemoryOptimize, Optimizer, FastFindInDatabase, CheckBox**



Именуване на изброими типове

- Допустими са няколко формата:
 - [Съществително] или [Глагол] или [Прилагателно]
- Използвайте еднакъв стил за всички членове



- Примери:
 - `enum Day { Monday, Tuesday, Wednesday, ... },`
`enum AppState { Running, Finished, ... },`
`enum WindowState { Normal, Maximized, ... }`



- Лоши примери :
 - `enum Color {red, green, blue, white},`
`enum PAGE_FORMAT {A4, A5, A3, LEGAL, ...}`



Именуване на специални класове

■ Атрибути

- Добавете окончанието '**Attribute**,
- Пример: **WebServiceAttribute**
- Лош пример : **WebService**



■ Колекции

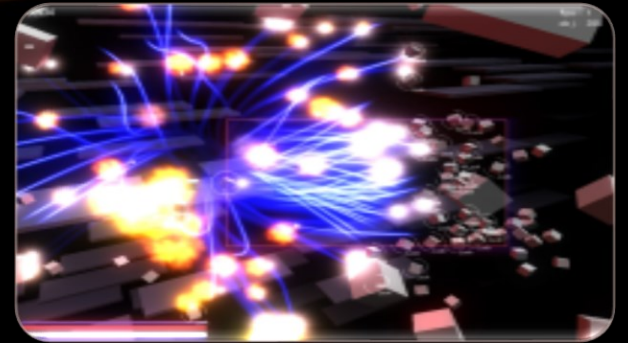
- Добавете окончанието '**Collection**'
- Пример: **StringsCollection**
- Лош пример: **ListOfStrings**



Именуване на специални класове (2)

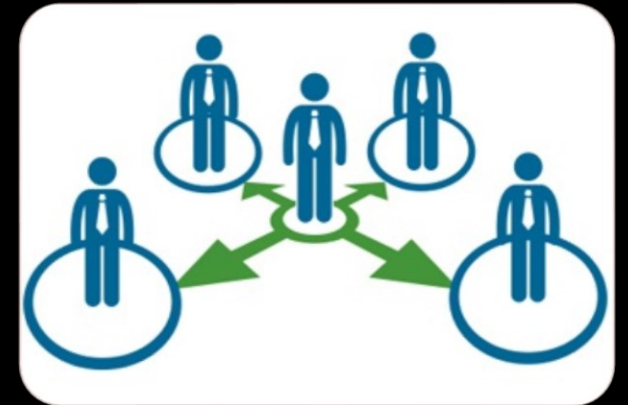
■ Изключения

- Добавете окончанието **'Exception'**
- Използвайте информативно име
- Пример: **FileNotFoundException**
- Лош пример: **FileNotFoundExceptionError**



■ Делегати

- Добавете окончанието **'Delegate'** или **'EventHandler'**
- Пример: **DownloadFinishedDelegate**
- Лош пример: **WakeUpNotification**



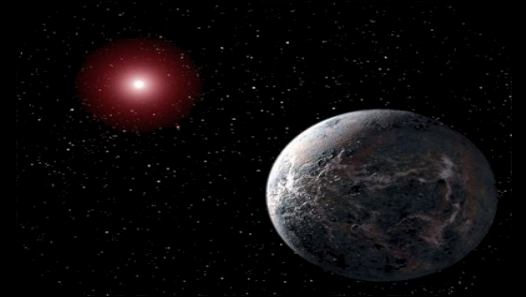
Дължина на имената на класовете

- Колко дълго трябва да е името на клас, структура и т.н?
 - Името да е толкова дълго, колкото е необходимо
 - Не съкращавайте имената, ако това ще ги направи неясни
- Примери: **FileNotFoundException**,
CustomerSupportNotificationService
- Лоши примери:
FNFException, **CustSuppNotifSrvc**



Именуване на пространства от имена

- Общи съвети
 - използвайте **PascalCase**
- Допустими са следните формати:
 - **Company.Product.Component...**
 - **Product.Component...**
- Пример:
 - **Microsoft.WinControls.GridView**
- Лоши примери:
 - **Microsoft_WinControlsGridView, Classes**



Именуване на папки с проекти

- Имената на папките с проекти трябва да следват името на пространството от имена
- Пример:
 - System
 - Collections
 - Generic
 - System.Collections.Generic...
- Лош пример:
 - `system_collections_generic, generic.src`



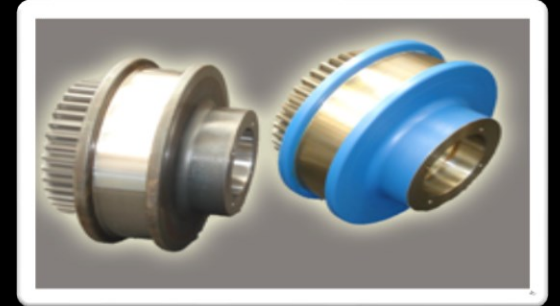
Именуване на файлове

- Файловете със програмен код трябва да имат имена, съответстващи на тяхното съдържание
 - Файлт, съдържащ клас `Student`, трябва да се казва **`Student.cs`**
- Примери:
 - **`StudentDAO.cs`, `Constants.cs`, `CryptographyAlgorithms.cs`**
- Лоши примери:
 - **`Program.cs`, `SourceCode.cs`, `Page1.aspx`, `WebApplication1.aspx`, `_d2.cs`, `Form1.xaml`**



Именуване на .NET асемблита

- Имената на .NET асемблитата трябва да следват основното пространство за имена с йерархията на своя клас
- Примери:
 - `Oracle.DataAccess.dll`
 - `Interop.CAPICOM.dll`
 - `Microsoft.WinControls.GridView.dll`
- Лоши примери:
 - `OracleDataAccess.dll`
 - `Microsoft_WinControlsGrid_View.dll`



Именуване на приложения

- Приложенията трябва да се именуват **смислено**

- Използвайте [Съществително] или [Прилагателно] + [Съществително]

- Използвайте **PascalCase**

- Примери:



- **BlogEngine**
- **NewsAggregatorService**

- Лоши примери:

- **ConsoleApplication4, WebSite2**
- **zadacha_14, online_shop_temp2**

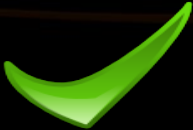




Именуване на методи

- Общи съвети
 - Използвайте **смислени** имена за методите
 - Имената трябва да отговарят на въпроса:
 - **Какво прави този метод?**
 - Ако не можете да намерите добро име на метода, помислете има ли ясна цел
- Примери: **FindStudent, LoadReport, Sine** 
- Лоши примери: **Method1, DoSomething, HandleStuff, SampleMethod, DirtyHack** 



Именуване на методи (2)

- Използвайте **PascalCase**
 - Пример: **LoadSettings** 
- Използвайте следните формати:
 - [Глагол], [Глагол] + [Съществително],
[Глагол] + [Прилагателно] + [Съществително]
- Примери: **Show, LoadSettingsFile, FindNodeByPattern, ToString, PrintList** 
- Лоши примери: **Student, Generator, counter, White, Approximation, mathUtils** 



Методи, които връщат стойност

- Методите, които връщат стойност, трябва да я описват


- Примери:

- **ConvertMetersToInches**, 

а не **MetersInches** или **Convert** или **ConvertUnit**

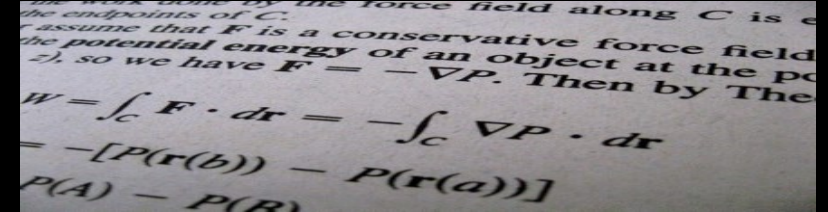


- **Meters2Inches** е допустимо 


- **CalculateSine** е добре, но **Sine** също е допустимо 

- Убедете се, че мерната единица е очевидна

- **MeasureFontInPixels** е за предпочитане пред **MeasureFont**



Единствена цел на методите

- Методите трябва да имат **една-единствена цел!**
 - Иначе не могат да се именуват правилно
 - Какво име ще дадете на метод, който прави годишен отчет на приходите, сваля обновления на софтуера от интернет и сканира системата за вируси?
 - **CreateAnnualIncomesReportDownloadUpdates AndScanForViruses**  звучи добре, а?
- Методи с няколко цели (weak cohesion) трудно се именуват
 - Вместо това трябва да се преработят

Консистентност при именуването на методите

- Спазвайте **еднотипност** в именуването в целия проект
 - **LoadFile**, **LoadImageFromFile**, **LoadSettings**, **LoadFont**, **LoadLibrary**, но не и **ReadTextFile**
- Използвайте **еднотипно** противоположностите:
 - **LoadLibrary** vs. **UnloadLibrary**, но не и **FreeHandle**
 - **OpenFile** vs. **CloseFile**, но не и **DeallocateResource**
 - **GetName** vs. **SetName**, но не и **AssignName**



Дължина на имената на методите

- Колко трябва да е дълго името на метода?
 - Името да е толкова дълго, колкото е необходимо
 - Не съкращавайте, вашата среда за разработка има свойство за автоматично допълване



- Примери:

- **LoadCustomerSupportNotificationService, CreateMonthlyAndAnnualIncomesReport**



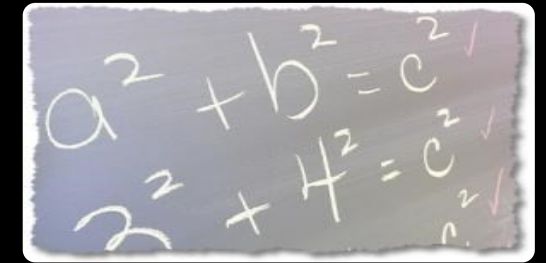
- Лоши примери:

- **LoadCustSuppSrvc, CreateMonthIncReport**



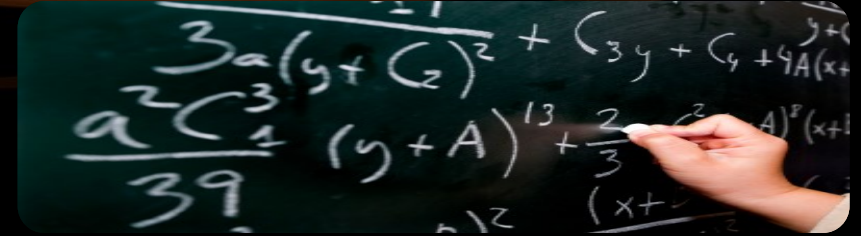
Именуване на параметрите на методите

- Имена на параметрите на методите
 - Предпочитан формат: [Съществително] или [Прилагателно] + [Съществително]
 - Да са в **camelCase** нотация
 - Да са **смислени**
 - Мерната единица да е очевидна
- Примери: **firstName, report, speedKmH, userList, fontSizeInPixels, font**
- Лоши примери: **p, p1, p2, populate, LastName, last_name, convertImage**



Именуване на променливи

- Имена на променливи
 - Да са в **camelCase** нотация
 - Предпочитан формат: [Съществително] или [Прилагателно] + [Съществително]
 - Да обясняват **целта** на **променливата**
 - Ако не можете да намерите добро име на променливата, помислете има ли една-единствена цел
 - Изключения: променливи с много малък обхват, напр. управляващата променлива на триредов цикъл for
 - Имената да са **образувани еднотипно** в целия проект



Именуване на променливи – пример

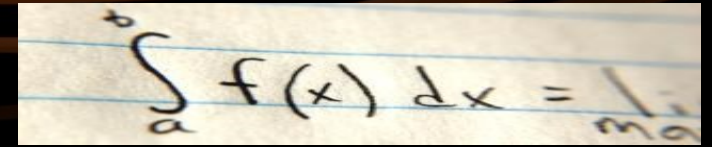
- Примери:

- firstName, report, config, userList, fontSize, maxSpeed, font, startIndex, endIndex, charsCount, configSettingsXml, dbConnection, createUserSqlCommand





- Лоши примери:

- foo, bar, p, p1, p2, populate, LastName, last_name, LAST_NAME, convertImage, moveMargin, MAXSpeed, _firstName, __temp, firstNameMiddleNameAndLastName






Още за именуването на променливи

- Името трябва да се отнася до проблема, който разрешаваме, а не до средствата, с които го правим
 - Съществителни от бизнес сферата на приложението са за предпочитане пред техническите термини
- Примери:
 - **accounts, customers, customerAddress, accountHolder, paymentPlan, vipPlayer** 
 - Лоши примери:
 - **paymentsPriorityQueue, playersArray, accountsLinkedList, customersHashtable** 

Именуване на булеви променливи

- Давайте на **булевите** променливи имена, които внушават **вярно** или **грешно**
- Използвайте положителни имена за булевите променливи
 - Лош пример:

```
if (!notFound) { ... }
```


 - Примери: **hasPendingPayment, customerFound, validAddress, positiveBalance, isPrime** 
 - Лоши примери: **notFound, findCustomerId, player, programStop, run, list, isUnsuccessful** 

Именуване на специални променливи

- Именуване на **броячи**
 - Нека да е еднотипно, напр. [Същ.и.] + 'Count'
 - Примери: **ticketsCount**, **customersCount**
- Променливи за **състояние**
 - Нека да е еднотипно, напр. [Същ.и.] + 'State'
 - Примери: **blogParseState**, **threadState**
- Променливи с **малък обхват** и малко време на живот
 - Напр. управляващи променливи в цикли
 - Може да се използват кратки имена, напр. **index**, **i**, **u**



Временни променливи

- Наистина ли мислите, че „временни“ променливи съществуват?
 - Всички променливи в програмата са временни, защото се използват само по време на изпълнение на програмата?
- Временните променливи винаги може да се именуват по-добре от **temp** или **tmp**:

```
// Swap a[i] and a[j]  
int temp = a[i];  
a[i] = a[j];  
a[j] = temp;
```



```
// Swap a[i] and a[j]  
int oldValue = a[i];  
a[i] = a[j];  
a[j] = oldValue;
```



Дължина на имената на променливите

- Колко трябва да е дълго името на променлива?
 - Зависи от **нейните обхват** и **време на живот**
 - По-„известните“ променливи да са с по-дълги и описателни имена
- Допустими примери за имена:

```
for (int i = 0; i < users.Length; i++)  
    if (i % 2 == 0)  
        sum += users[i].Weight;
```



```
class Student  
{  
    private string lastName;  
}
```



- Недопустими примери за имена :

```
class LinkedList  
{  
    public int flag { get; set; }  
}
```



```
class Student  
{  
    private int i;  
}
```



Именуване на константи

- Използвайте **PascalCase**
- Използвайте смислени имена, които описват стойността на константите
- Примери:

```
private const int ReadBufferSize = 8192;  
private const int DefaultFontSizeInPixels = 16;  
public static readonly PageSize DefaultPageSize = PageSize.A4;
```





- Лоши примери:

```
public const int Max = 512; // Max what? Apples or Oranges?  
public const int Buf256 = 256; // What about BUF256 = 1024?  
public const string Greater = "&gt;"; // GreaterThanHtmlEntity  
public const int FontSize = 16; // 16pt or 16px?  
public const PageSize PAGE_SIZE = PageSize.A4; // PascalCase convention
```



Никога не слагайте подвеждащи имена!

- Подвеждащото име е дори по-зле и от съвсем неясно име
- Пример:
 - Представете си метод, който пресмята сумата на всички елементи от масив
 - Добри имена: **Sum** или **CalculateSum** 
 - Лоши имена: **CalculateAverage** или **Max** или **CheckForNegativeNumber** 
 - Внимавайте с "copy/paste" програмирането!

Кое не е наред в този код?

```
FileStream fs = new FileStream(FILE_NAME, FileMode.CreateNew);
// Create the writer for data.
BinaryWriter w = new BinaryWriter(fs);
// Write data to Test.data.
for (int i = 0; i < 11; i++)
{
    w.Write( (int) i);
}
w.Close();
fs.Close();
// Create the reader for data.
fs = new FileStream(FILE_NAME, FileMode.Open, FileAccess.Read);
BinaryReader r = new BinaryReader(fs);
// Read data from Test.data.
for (int i = 0; i < 11; i++)
{
    Console.WriteLine(r.ReadInt32());
}
r.Close();
fs.Close();
```



Source: <http://msdn.microsoft.com/en-us/library/36b93480.aspx>

Обобщение

1. Практики при именуване
2. Общи съвети
 - Смислени имена (според контекста)
 - Една-единствена цел
 - Еднотипност на имената в цялото приложение
 - Избягвайте грешни и подвеждащи имена



Именуване на идентификатори



Въпроси?



Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni
Foundation

