

# Упражнения: Качествени методи

## Problem 1. .NET Reference Source

Прегледайте **поне три** примера в програмния код на .NET Reference (<http://referencesource.microsoft.com>). Разгледайте употребата на методи. Документирайте това, което откриете. Ако намерите неща, които не ви харесват, документируйте и тях.

Вижте долния пример:

Клас / Метод	Линк	Бележки
System.IO.File	<a href="http://referencesource.microsoft.com/#mscorlib/system/io/file.cs">http://referencesource.microsoft.com/#mscorlib/system/io/file.cs</a>	Методите са ясно написани, не са твърде дълги, именувани са добре...
System.IO.File	<a href="http://referencesource.microsoft.com/#mscorlib/system/io/file.cs">http://referencesource.microsoft.com/#mscorlib/system/io/file.cs</a>	Справяне с изключения – фреймуъркът (framework) използва помощни (helper) класове, за да хвърля изключения
...	...	...

Обърнете внимание, че някои от нещата, които не са ви харесали в кода, може да са били част от конвенциите за програмен код на Microsoft (например, Microsoft преди имаха обичая да пишат началната фигурна скоба на същия ред, а не на отделен). Въпреки това се чувствайте свободни да документирате всичко, което ви или не ви харесва в кода.

Някои неща, които да потърсите:

- Дължина на метода – да не е твърде дълъг
- Специализация и зависимост
- Едно и също ниво на абстракция
- Повторения на програмен код
- Правилна употреба на параметри – методът да не зависи от външни параметри и да не променя стойността на своите
- Обработка на изключенията

Допълнително можете да потърсите и неща, които вече сте научили:

- Форматиране на кода
- Ясно именуване – променливи, методи, пространства от имена и т.н.
- Документация и коментари
- Използване на променливи
- Вмъкване на цикли и условни команди
- Праволинеен код

## Problem 2. Преправяне на лош код

Намерете и документирайте **поне три** примера за лоша употреба на методи. Може да търсите в най-популярните source control хранилища, например **GitHub** (<https://github.com>) или **CodePlex** (<https://codeplex.com>), или пък може да погледнете нарочни примери за лош код: **GovnoKod** (<http://govnokod.ru>), **Bad Programming** (<http://badprogramming.com>) или **Reddit BadCode** (<http://reddit.com/r/badcode>).

Примерите може да са от едно и също парче код (ако е достатъчно дълго) или от различни такива. Може да използвате и някое от вече предоставените ви по-долу парчета програмен код.

Където е възможно, опитайте да преправите кода.

No.	Пример	Проблем	Решение
1	<a href="http://pastebin.com/tJyw8LjT">http://pastebin.com/tJyw8LjT</a>	Гигантски метод	Трудно е да се поправи. Да се извлекат няколко метода, да се използват по-смислени имена (readMsg -> методът не само чете съобщение, но и изпълнява много други задачи)
...	...	...	...

## Problem 3. Рекурсията винаги ли е необходима?

Напишете две реализации на алгоритъма за намиране на редицата на Фибоначи: една с цикъл и една с рекурсия. Сравнете изпълнението им. Можете да погледнете в класа **System.Diagnostics.Stopwatch** за начин да пресметнете времето за изпълнение на метода.

**Бележка:** Ако копирате рекурсивната реализация от Интернет, уверете се, че не използва масив, за да съхранява стойностите които вече са пресметнати. Това се прави понякога като оптимизация, но ние трябва да сравним изпълнението на чиста рекурсия vs. цикъл.

Подайте няколко параметъра и вижте колко време отнема на метода, за да приключи. Попълнете в таблицата по-долу. Напишете "hangs" („увисва“), ако изпълнението не приключи до 45-60 секунди.

Ако желаете, можете да подадете някои междинни стойности и да направите графика: поставете параметъра на оста X, а времето за изпълнение на оста Y.

Функция	n = 10	n = 50	n = 100	n = 1000	n = 10 000	n = 100 000	n = 1 000 000	n = 10 000 000
Цикъл								
Рекурсия								

По желание можете да повторите това, за да сравните цикличната и рекурсивната реализация на алгоритъма за пресмятане на факториал.

## Problem 4. Преправяне на методи

Използвайте VS **решението "Methods.sln"**.

- Преправете кода така, че да следва насоките за висококачествени методи.

- Уверете се, че обработвате грешките коректно: ако методите не могат да изпълнят това, което подсказва името им, **хвърлете изключение** (не връщайте грешен резултат).
- Подсигурете силна **специализация** и слаба **зависимост**, добро **именуване**, никакви странични ефекти и т.н.

## 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".



- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под **свободен лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).

