# Упражнения: Използване на величини и изрази

## Опишете и преработете лошия код

Намерете **поне пет** примера за лошо използване на променливи. 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 вече предоставеното ви парче код.

Документирайте тези примери. Опитайте да подобрите кода където е възможно. Можете да погледнете и примерните коментари по-долу:

**Code:** <http://pastebin.com/RGAyFyGx>

|  |  |  |
| --- | --- | --- |
| **No.** | **Лоша употреба** | **Предложение за подобрение** |
| 1 | Settings\_Load(): bad naming: app1, …, app6; ico1, …, ico6 | Да се преименуват променливите с по-смислени имена.  Бележка: Това парче код може да не е достатъчно, за да се ориентирате и да изберете адекватни по-добри имена |
| 2 | Public field Properties.Settings s | Да се махне public полето; или да се капсулира в свойство, или да се подаде като параметър на метод където трябва |
| 3 | … | … |

## Преработете ваш програмен код

Намерете някоя от своите изпитни задачи или екипни работи и **преправете употребата на променливи** в нея.

Някои неща, които да погледнете:

* Имена на променливите – неописателни, безсмислени имена, подвеждащи имена
* Обхват и време за използване на променливите – твърде голям обхват, променливи инициализирани далеч преди първата си употреба
* Време за живот на променливите – ненужно дълъг живот на променливата; обмислете да промените някои полета в параметри на методи
* Глобални променливи – обикновено static променливи, които могат да се променят от твърде много места
* Видимост на променливите – уверете се, че сте подсигурили най-ниската възможна видимост
* Специализация и зависимост – обмислете намаляването на видимостта, премахването на глобални променливи и т.н. за да намалите зависимостта между методи и класове
* Форматиране на кода: уверете се, че кодът е добре форматиран и лесно четим