IPS Instituto Politécnico de Setúbal
Escola Superior de Tecnologia de Setúbal

PA 2021/22

Programação Avançada

Enunciado de Laboratório

9

Refactoring - (tutorial)

Objetivos:

- Identificação de *bad smells* no código
- Criação de testes unitários

Introdução:

O processo de *refactoring* é o processo de alterar um programa de *software*, através da melhoria da sua estrutura interna (desenho), sem alterar o seu comportamento. Este processo melhora o desenho do programa de software, torna o código mais eficiente, mais fácil de manter e – por conseguinte – facilita a compreensão do código, pois na maioria dos ambientes de desenvolvimento de *software*, outras pessoas que não desenvolveram o código terão que o ler.

Neste laboratório pretende-se aplicar as devidas técnicas de *refactoring* aos diversos tipos de *bad smells* exemplificados, a fim de promover o desenvolvimento de código "limpo". Aliando a literatura teórico-prática, sugere-se a leitura do conteúdo existente em https://refactoring.guru/refactoring/smells.

Comece por clonar o projeto existente no GitHub no seguinte endereço:

https://github.com/estsetubal-pa-geral/Lab9Template.git

O projeto contém uma implementação – incompleta - do jogo de xadrez, apenas com torres, bispos e peões. Xadrez é um jogo de tabuleiro, disputado entre dois jogadores, cujo objetivo é conquistar o "rei" do seu adversário. Neste *template* dada a sua simplicidade, as regras de validação ao nível de movimentação das peças não estão contempladas.

Nível I

A classe **Game** contém *bad smells* do tipo *dead code*.

1. Efetue o devido refactoring à classe, mencionando (via JavaDoc) o tipo de técnica(s) utilizada(s).

2. Verifique que consegue executar o *main()*, existente na classe **Main**.

Nível II

A classe **Game** contém um bad smell do tipo *duplicate code*.

- Crie teste unitários, para validar a correção dos métodos moveWhite*()* *e* moveBlack*()*

Nível III

a) Efetue o devido refactoring aos métodos moveWhite() e moveBlack(), mencionando (via JavaDoc) o tipo de técnica(s) utilizada(s).
b) Verifique que os testes unitários criados anteriormente não tiveram qualquer impacto no comportamento do jogo, após o refactoring dos dois métodos anteriores.

Nível IV

A classe **Piece** contém um bad smell do tipo *switch statement*.

- Crie teste(s) unitário(s) que valide(m) a correção do método *getValue()*.

Nível V

a) Efetue o devido refactoring à classe, mencionando (via JavaDoc) o tipo de técnica(s) utilizada(s).

b) Verifique que os testes unitários criados anteriormente não tiveram qualquer impacto no comportamento do método *getValue()*, após o refactoring.