# Relatório de code smells

## Refactoring do Projeto

Relativamente a code smells no nosso projeto, decidimos utilizar o plug-in JDeoderant, este plug-in auxiliou-nos a detetar e remover alguns code smells tais como ‘Feature Envy’, ‘Is Long Method’ e ‘Is God Class’. Para além deste plug-in ainda utilizámos o menu ‘Refactor’ do eclipse, este menu não deteta code smells, apenas ajuda a remover potenciais code smells que tenhamos detetado manualmente por exemplo: um método/construtor com demasiados parâmetros, constantes repetidas no código, métodos demasiado grandes, etc. Neste menu, as opções mais utilizadas foram a extração de métodos com o objetivo de diminuir o tamanho dos métodos, extração de constantes e alteração da assinatura dos métodos.

Após as otimizações feitas ficámos com apenas um code smell, a classe ‘Rule’ tem no seu construtor 11 parâmetros, decidimos não alterar os parâmetros pois considerámos que não faria sentido dividir os parâmetros em objetos, faria mais sentido uma regra ser definida por todos esses parâmetros. Para além deste code smell resolvemos todos os outros. Como tivemos auxílio de uma biblioteca chamada SWT, temos um pacote que nos ajudou a criar as interfaces gráficas, este pacote contém um code smell que não resolvemos porque não faria sentido, uma vez que não faz parte do que fizemos.

Por fim, ao utilizar o plug-in JDeoderant, apesar de termos muitas sugestões de refactoring de ‘feature envy’, ‘god class’ e ‘long method’ não implementámos todas, apenas as sugestões que achámos que fariam sentido segundo a estrutura do projeto. Foram gerados ficheiros ‘.txt’ com sugestões de refactoring não implementadas.

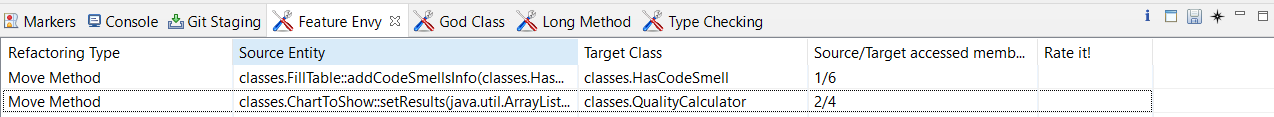


Figura 2 - Sugestões de refactoring para 'feature envy' não implementadas

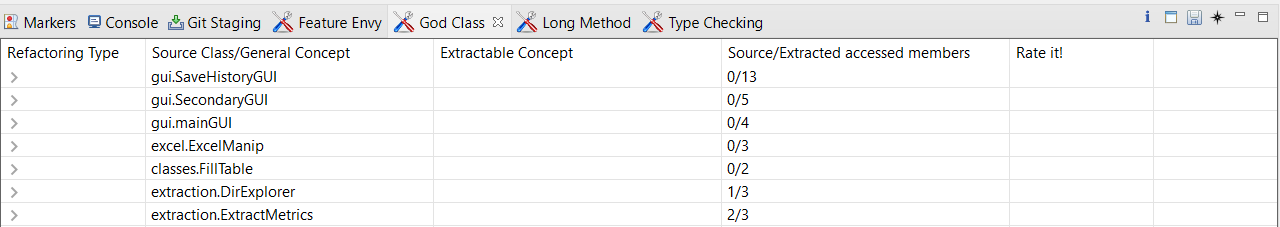


Figura 3 - Sugestões de refactoring para 'god class' não implementadas

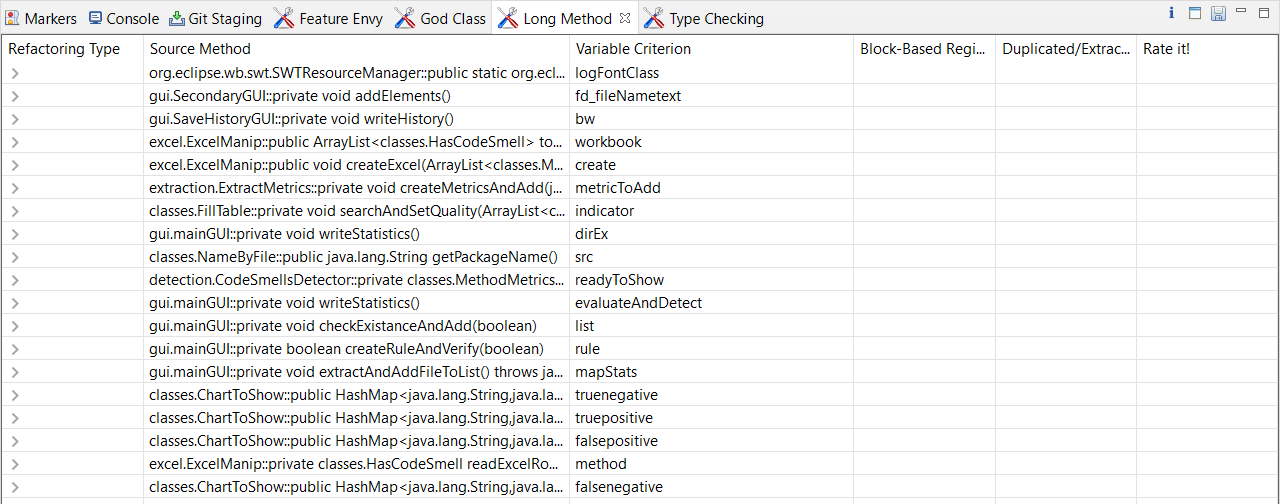


Figura 3 - Sugestões de refactoring para 'long method' não implementadas