**Aluno:**

Francisco Parrinha, n58360

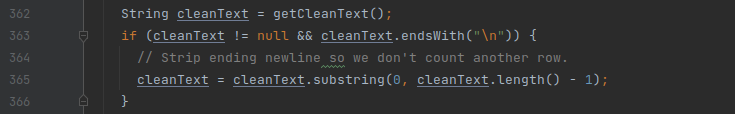
**Code Smells**

1. Long Method:

* Método *computePrefHeigh*” em: *ganttproject/src/main/java/biz.gantproject/lib/fx/treetable/LabeledSkinBase.java*

O que me fez reparar neste code smell foi o facto do método ser demasiado extenso. De seguida, reparei que o bloco de código (linhas 362-366) apresentava uma operação auxiliar ao método. Esta operação é para repartir *strings*.

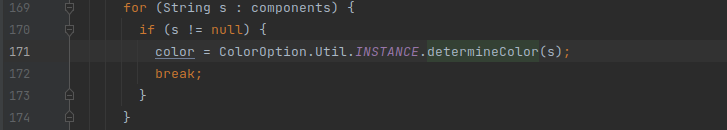
Sendo auxiliar, o bloco de código pode ser retirado e colocado numa outra função à parte, por exemplo *getStrippedString*. Este é o seguinte código:



* Método *parse* em: *biz.ganttproject.core/src/main/java/biz/ganttproject/core/chart/render/RectangleRenderer.java*

A função *parse*, apesar de não ser muito grande, é mais comprida do que o necessário. A atribuição de cor é uma operação auxiliar ao método e pode ser refactorizada.

O bloco de código (linhas 169 - 174) apresentado pode ser retirado e colocado numa outra função, *setColor*, por exemplo. Isto modulariza melhor o código e permite que atribuições de cor a coleções possam ser reutilizadas. Eis o código em questão:

**

2. Comments:

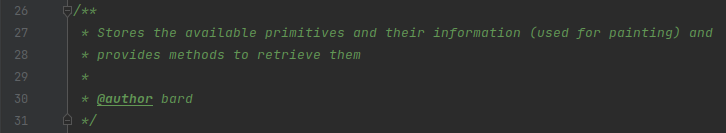
* Classe *Canvas* em: *biz.ganttproject.core/src/main/java/biz/ganttproject/core/chart/canvas/*

A classe apresenta uma série de *nested classes*. Toda a classe *Canvas* não apresenta comentários (nem *Javadoc*) a não ser o que está escrito entre as linhas 26 e 31, isto incluí as suas *nested classes.* Por isso, a descoberta deste *code smell* foi relativamente rápida.

Nenhum método e nenhuma *nested class* está comentada, dificultando a compreensão dos mesmos. Algumas linhas estão comentadas e a falta de comentários também dificulta perceber o porquê dessas linhas não serem úteis.

Para refactorizar, basta criar o *Javadoc* dos métodos e dar uma breve explicação das *nested classes*. Alguns exemplos da ausência do *code smell*:

1. O único comentário:



1. Alguns exemplos de ausência de comentários

