**Student**

Francisco Parrinha – 58360

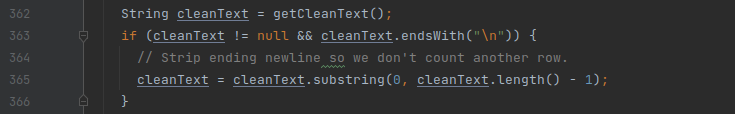
**Code Smells**

1. Long Method:

* Method *computePrefHeight*” in: *ganttproject/src/main/java/biz.gantproject/lib/fx/treetable/LabeledSkinBase.java*

The method starts at line number 350 and ends at number 397, containing 47 lines. There are several refactoring possibilities. The code within lines number 362 and 366 serves as an auxiliary operation to the method. This operation is to strip strings.

The following block of code could be removed from this function to a new *getStrippedString* method, and later be called on the original *computePrefHeight* function. The code to be refactored:



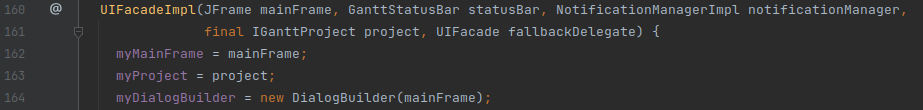
2. Long Parameter List

* Class *UIFacadeImpl* em: *ganttproject/src/main/java/net/sourceforge/ganttproject/UIFacadeImpl.java*

The following class has too many parameters (5) in its constructor, making the code much harder to comprehend. To refactor, it is necessary to create a *nested* builder class that has several constructors, simplifying the parameter list and making it easier to detect bugs. This builder class would then be dependent on the original class.

The constructors of this new builder would then be used when creating new instances of UIFacadeImpl class.

The code smell:



3. Comments:

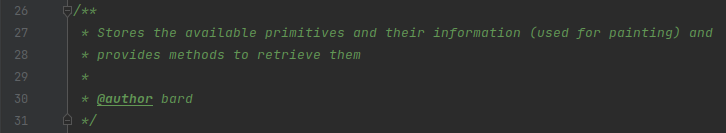
* Classe *Canvas* em: *biz.ganttproject.core/src/main/java/biz/ganttproject/core/chart/canvas/*

This class contains several nested classes The entire *Canvas* class only contains one comment (line 26 -31) and it does not have any *Javadoc* written excluding this one.

All methods and nested classes are to be commented out, making the code understanding much harder.

Creating the Javadoc is enough to refactor this code-smell. Here are some examples of this anti-pattern

1. The only comment:



1. Some examples of the lack of code: