DP2-1920-GI-02: Sprint 4 June 1, 2020

**REFACTORING AND SONARCLOUD**

In part I of this document, refactorings (improvements in the quality on the code) have been carried out based on a detected code smell. In part II, improvements in the quality of the code are evidenced by reducing code smell and we can see using SonarCloud screenshots.

**PART I: REFACTORING**

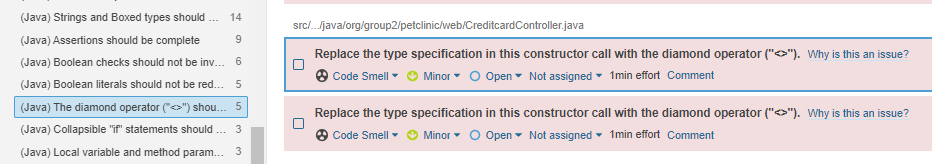
**REFACTORING 1**

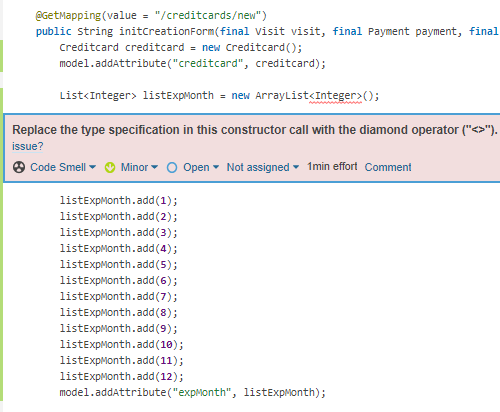
Typo of code smell:

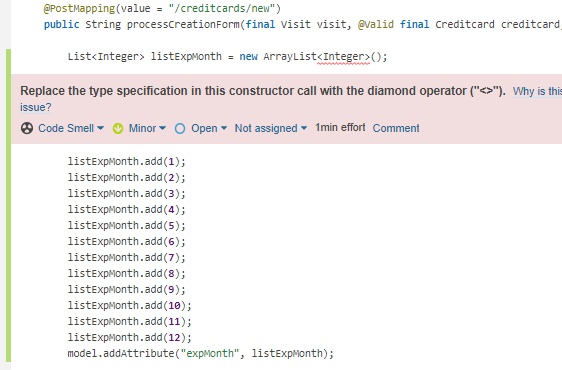
Composition Methods – Extractin Method – Duplicated Code.

Problem:

It was detected an error in the CreditcardController.java class twice. It has been detected twice because the code is duplicated in the get and post methods. Both request want to perform the same functionality: create a list of number equivalent to months to be used as a dropdown.





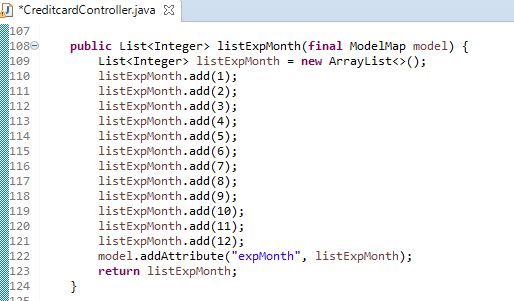


Solution:

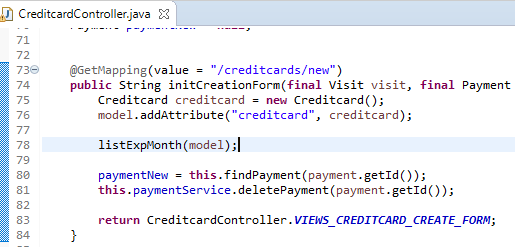
The solution has been to create a public method that collects the duplicate code. In this method we have corrected the error that SonarCloud marked us: put a diamond operator. Then this method will be called in the corresponding place of the get and post method.

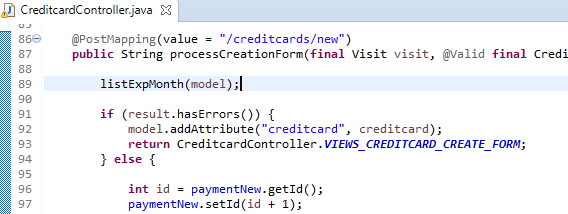
This reduces the percentage of duplicate code that exists in our application.

The new public method:



Calls to that method:





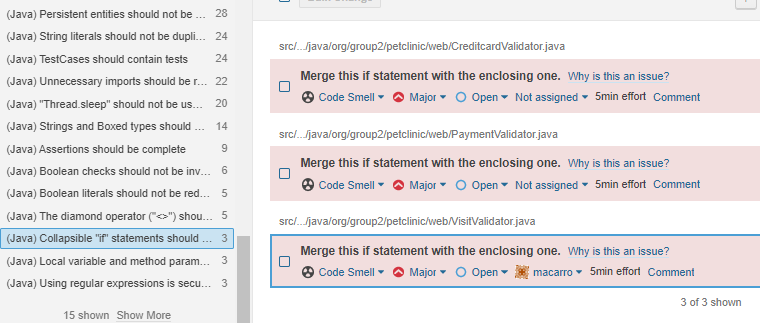
**REFACTORING 2**

Typo of code smell:

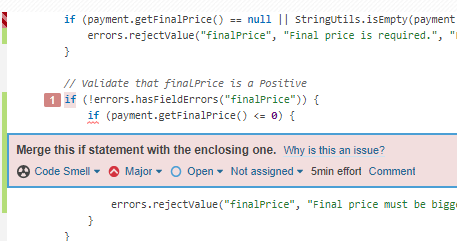
Simplifying Conditionals – Long method.

Problem:

A code smell has been detected in wich there are two if statemets in a row, so it’s unnecesary. This repeats three times in differents classes.



Example of how it looks in error inside a class:

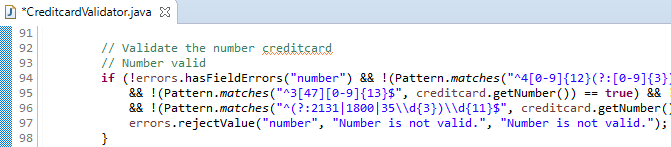


Solution:

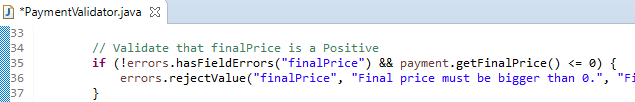
For each of the classes where it was in error: one if statement has been removed and his condition has been included in the other if statement.

This creates a cleaner and more summarized code. Shorten the method.

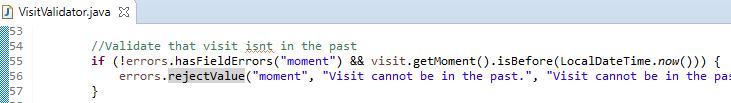
For CreditcardValidator.java:



For PaymentValidator.java:



For VisitValidator.java:



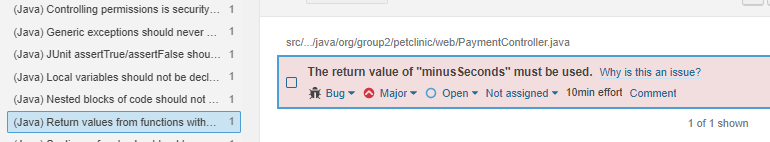
**REFACTORING 3**

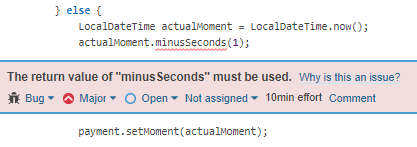
Typo of code smell:

Organizing Data – Temporary Field.

Problem:

A code smell has been detected in wich we see that a method is called to perform an operation but the value it returns is not stored in any variable. This can cause problems that the method does not do the operation that we want to use later.

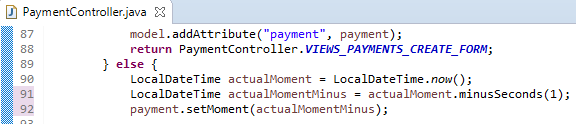




Solution:

It has been solved by storing in an variable the operation that this method performs. Then this variable is the one used.

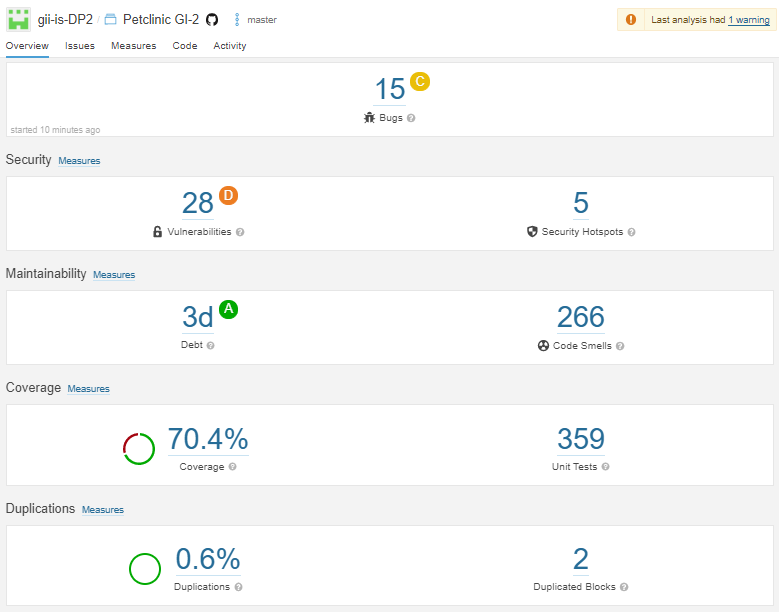
This makes the value of the variable available at any time and we make sure that the operation is carried out.



**PART II: SONARCLOUD**

**SONAR CLOUD BEFORE REFACTORINGS**

You can see that it is made up of **266** code smells.



**SONAR CLOUD AFTER REFACTORINGS**

You can see that it is made up of **261** code smells.

So we have improved our code quality by correcting 5 code smells.

