Refactoring

Kwinten Missiaen, Steven Thuriot, Koen Van den dries, Bart Vangeneugden

Methodologies for Design of Software

Tom Holvoet
Mario Cruz Torres

Academiejaar: 2009 - 2010

Refactoring

Kwinten Missiaen, Steven Thuriot, Koen Van den dries, Bart Vangeneugden

Methodologies for Design of Software

Tom Holvoet
Mario Cruz Torres

Academiejaar: 2009 - 2010

Contents

1 Controllers

The refactoring of the controllers wasn't too much work as these were quite good as is. In other words, not too many bad smells were detected when reviewing them.

The biggest problem here were variables that were given unclear names. This might cause uncertainties or problems when you, or especially when another member of the team, starts reworking the existing code. This also makes the need to read the Javadocs of said methods smaller or in some cases even unneeded. As a result, the work that needs to be done, will be done sooner and will also frustrate the programmer less. As you can see, even something as small as this, is worth the trouble of doing right from the start. The controllers that had the most problems with this were the repository manager and the focus work classes.

The next problem were the usage of a few switch cases throughout the code. The focus factory has a switch case to determine which type of focus the user wants to create. The refactoring book clearly states that switch cases cause a bad smell. However, being a factory, there is no good way around this. Because this is a factory, it also solves the bad smell of the switch cause itself. The problem with switch cases is that these may appear more than once in your code. Since the whole point of the factory is to take care of the creation of the wanted focus object, this switch case will never appear again throughout the model. In other words, this piece of code should not be sensitive to duplicated code. The only other place this appears is in the GUI. The case in the GUI has therefore been adjusted in such a way that it now asks the factory for the possibilities and prints these on the screen, rather than the hardcoded print it used to have. This takes away the control the GUI has over this and gives it back to the factory. Because of this, the only occurrence of the switch case can be found here. We have thought about using polymorphism to solve this problem. This, however, was not the way to go. As a result of this, the GUI would become massive and almost impossible to change in the future. This is, of course, even worse than having a switch case there.

The last problem was also found in the focus factory. According to the type you want, a certain amount of parameters need to be passed. Before, the method asked to pass all the parameters. This result in the programmer having to pass more parameters than needed in some cases, some of them being null objects. The method was reworked so an array is passed instead. The method is now much cleaner and easier to use.