

Three bad smells

Duplicated Code

Der erste Schritt vom „Refactoring“ war die Methode „amountFor“ aus der Methode „Statement“ zu extrahieren. Nachdem die Methode erstellt wurde, musste die Switch-Anweisung aus der Methode „Statement“ entfernt werden, sonst existiert die Methode mehrmals und der Compiler kann nicht fehlerfrei arbeiten. Außerdem kann später das Template, welches für das Projekt erstellt wurde fehlerfrei arbeiten, denn sonst würde dies zu einem Fehler führen.

Long Methods

Anfangs war die Methode „Statement“ in der Klasse „Customer“ sehr lang und dadurch sehr unübersichtlich. Was dazu führen könnte, dass Fremde anfangs nicht wissen, wofür der Code zuständig ist.

Um diese Methode übersichtlicher zu machen wurde schon als ersten Schritt die Switch-Anweisung ausgelagert. Durch den Schritt „Extract Method“ wurde diese Methode etwas angenehmer beim Lesen des Codes.

Eine weitere Maßnahme war „Replace Temp with Query“. Dadurch können weitere Methoden ausgelagert werden und eine Methode schlanker zu gestalten, wie bei den beiden Methoden „totalAmount“ und „frequentRenterPoints“.

Lazy Class / Method

In diesem Fall geht es darum, dass jede Methode und auch Klasse Ressourcen kostet. Ist eine Klasse oder eine Methode diesen Aufwand nicht wert, so wird diese gelöscht und/oder mit einer anderen zusammengelegt.

Dies wurde zu anfangs gemacht, als man 2 Methoden nutzte, um den Preis für die Miete des Filmes zu berechnen. Gegen Ende wurden diese durch mehrmaliges nutzen von „Move Methode“ zu der Klasse gebracht, die alle nötigen Parameter beinhaltet. Dies resultierte darin, dass nur noch eine Methode gebraucht wurde.