**Design Pattern – 3**

**Strategy**

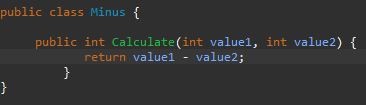
In Strategy pattern, objects are created and they might have multiple strategies and algoritms. The strategy object changes the executing algorithm of the context object, depending on which algorithm is chosen.

**Code Snippet:**

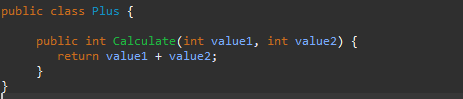
Interface:



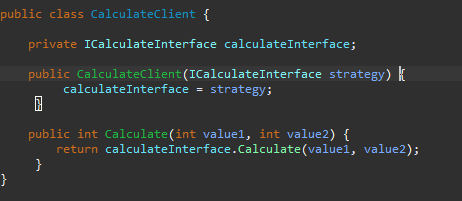
Algorithm Minus:

****

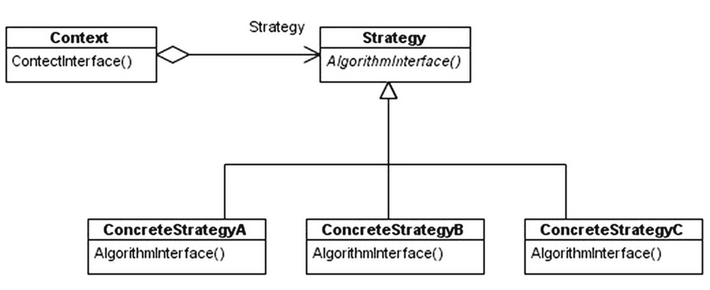
Algorithm Plus:

****

Client Class:

****

**Structure:**

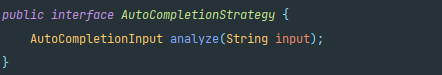
****

**Code Location:**

The classes are in the folloying path:

“*src-gen\main\java\org\jabref\gui\autocompleter*”

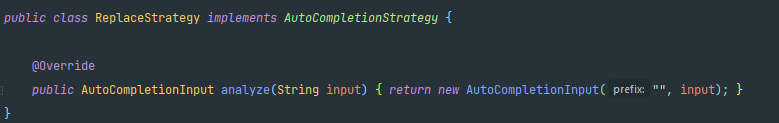
The interface is created with the method analyze():



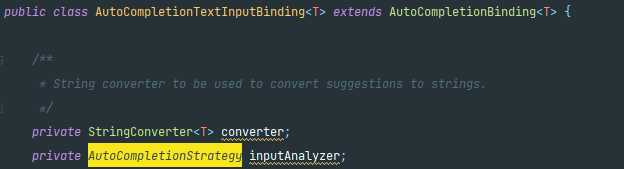
Two different classes with the same interface implemente the method analize() in different ways (with different algorithms):

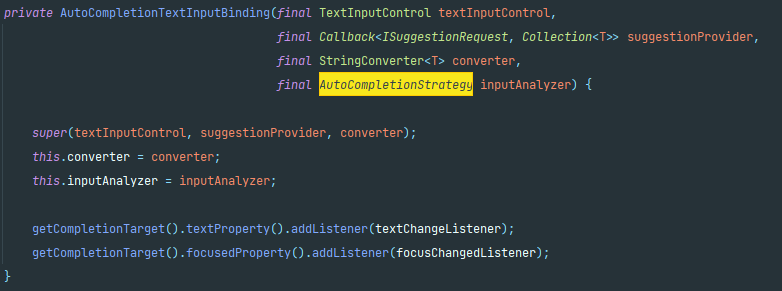
First algorithm:



**** Second algorithm:

Client Class that chose any algorithm to use beacause of the interface:

****

****

**Justification:**

As we can see in the images, there is a variable (inputAnalyzer) that use the interface type (AutoCompletionStrategy). It means that this variable can save any algorithm, depending on which one it recives.