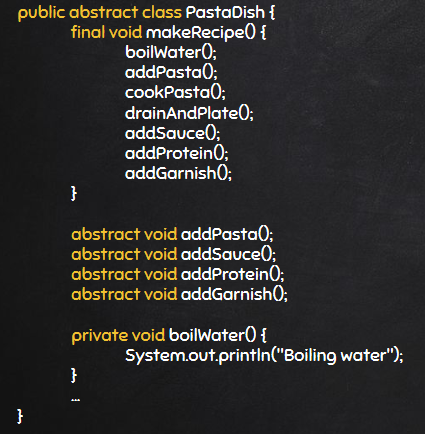
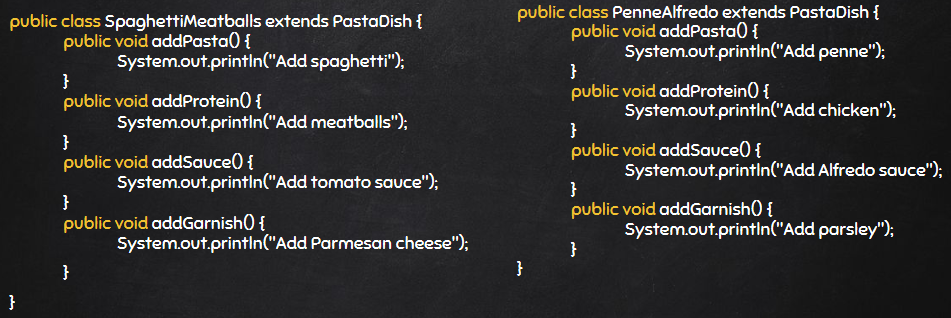
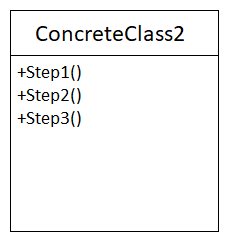
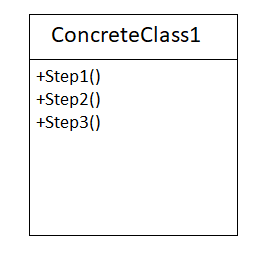
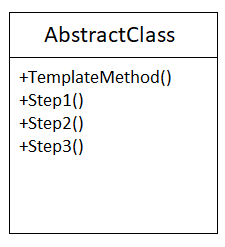
**Design Pattern – 1**

**Template**

Firstly, defines an algorithm generaly in a superclass and let some steps to be defined in the subclasses. This tecnique is used when we have two or more classes with similar funcionalities. This is a good soluction beacause is better than make changes in two diferente places.

**Code Snippet:**

**Structure:**



**Code Location:**

The classes are in the folloying path:

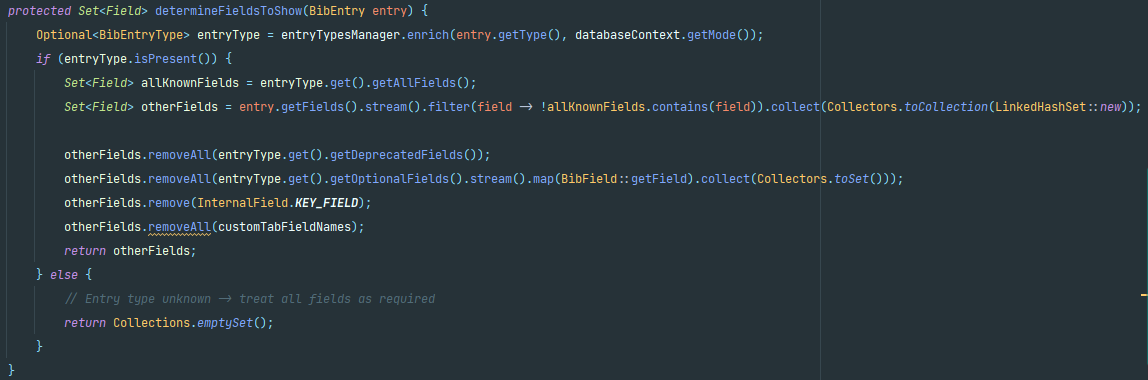
“*src\main\java\org\jabref\gui\entryeditor\FieldsEditorTab.java*”

A new abstract class (superClass) is created:



A new abstract method is created:

“*src\main\java\org\jabref\gui\entryeditor\OptionalFieldsTabBase*”

Method determideFieldsToShow(BibEntry entry):

“*src\main\java\org\jabref\gui\entryeditor\OthersFieldsTab*”

Method determideFieldsToShow(BibEntry entry):



**Justification:**

As shown in the superclass(FieldsEditorTab), it has to be abstract and have at least one abstract method, with no implementation. This implementation must be done in the subclasses (OthersFieldsTab and OptionFieldsTabBase) and it should be different from one class to another.