# Design patterns

1. Observer pattern

found @ src/main/java/org/jabref/gui/LibraryTab/IndexUpdateListener.java



Uma imagem com texto

Descrição gerada automaticamente

I identified this as an implementation of the observer design pattern because a subject (public class LibraryTab that holds the private class IndexUpdateListener) keeps an observer (indexingTaskManager) and, when an event happens, the subject is notified of it and calls the correct method from the observer so he can be updated accordingly, as show in the snippet above.

1. Factory pattern

found @ src/main/java/org/jabref/model/entry/field/FieldFactory.java



Uma imagem com texto

Descrição gerada automaticamente

I identified this as an impure implementation of the factory pattern (since it deviates from the textbook definition). A Product (interface Field, present in the same package of the Creator) has different implementations, as in different field properties, which will work as concrete products. A Creator (class FieldFactory) has various methods that return those different types of fields (mostly as sets of those), therefore returning existing objects with a specific property. It deviates from the textbook definition because the Creator doesn’t have methods that return new objects, but sets of existing ones with a specific property, and the Product doesn’t have different implementations per se, but has a specific property attached, simulating polymorphism.

1. Singleton pattern

found @ src/main/java/org/jabref/preferences/JabRefPreferences.java

Uma imagem com texto, laranja

Descrição gerada automaticamente



Uma imagem com texto

Descrição gerada automaticamente

I identified this as a textbook implementation of the singleton pattern. It has the two steps needed for it: has a private constructor JabRafPreferences (so other objects can’t use the `new` operator with the class) and there’s a static method that calls the private constructor to assign the desired object to a static variable, so all calls to the method return the cached object.