**The first Design Pattern Identified is: Caching Pattern.**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

**It can be found in the path: jabref -> src -> main -> java -> org. jabref -> logic -> citationstyle -> CitationStyleCache.java**

The caching pattern avoids expensive re-acquisition of resources by not releasing them immediately after use. The resources retain their identity, are kept in some fast-access storage, and are re-used to avoid having to acquire them again.

In this case the class: **CitationStyleCache.java** is going to work on caching citations for quicker access, while **CitationStyleGenerator.java** will generate them slower.

**The second Design Pattern Identified is: Registry Pattern.**

**Text

Description automatically generated**

**It can be found in the path: jabref -> src -> main -> java -> org. jabref -> model -> entry -> types -> AuthorList.java**

The Registry Pattern denotes the storage of the objects of a single class and provides a global point of access to them. Similar to Multiton pattern, only difference is that in a registry there is no restriction on the number of objects.

In this case the class: **AuthorList.java** serves as a registry of objects of type **Author.java.**

**The third Design Pattern Identified is: Singleton Pattern.**

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

Instancing of the Singleton.

**It can be found in the path: jabref -> src -> main -> java -> org. jabref -> preferences -> AppearencePreferences.java**

This design pattern ensures a class only has one instance and provides a global point of access to it.

Particularly, the **AppearancePreferences.java** class is only instantiated once, in the **JabRefPreferences.java** class.