


# Design Patterns

The first Design Pattern Identified is: Caching Pattern.

```
20 public class CitationStyleCache {  
21  
22     private static final int CACHE_SIZE = 1024;  
23  
24     private PreviewLayout citationStyle;  
25     private final LoadingCache<BibEntry, String> citationStyleCache;  
26  
27     @  public CitationStyleCache(BibDatabaseContext database) {
```

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.

# Design Patterns

The second Design Pattern Identified is: Registry Pattern.

```
116 public class AuthorList {
117
118     private static final WeakHashMap<String, AuthorList> AUTHOR_CACHE = new WeakHashMap<>();
119     private final List<Author> authors;
120     private AuthorList latexFreeAuthors;
121
122     /**
123      * Creates a new list of authors.
```

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**.

# Design Patterns

The third Design Pattern Identified is: Singleton Pattern.

```
12 public class AppearancePreferences {
13     private final BooleanProperty shouldOverrideDefaultFontSize;
14     private final IntegerProperty mainFontSize;
15     private final ObjectProperty<Theme> theme;
16
```

```
428     private PreviewPreferences previewPreferences;
429     private SidePanePreferences sidePanePreferences;
430     private AppearancePreferences appearancePreferences;
431     private ImporterPreferences importerPreferences;
```



Instanting of the Singleton.

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