Design Patterns:

1 - Façade

src -> model -> entry -> field -> Field

Interface

Simple interface that defines the expected and default returns of the methods to be

implemented by the classes using it. Examples of classes that implement "Field":

"InternalField", "SpecialField", "StandardField" and "UnknownField".

2 - Prototype

src -> model -> entry -> BibEntry

Lines 633-640

public Object clone() {

BibEntry clone = new BibEntry(type.getValue());

clone.fields = FXCollections.observableMap(new ConcurrentHashMap<>(fields));

clone.commentsBeforeEntry = commentsBeforeEntry;

clone.parsedSerialization = parsedSerialization;

clone.changed = changed;

return clone;

}

Class implements Cloneable interface, allowing current instance to be cloned, providing

only a new ID for the cloned object while saving all information from its' original

counterpart.

3 - Singleton

src -> gui -> externalfiletype -> ExternalFileTypes

Entire class

Class has private constructor, guaranteeing that it can only be instantiated by itself and has a getInstance() method. It can, however, be accessed via public methods like "getExternalFileTypeSelection()".