Design Patterns:

1 - Composite

scr -> logic -> bst -> VM

lines -> 102/104

public interface BstFunction {

void execute(BstEntry context);

}

"VM" is a common interface which is used for different objects that do the same thing

in different contexts, such as, on "TextPrefixFunction" class, "FormatNameFunction" class,

"ChangeCaseFunction" class, "PurifyFunction" class and "WidthFunction" class.

2 - Template

scr -> logic -> formatter -> bibtexfields -> Formatter

Entire class

Class used to define the skeleton code of an algorithm in the superclass and use subclasses to override specific steps of the algorithm without changing its structure.

Examples of subclasses that extend superclass "Formatter": "NormalizeNamesFormatter",

"NormalizeMonthFormatter", "NormalizeEnDashesFormatter", "NormalizePagesFormatter" and "NormalizeDateFormatter".

3 - Facade

scr -> logic-> integrity -> ValueChecker

Interface

public interface ValueChecker {

/\*\*

\* Validates the specified value.

\* Returns a error massage if the validation failed. Otherwise an empty optional is returned.

\*/

Optional<String> checkValue(String value);

}

It's used a simple interface to a complex subsystem of classes. The interface validates a specified

value in each class that implements the "ValueChecker" Interface. This design pattern used in classes,

such as, "TitleChecker", "PagesChecker", "Year Checker", "ValidCitationKeyChecker" and many more,

providing a simplified and straightforward way to check the information needed.