

# Design Patterns

The first design pattern: **Singleton Class**

```
private BibtexCaseChanger() {  
}
```

Located in: **jabref > logic > bst > BibtextCaseChanger.java**

A private constructor, as the above, ensures there is only one instance of this class to be produced and exist throughout an application's lifetime and guarantees that a single global point of access exists to it.

The second design pattern: **Template Method**

```
/**  
 * Base class for export formats based on templates.  
 */  
public class TemplateExporter extends Exporter {
```

Located in: **jabref > logic > exporter > TemplateExporter.java**

This class creates a template implementation for all export formats, leaving the implementation to its subclasses.

The third design pattern: **Factory Pattern**

```
public class ExporterFactory {
```

Located in: **jabref > logic > exporter > ExporterFactory.java**

The class implements a factory creation method for the many exports, hiding their implementation behind an interface.