# Patterns

## Prototype

Podemos verificar este padrão no pacote org.jabref.logic.integrity, na classe *IntegrityMessage* onde a interface *prototype* é a *clonable* e a classe prototype concreta é a *IntegrityMessage,* esta classe é usada por diversas outras, também podemos verificar outras classes que implementam o *clonable*, como a *BibtexString* e a *BibEntry.* Este pattern permite-nos copiar objetos existentes sem fazer seu código ficar dependente de suas classes.



## Factory Pattern – ActionFactory

Podemos encontrar este *pattern* no package org.jabref.gui.actions onde a classe *ActionFactory* é a classe criadora, o produto pode ser considerado a interface *Action* (*org.jabref.gui.actions*) visto que esta é comum a todos os objetos criados e os outros parâmetros passados são classes java como o command, aqui a classe criadora fornece uma interface para criar objetos, mas permite que as se altere o tipo de objetos que serão criados, no caso podem ser criados menus e botões que podem realizar diversas ações.

 

## Singleton Pattern – ExternalFileTypes

Podemos encontrar este *pattern* na classe *ExternalFileTypes* (*org.jabref.gui.externalfiletype*), onde o *singleton* é esta mesma classe e o cliente é por exemplo a classe *JabRefDesktop*  (*org.jabref.gui.desktop*). Este *pattern* garante-nos que existe apenas uma instância da classe *ExternalFileTypes* ao longo de toda a execução do programa.



Realizado por: Diogo Rosa 57464