## **Guilherme Poças 60236**

### Code Patterns

1. Factory method pattern

*public interface* ActiveActionProvider {  
 */\*\* Provides the active action, which might depend on external influences \*/  
 public abstract* AbstractAction getActiveAction();  
}

Location: ganttproject/src/main/java/net/sourceforge/ganttproject/action/ActiveActionProvider.java

This interface is used to create an Action object, but hides the creation code on the actual class, making the creation process independent of the concrete class.

1. Facade pattern

*class* UIFacadeImpl *extends* ProgressProvider *implements* UIFacade

Location: ganttproject/src/main/java/net/sourceforge/ganttproject/UIFacadeImpl.java

This class allows other parts of the program to access functionalities and methods of other classes responsible for the UI, wrapping them in a subsystem, and making it easier for the user to use them

1. Template method pattern

*public abstract class* ImporterBase *implements* Importer

*public class* ImporterFromGanttFile *extends* ImporterBase

*public class* ImporterFromTxtFile *extends* ImporterBase

Location: ganttproject/src/main/java/net/sourceforge/ganttproject/importer/ImporterBase.java

The two importer classes import from different types of files, so have different responsibilities, but a similar function. Because of this, they defer some similar methods to the ImporterBase abstract class, making the code cleaner and avoiding repeated code.