**Student**

Francisco Parrinha – 58360

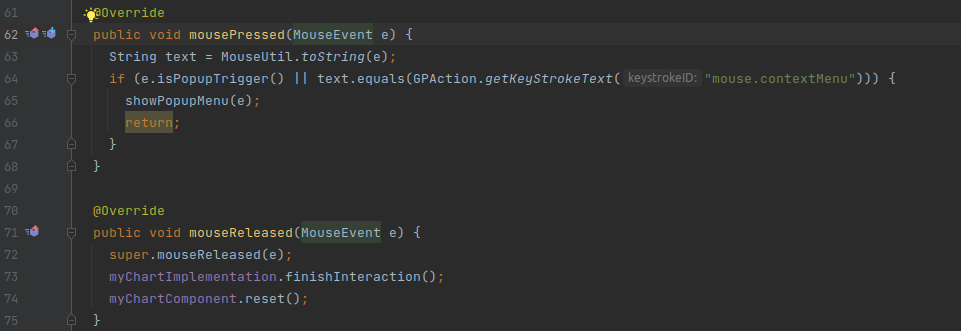
**Design Patterns**

1. Observer:

* Class *MouseListenerBase* in: *ganttproject/src/main/java/net/sourceforge/ganttproject/chart/mouse/MouseListenerBase.java*

An observer pattern is a design pattern where an object, named *subject*, maintains a list of its dependencies. also known as *observers*.

The following class is an observer. The following code snippet shows the update function used to inform the subject:

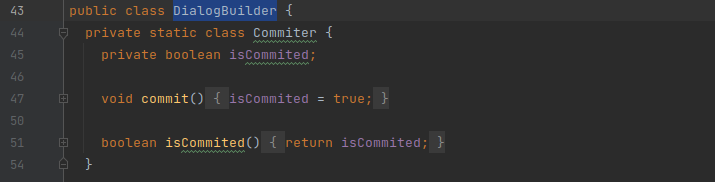


2. Builder:

* Class *DialogBuilder* in: *ganttproject/src/main/java/net/sourceforge/ganttproject/DialogBuilder.java*

A builder pattern is a design pattern that offers a flexible solution to various object-oriented programming. Its intent is to separate the construction of an object from its realization.

The following class is a builder class. It simplifies the construction of dialog objects.

The design pattern:

2. Facade:

* Class *UIFacadeImpl* in: *ganttproject/src/main/java/net/sourceforge/ganttproject/UIFacadeImpl.java*

A facade pattern is a design pattern commonly used in object-oriented programming. A facade is an object that serves as a front-facing interface, making complex code more accessible. It can improve its readability and its usability.

Objects instantiated from this class are facades. This class serves the purpose explained above by adding several minimal methods that have larger and more complex implementations in other classes.

A screenshot of the class:

