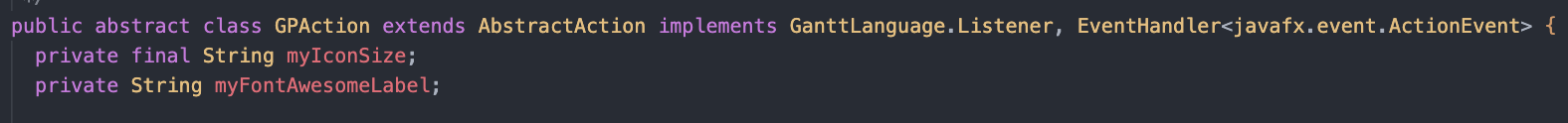
Joana Wang 60225

**Design Patterns**

* Command

**Code snippet:**

****

**Location:**

Ganttproject>src>main>java>net>sourceforge>ganttproject>action>edit

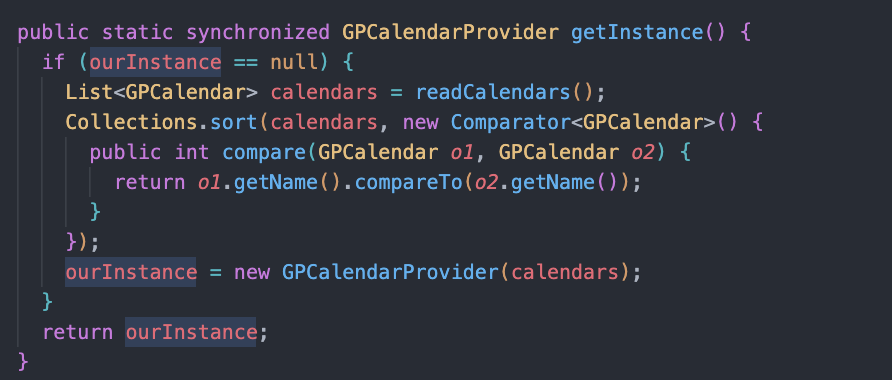
>EditMenu.java

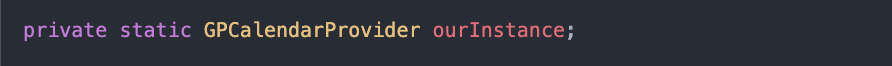
**Explanation:**

A request is encapsulated as an object, which serves as a command, invoked by the invoker object to complete tasks that it supposed to do.

* Singleton

**Code snippet:**





**Location:**

ganttproject>src>main>java>net>sourceforge>ganttproject>calender>

GPCalendarProvider.java

**Explanation:**

In this pattern, a class is restricted to having only one instance. Thus, the instance is stored in a private static variable and a static method is provided that returns a reference to the instance.

* Template method

**Code snippet:**

**Location:**

* ganttproject>src>main>java>net>sourceforge>ganttproject>action>resource>ResourceAction.java
* ganttproject>src>main>java>net>sourceforge>ganttproject>action>resource>ResourceNewAction.java
* ganttproject>src>main>java>net>sourceforge>ganttproject>action>resource>ResourceDeleteAction.java
* ganttproject>src>main>java>net>sourceforge>ganttproject>action>resource>ResourcePropertiesAction.java

**Explanation:**

This method design pattern is to define an algorithm in general, leaving the implementation details to its subclasses. In this specific case, the *ResourceAction.java* is an abstract class, extended by three other subclasses, where each one implements their own abstract method.