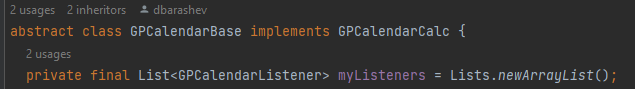
**Design Patterns**

* **Observer**:

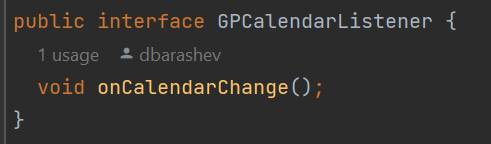
The next piece of code typically represents an example of Observer Pattern in which an [object](https://en.wikipedia.org/wiki/Object_(computer_science)#Objects_in_object-oriented_programming), named the subject(in this case all classes that extend the abstract class GPCalendarBase - *WeekendCalendarImpl* and *AlwaysWorkingCalendarImpl*), maintains a list of its dependents, called observers (designated *myListeners*), and notifies them automatically of any [state changes](https://en.wikipedia.org/wiki/Event_(computing)), usually by calling one of their [methods](https://en.wikipedia.org/wiki/Method_(computer_science)). In next example calendar listeners are notified (through the function *fireCalendarChanges()*) when calendar is changed, namely, when weekends days change or holidays list change.











Source: biz.ganttproject.core.calendar