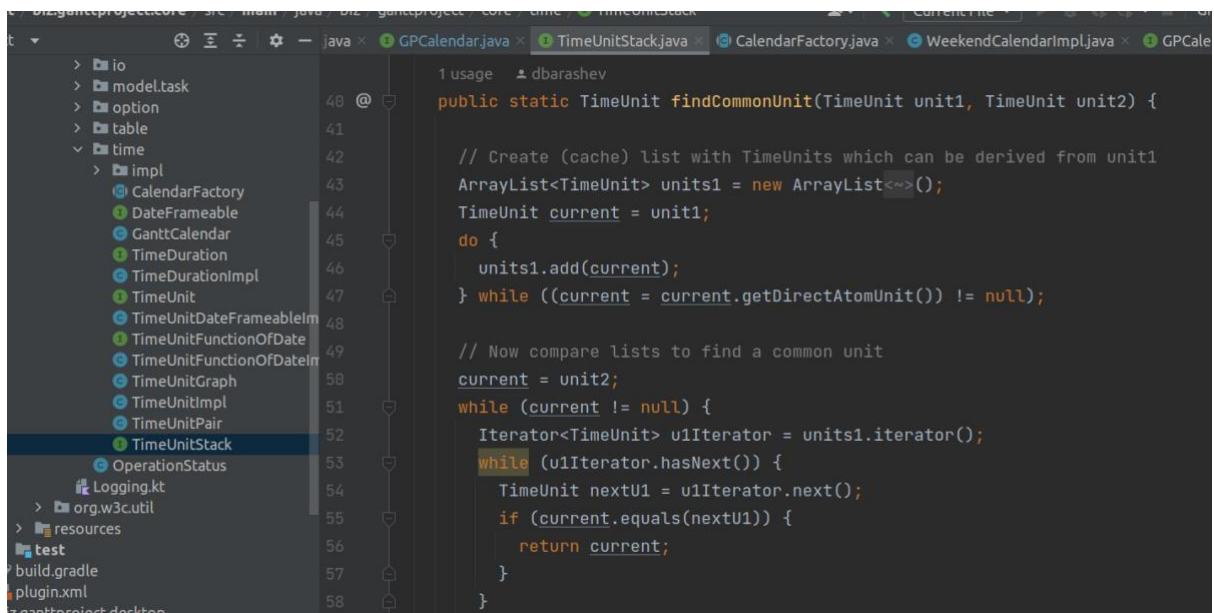


Iterator Pattern



The screenshot shows an IDE with the file explorer on the left displaying a project structure. The file `TimeUnitStack.java` is selected. The main editor shows the code for `TimeUnitStack`, which implements the `TimeUnit` interface. The code uses the Iterator pattern to find a common unit between two `TimeUnit` objects. The `findCommonUnit` method creates a list of units from the first unit and then iterates through the second unit's units to find a common one.

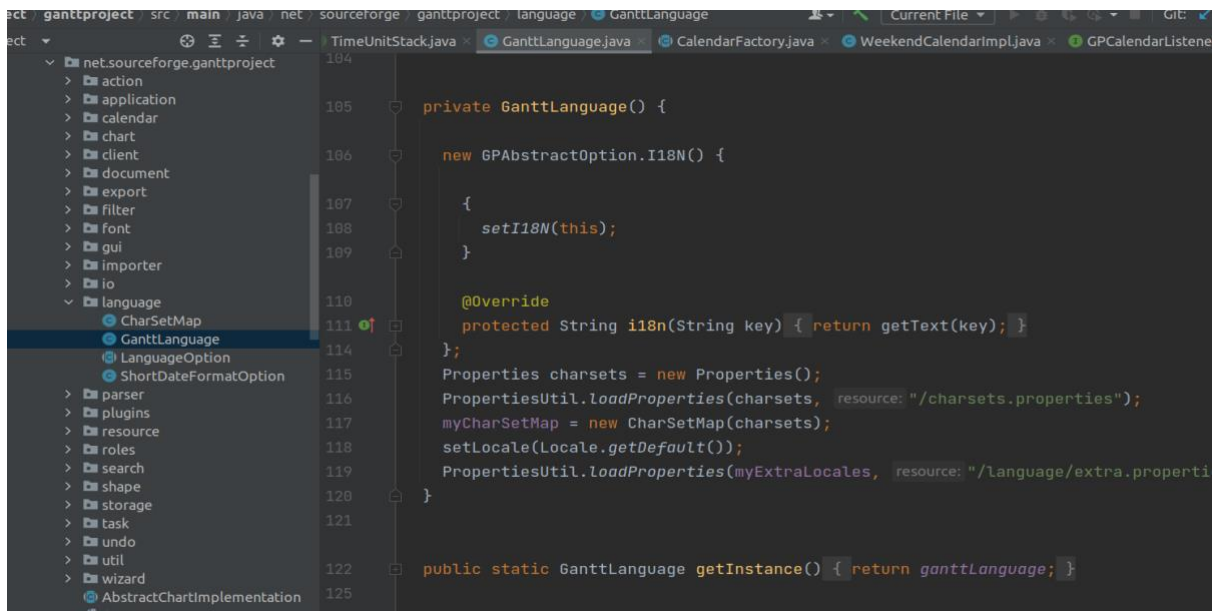
```
1 usage dbarashev
public static TimeUnit findCommonUnit(TimeUnit unit1, TimeUnit unit2) {
    // Create (cache) list with TimeUnits which can be derived from unit1
    ArrayList<TimeUnit> units1 = new ArrayList<>();
    TimeUnit current = unit1;
    do {
        units1.add(current);
    } while ((current = current.getDirectAtomUnit()) != null);

    // Now compare lists to find a common unit
    current = unit2;
    while (current != null) {
        Iterator<TimeUnit> u1Iterator = units1.iterator();
        while (u1Iterator.hasNext()) {
            TimeUnit nextU1 = u1Iterator.next();
            if (current.equals(nextU1)) {
                return current;
            }
        }
        current = current.getDirectAtomUnit();
    }
}
```

Ficheiro: `biz.ganttproject.core/src/main/java/biz/ganttproject/core/time/TimeUnitStack.java`

Texto: A classe `Util` que é um inner class da Interface `TimeUnitStack` usa o pattern `Iterator` no método estático `findCommonUnit`.

Singleton pattern



The screenshot shows an IDE with the file explorer on the left displaying a project structure. The file `GanttLanguage.java` is selected. The main editor shows the code for `GanttLanguage`, which implements the `GanttLanguage` interface. The class uses the Singleton pattern, with a private constructor and a static `getInstance` method that returns the single instance of the class.

```
104
private GanttLanguage() {
    new GPAbstractOption.I18N() {
        {
            setI18N(this);
        }
    };

    @Override
    protected String i18n(String key) { return getText(key); }
};

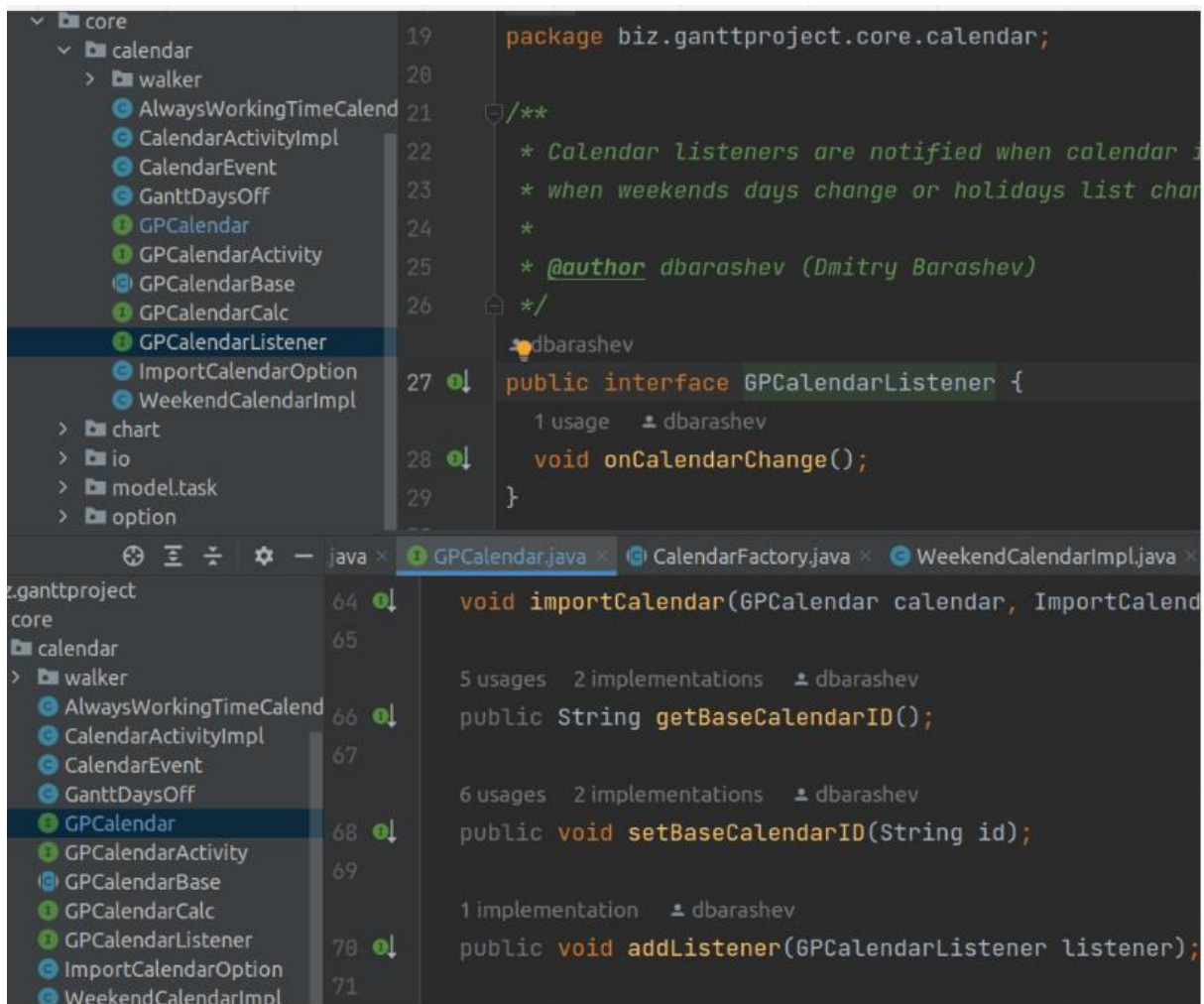
Properties charsets = new Properties();
PropertiesUtil.loadProperties(charsets, resource: "/charsets.properties");
myCharSetMap = new CharSetMap(charsets);
setLocale(Locale.getDefault());
PropertiesUtil.loadProperties(myExtraLocales, resource: "/language/extra.properties");

122 public static GanttLanguage getInstance() { return ganttLanguage; }
125
```

Ficheiro: `ganttproject/src/main/java/net/sourceforge/ganttproject/language/GanttLanguage.java`

Texto: A classe `GanttLanguage` tem um construtor privado e um método estático `getIntance()` que devolve a única instância da classe.

Observer Pattern



Ficheiro: biz.ganttproject.core/src/main/java/biz/ganttproject/core/calendar/

Texto: No package biz.ganttproject.core.calendar há um observer pattern onde a interface do Subject é o GPCalendar e a interface do Observer é GPCalendarListener.