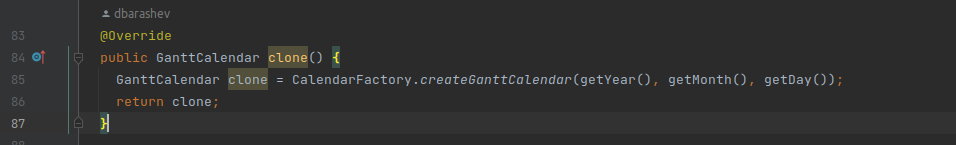
**Identified design patterns**

Team member number:

**Patern 1 – Creational Pattern (cloning)**

Code snippet



**The exact location on the codebase**

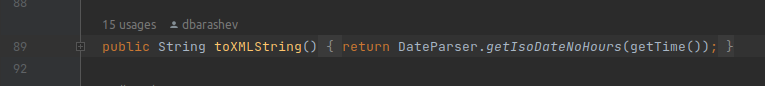
biz.ganttproject.core/src/main/java/biz/ganttproject/core/time/GanttCalendar.java

**Explanation of the rationale for identifying this as a pattern instantiation**

Creational patterns deal with cloning of objects, and although the GregorianCalendar util already has a clone method that could be used instead of this one, it is still a cloning method.

**Patern 2 – Adapter Pattern**

**Code snippet**



**The exact location on the codebase**

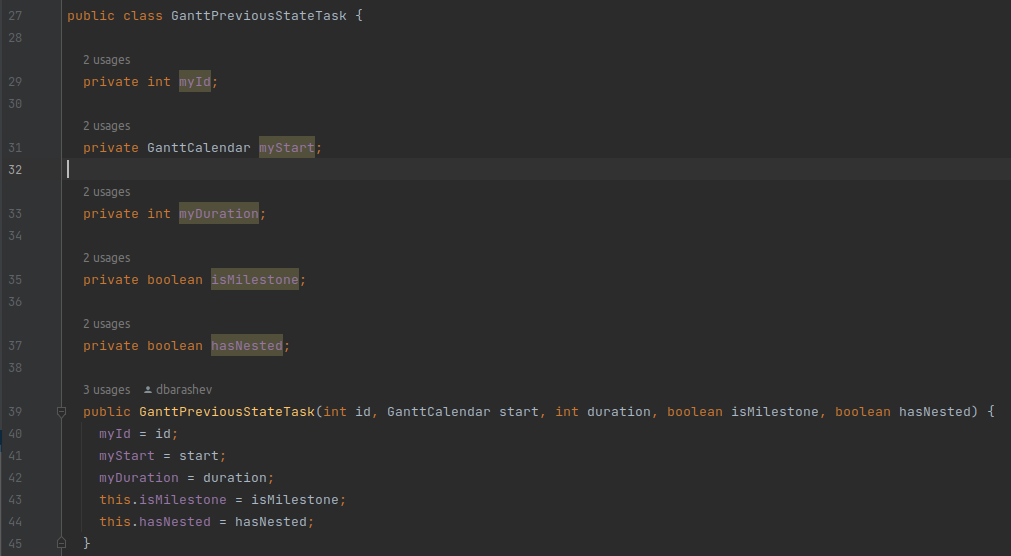
biz.ganttproject.core/src/main/java/biz/ganttproject/core/time/GanttCalendar.java

**Explanation of the rationale for identifying this as a pattern instantiation**

An adapter pattern facilitates communication between two existing systems by providing a compatible interface. Well, by parsing our calendar to an xml format, we establish this compatible interface with, for example, web pages.

**Patern 3 – Memento Pattern**

**Code snippet**



**The exact location on the codebase**

ganttproject/src/main/java/net/sourceforge/ganttproject/GanttPreviousStateTask.java

**Explanation of the rationale for identifying this as a pattern instantiation**

This class saves a previous state of the task, meaning the user can go back to it, or “undo”.