
Software Engineering

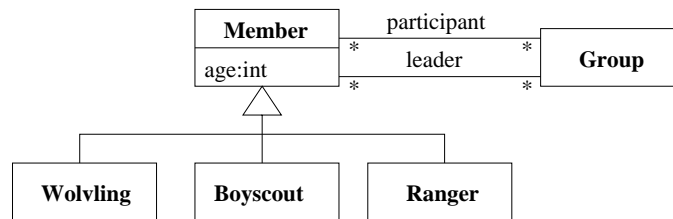
<http://proglang.informatik.uni-freiburg.de/teaching/swt/2005/>

Exercise Sheet 10

Deadline: June 28th, 2005

Exercise 1 – OCL: (5 points)

The following class diagram models a boyscout group:



Implement the following constraints in OCL:

1. Wovlings have to be between 7 and 11 years old.
2. Every Group has at least one leader who is a Ranger.
3. Every Ranger leads at least one Group.
4. All leaders of a Group are at least 16 years old.
5. Every Group has at least 1 leader per 15 members.

Solution:

1. context Member inv
self.oclIsTypeOf(Wovling) implies self.age >= 7 and self.age <= 11
2. context Group inv
leader->exists(oclIsTypeOf(Ranger))
3. context Member inv
self.oclIsTypeOf(Ranger) implies leader->size() > 0
4. context Group inv
leader->forall(age >= 16)
5. context Group inv
participant->size() < leader->size() * 15

Exercise 2 – Design Patterns: (5 points)

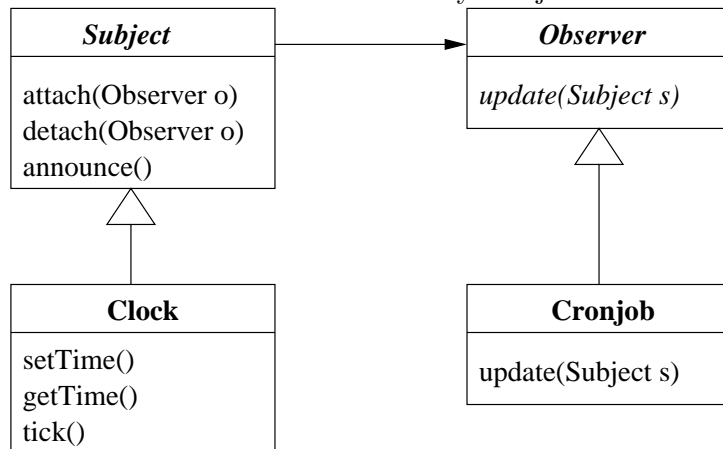
Some commands shall be executed on a computer at specific times or after specific time intervals (cronjobs). So they all have to know the system time and get informed if it changes. Which Design Pattern can be useful in implementing this behaviour?

Draw a class diagram of this problem.

Implement this system. Assume that the system clock only stores a number of seconds. Give one Cronjob class as stub, e.g. leaving out the functionality it implements but giving only the framework.

Solution:

We use the *Observer* Pattern. Every cronjob observes the system clock.



The implemented solution can be found on the webpage (file “Cron.java”).