Softwarekonstruktion - Exercise 3

3 Observer Design Pattern

This exercise should be solved until **Wednesday (23:59 latest), November 3rd, 2010**. You have to submit your solution to your tutor by email:

Holger Schmidt: holger.schmidt [at] cs.tu-dortmund.de
Gregor Kotainy: gregor.kotainy [at] tu-dortmund.de

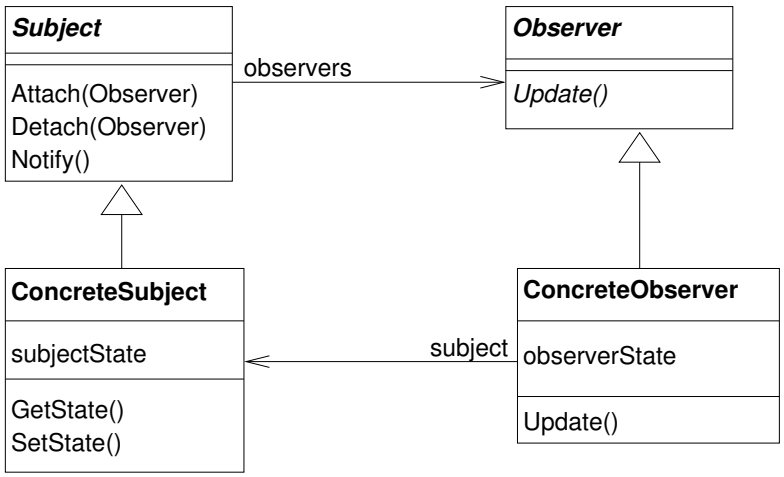
You have to work in groups of two or three persons. Only one person per group has to submit a group's solution. State the names and matriculation numbers of the group members in your email and as a comment in each of your source code files.

3.1 Observer Design Pattern

3.1.1 Implement a simple Java application that simulates an arrival board at a train station:

- Arriving trains are stored in a simple list. Trains that have arrived can be manually deleted from the list. Use a data structure that implements the interface `java.util.Collection` (e.g., `java.util.ArrayList`) to realise the list (see `http://java.sun.com/javase/6/docs/api/`). To simplify matters, a train can be represented as a String (e.g. ``` ``Duisburg - Dortmund, 9:34'' ```).
- The list of arriving trains can be observed by any number of arrival boards. These should react to any kind of change that occurs in the list. The display of the arrival board can be realised by a simple output in `System.out`.

- Use the design pattern *Observer* presented in the lecture:



- To test your implementation, write a main method in which first some trains are added to the list and then a few of them are deleted again.