Introduction to Java for C++ Programmers

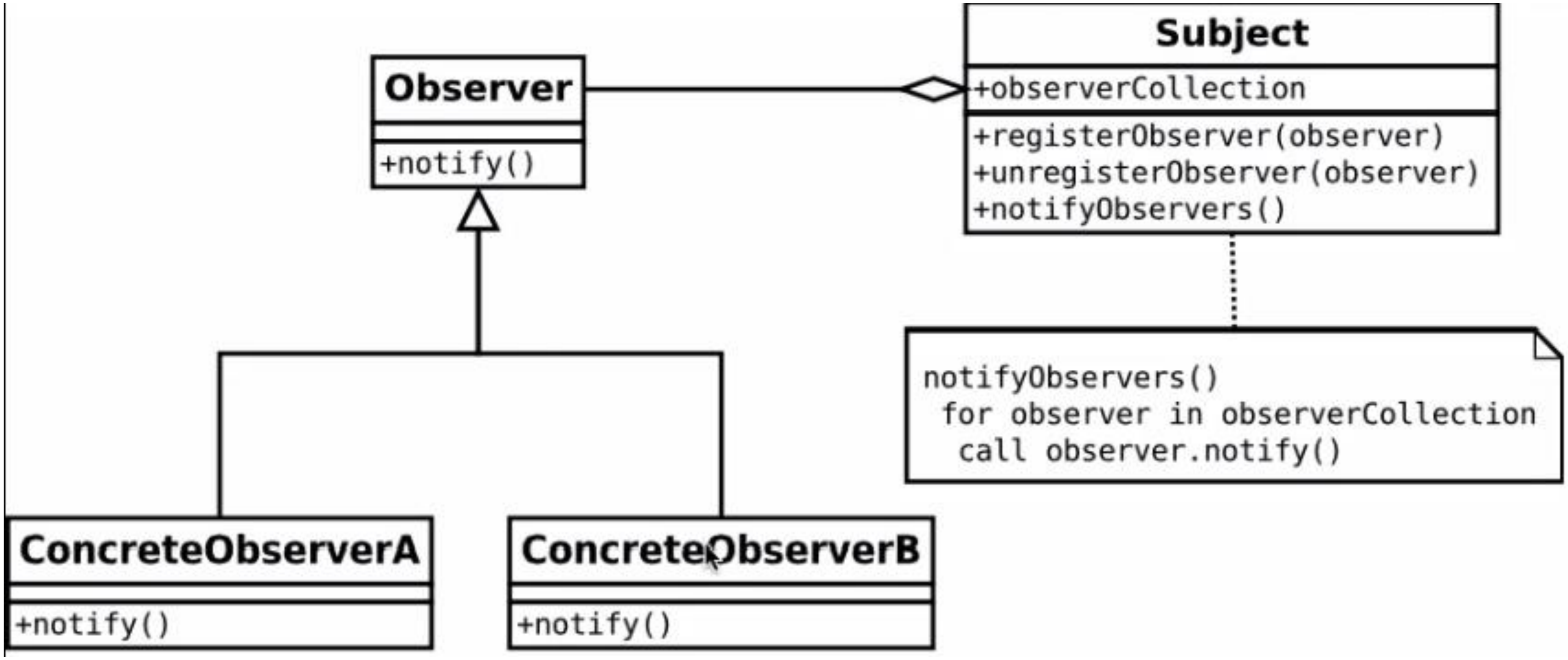
Observer Pattern

Mehrnaz Zhian

Observer pattern

- The observer design pattern is used to notify the objects of state changes.
- Simple way it works like, an object called an observer can register to another object as an interested party, and if changes occurs in it then the observer wants to know about it.
- You can have multiple observer objects to one particular object to whom state you want to observe.

- The object observers register themselves with is called a subject.
- Subject is the one that notifies its list of registered observers, so the subject object maintains the list of interested observers.
- Then it call on the observer objects methods to notify them when the changes occur.



```
notifyObservers()
 for observer in observerCollection
  call observer.notify()
```

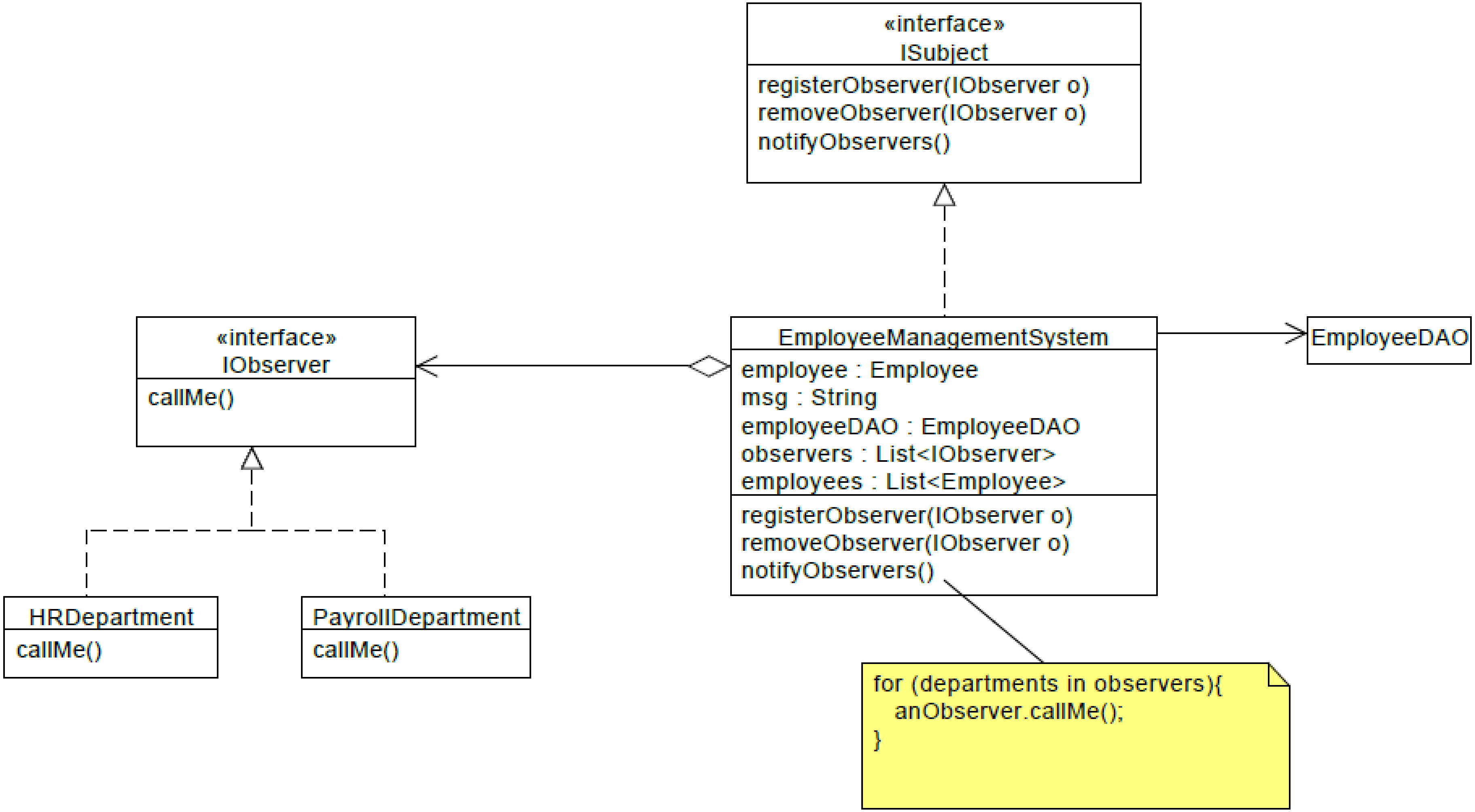
Problems Observer Pattern Solve

- Addresses the following problems
  - One-to-many dependency between objects should be defined without making the objects tightly coupled.
  - It should be ensured that when one object changes state an open-ended number of dependent objects are updated automatically.
  - It should be possible that one object can notify an open-ended number of other objects.

Solutions Observer Pattern describe

- Define **Subject** and **Observer** objects
- so that when a subject changes state, all registered observers are notified and updated automatically.

Example



```
for (departments in observers){
   anObserver.callMe();
}
```