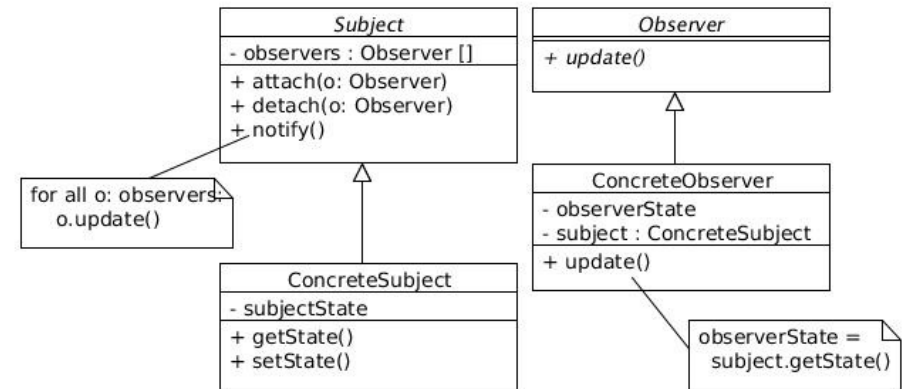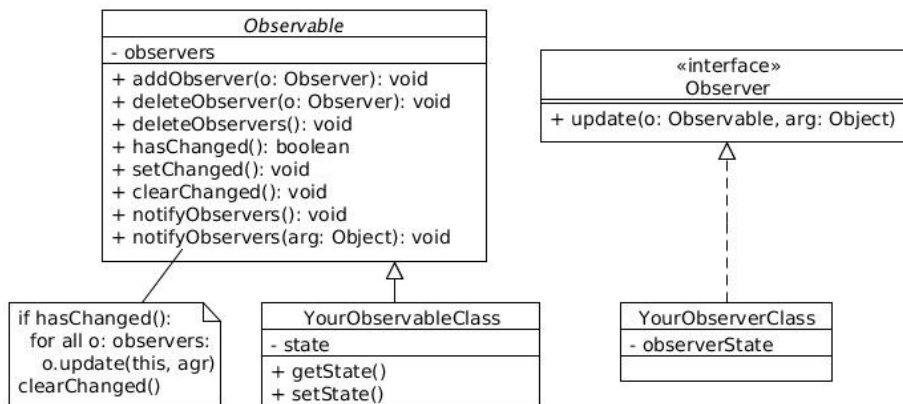# Observer Design Pattern

Problem:

- Need to maintain consistency between related objects.
- Two aspects, one dependent on the other.
- An object should be able to notify other objects without making assumptions about who these objects are.
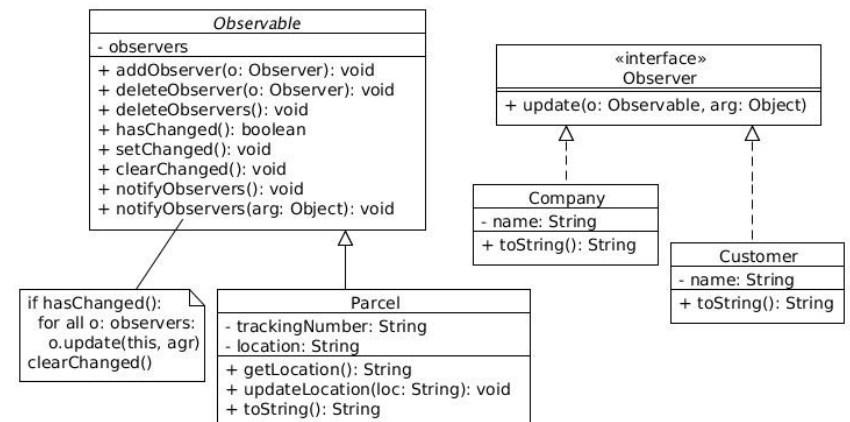
Observer: Standard Solution

Subject
- observers : Observer []
+ attach(o: Observer)
+ detach(o: Observer)
+ notify()

Observer
+ update()

for all o: observers:
   o.update()

ConcreteSubject
- subjectState
+ getState()
+ setState()

ConcreteObserver
- observerState
- subject : ConcreteSubject
+ update()

observerState = subject.getState()

Observer: Java Implementation

Observable
- observers
+ addObserver(o: Observer): void
+ deleteObserver(o: Observer): void
+ deleteObservers(): void
+ hasChanged(): boolean
+ setChanged(): void
+ clearChanged(): void
+ notifyObservers(): void
+ notifyObservers(arg: Object): void

«interface»
Observer
+ update(o: Observable, arg: Object)

if hasChanged():
   for all o: observers:
      o.update(this, agr)
clearChanged()

YourObservableClass
- state
+ getState()
+ setState()

YourObserverClass
- observerState

Observer: Example in Java

Observable
- observers
+ addObserver(o: Observer): void
+ deleteObserver(o: Observer): void
+ deleteObservers(): void
+ hasChanged(): boolean
+ setChanged(): void
+ clearChanged(): void
+ notifyObservers(): void
+ notifyObservers(arg: Object): void

«interface»
Observer
+ update(o: Observable, arg: Object)

Company
- name: String
+ toString(): String

Customer
- name: String
+ toString(): String

if hasChanged():
   for all o: observers:
      o.update(this, agr)
clearChanged()

Parcel
- trackingNumber: String
- location: String
+ getLocation(): String
+ updateLocation(loc: String): void
+ toString(): String

Uses of Observer in Java

In reality, people usually implement their own

- Usually can't or don't want to sub-class from `Observable`

- Can't have your own class hierarchy and multiple inheritance is not available

- Has been replaced by the Java Delegation Event Model (DEM)

  - Passes event objects instead of update/notify

Listener is specific to GUI classes