

Softwaretechnik

WiSe 09/10

Übungsblatt 11

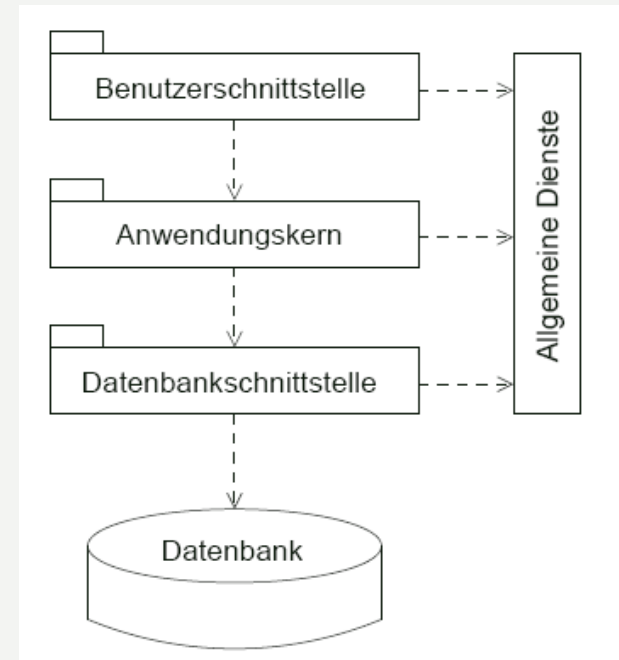
Aufgabe 1

Entwerfen Sie eine aus zwei Schichten bestehende Systemarchitektur für das Wetterstationssystem.

Aufgabe 2

Die in Übung 10 konstruierte Benutzerschnittstelle des Wetterstationssystems soll mit dem Anwendungskern verbunden werden. Modellieren Sie die Anbindung so, dass die Sichtbarkeitsregel aus der Vorlesung erfüllt wird. Dazu soll die Technik der indirekten Kommunikation mit Beobachtern (Observern) angewendet werden.

- Zwei Schichten, in diesem Fall
- Wetterstation (**Anwendungskern**)
- WSFrame (**Benutzerschnittstelle**)
- Abhängigkeiten gemäß Vorlesung:
- **Anwendungskern kennt UI NICHT!**
- Bisher sind UI und Kern der Wetterstation noch nicht verbunden...

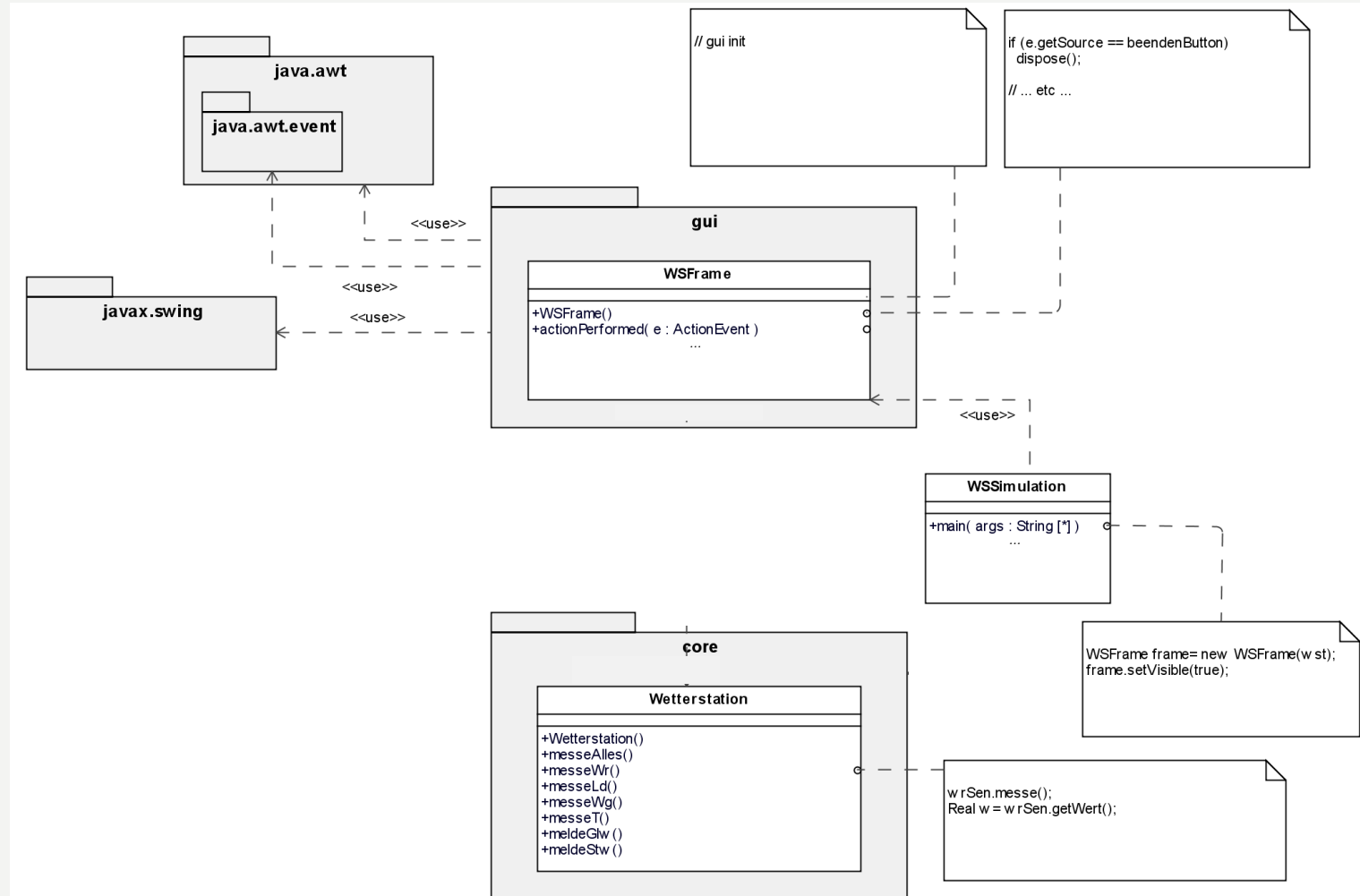


UI

UI

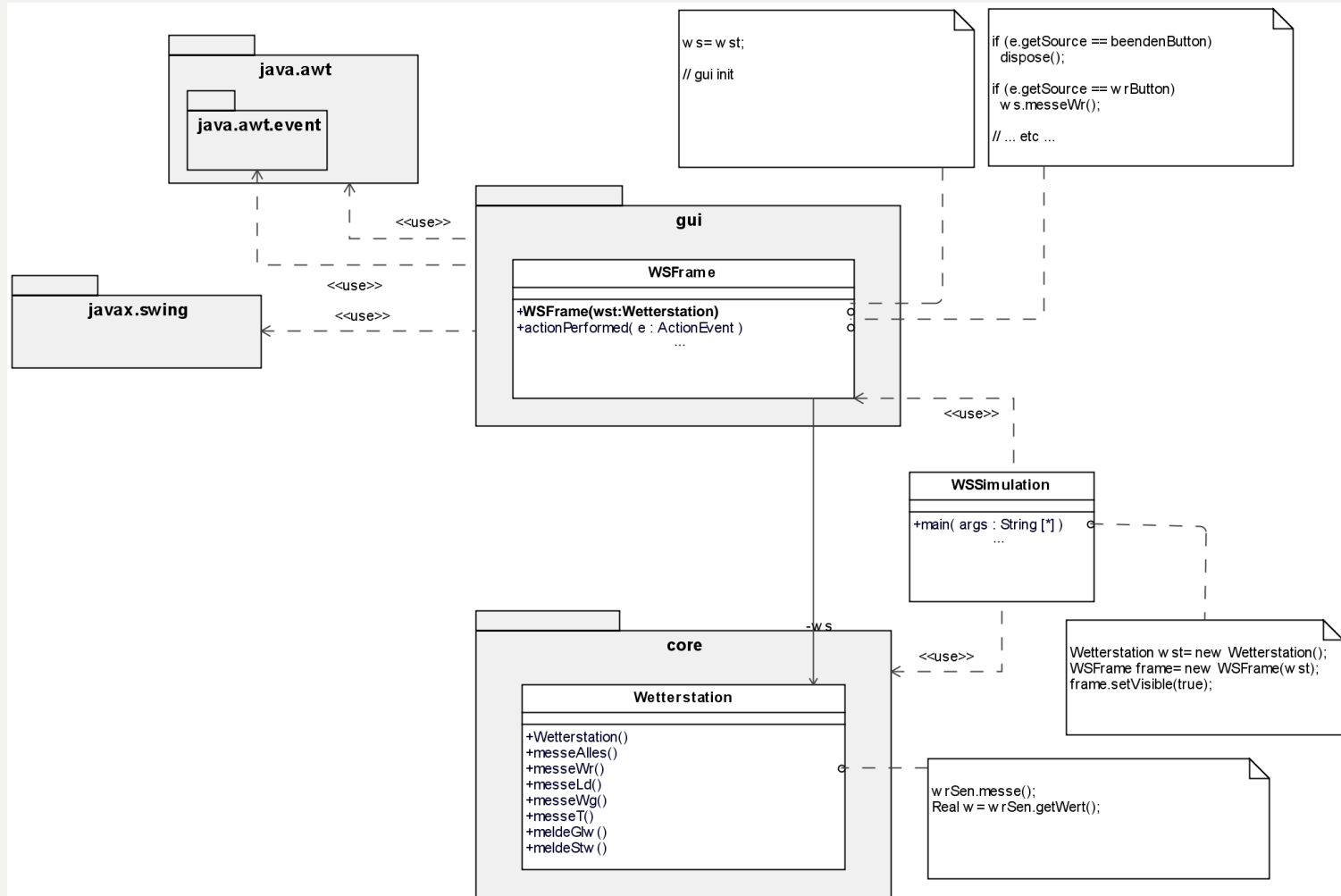
Core

Core

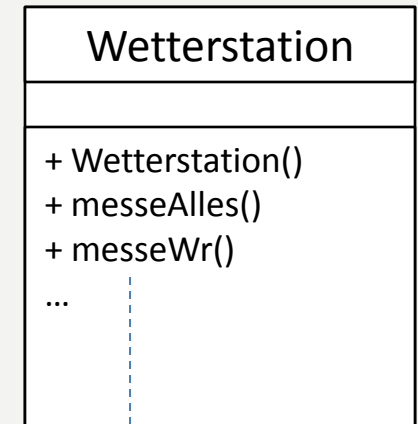
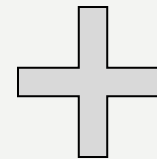
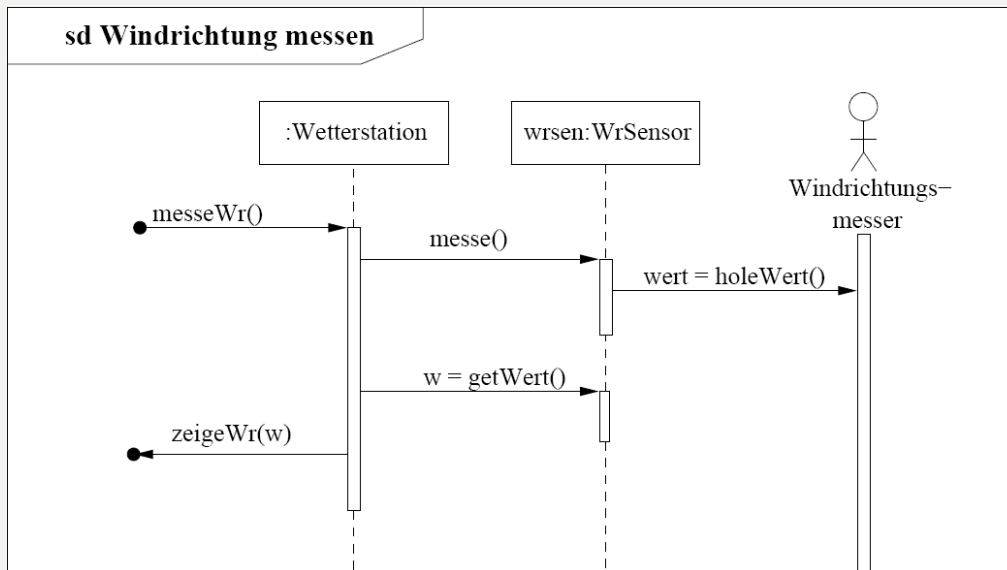


- „... [Die GUI] soll mit dem Anwendungskern verbunden werden. Modellieren Sie die Anbindung so, dass die Sichtbarkeitsregel aus der Vorlesung erfüllt wird.“
- Frage 1: Wie stößt die UI neue Berechnungen an?
- Frage 2: Wie kommuniziert der Kern neu gemessene Werte?

- **Frage 1:** Wie stößt die UI neue Berechnungen an?
 - WSFrame muss auf Wetterstation zugreifen können
 - Ziel: Aufruf der Wetterstationsmethoden
- **Lösung:**
 - **Assoziation** zw. WSFrame und Wetterstation
 - Erzeugung Wetterstation + **Injektion** in WSFrame in der Start-Klasse.

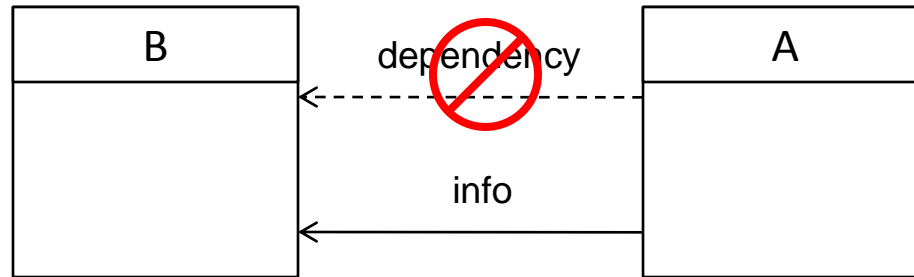


- **Frage 2:** Wie meldet der Kern Werte an die UI?
 - Bisher:

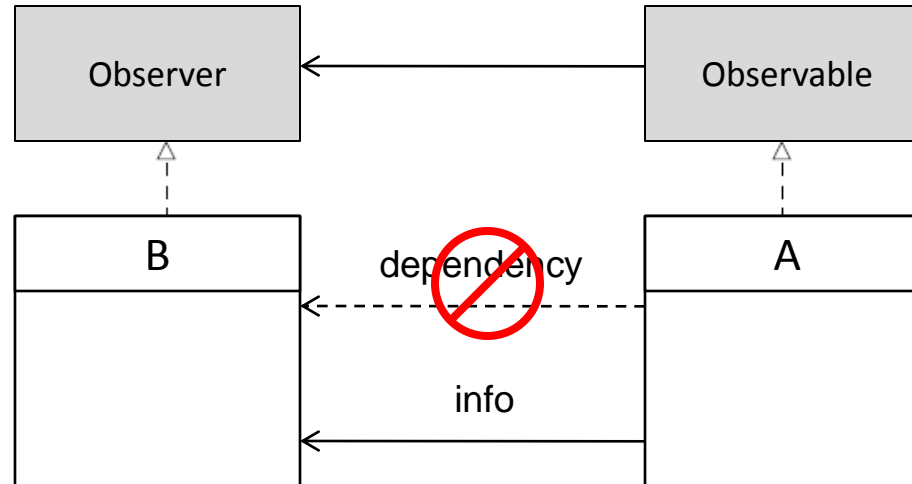


```
wr.messe();  
Real w= wr.getWert();  
gui.zeigeWr(w);
```

- **Problem:** Hier kennt der Kern die UI...

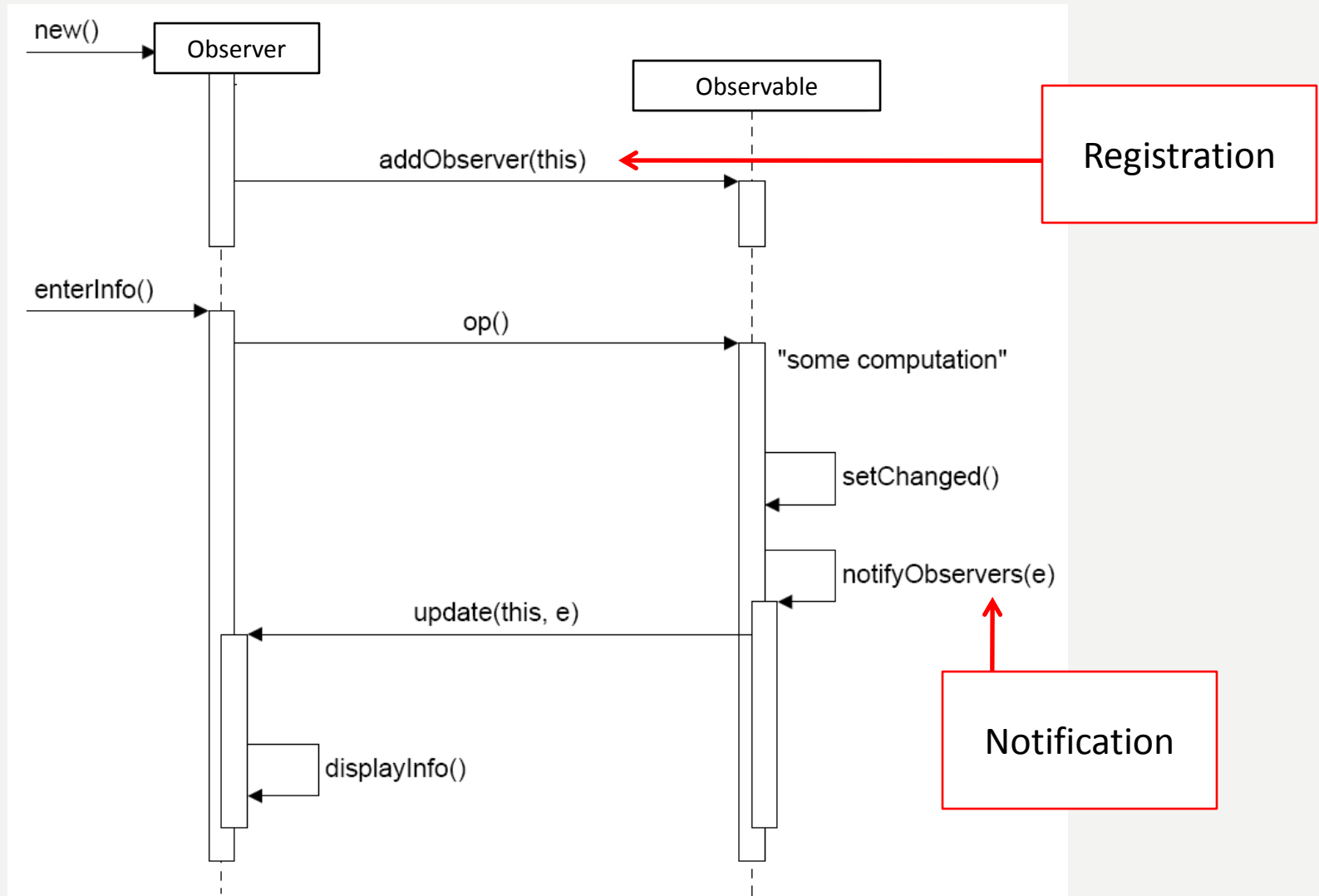


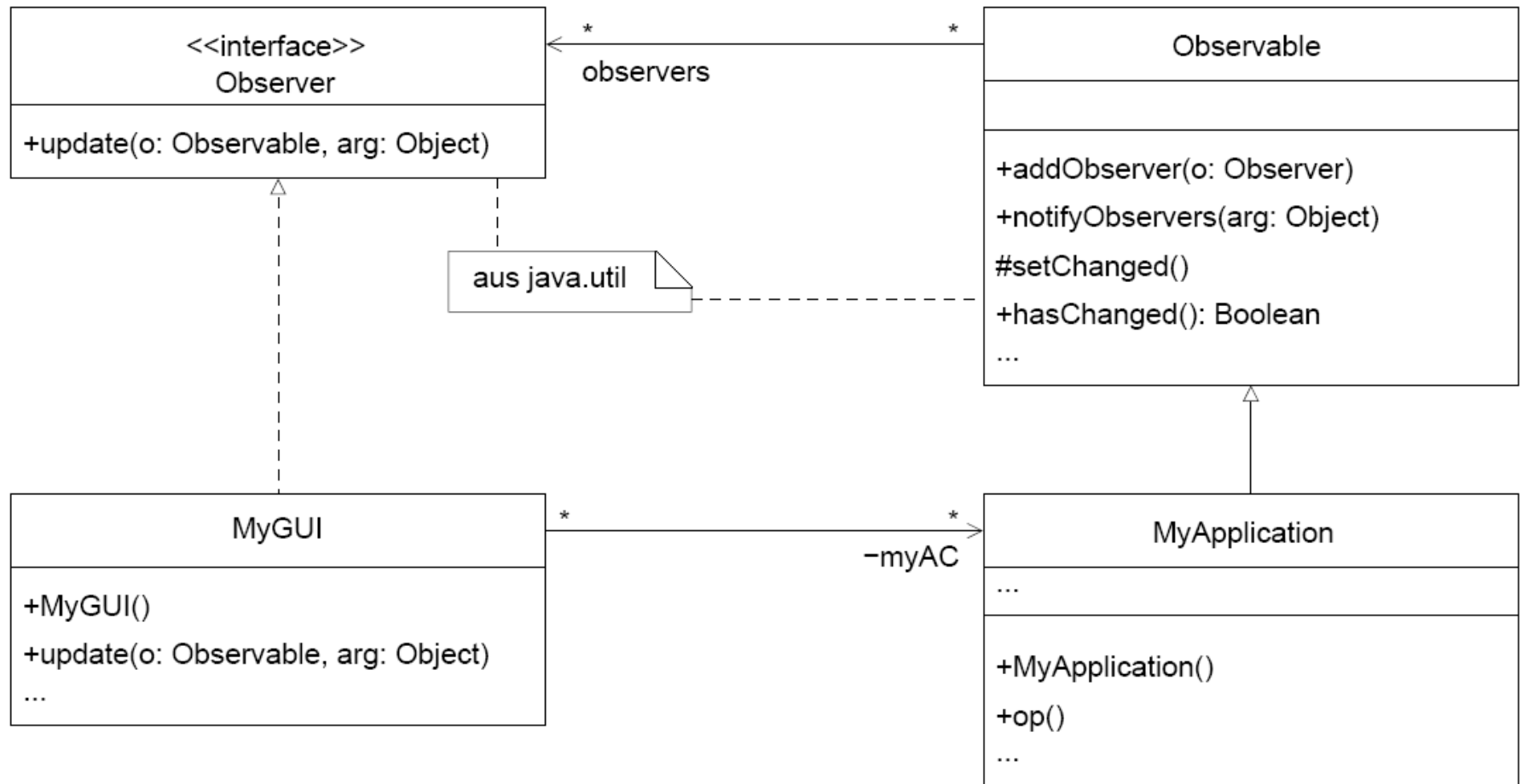
- Das Observer (Beobachter-) Pattern beschreibt, wie **A** Information an **B** schickt, ohne **B** zu kennen.



- **Grundidee:**

- A ist ein **Observable**: Es enthält interessante Informationen, die beobachtet werden sollen
- A kennt zwar B nicht, kennt aber eine Liste von **Observern**, die informiert werden wenn etwas interessantes passiert
- Der Trick: **B kann sich bei A als Observer registrieren.**





Drei Schritte:

a) Implementierung eines **Datenübertragungsobjekts**

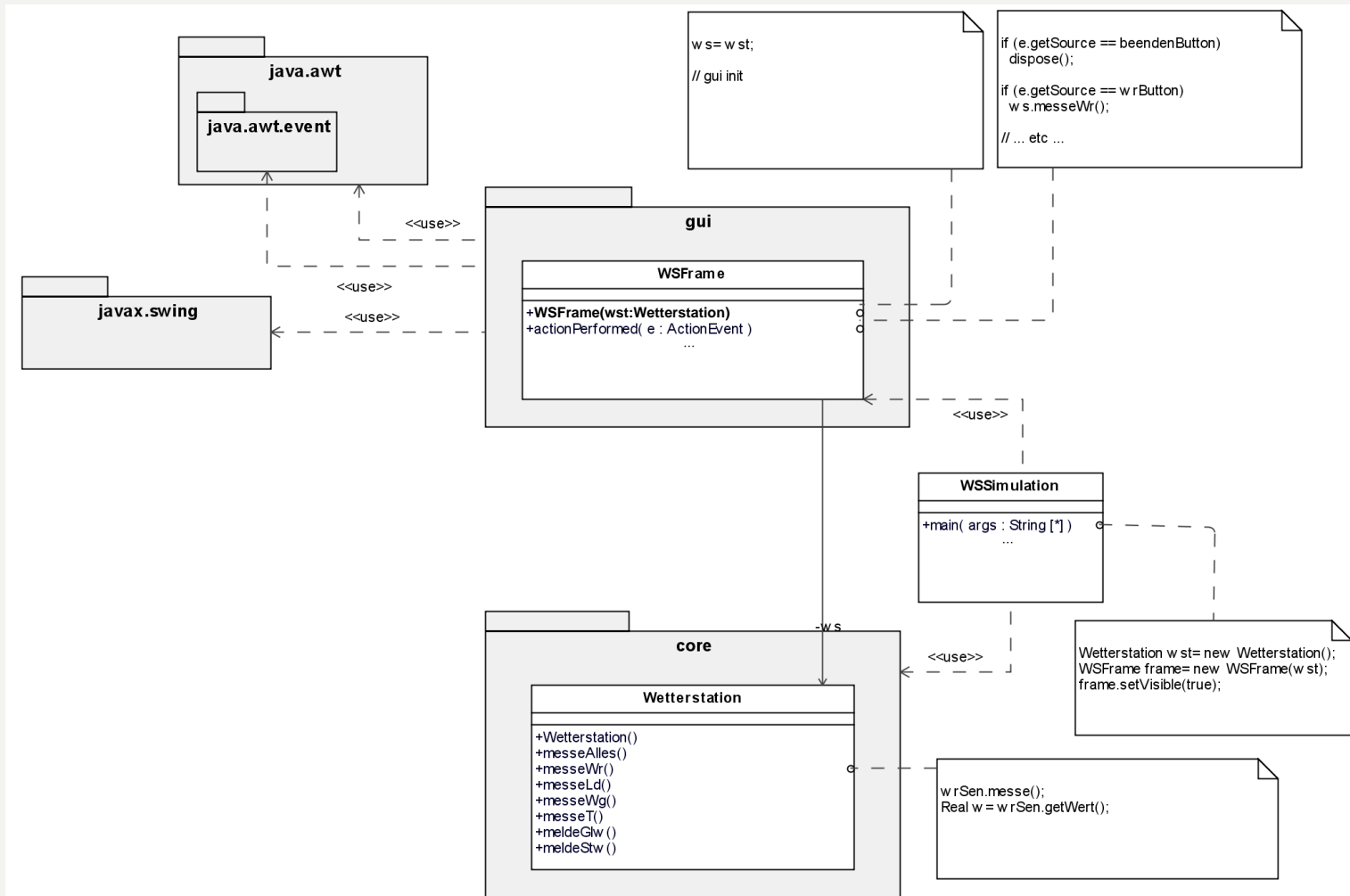
- Events enthalten i.d.R. Meta-Daten und Nutzdaten
- Event-Objekt muss i.d.R. selbst erstellt werden

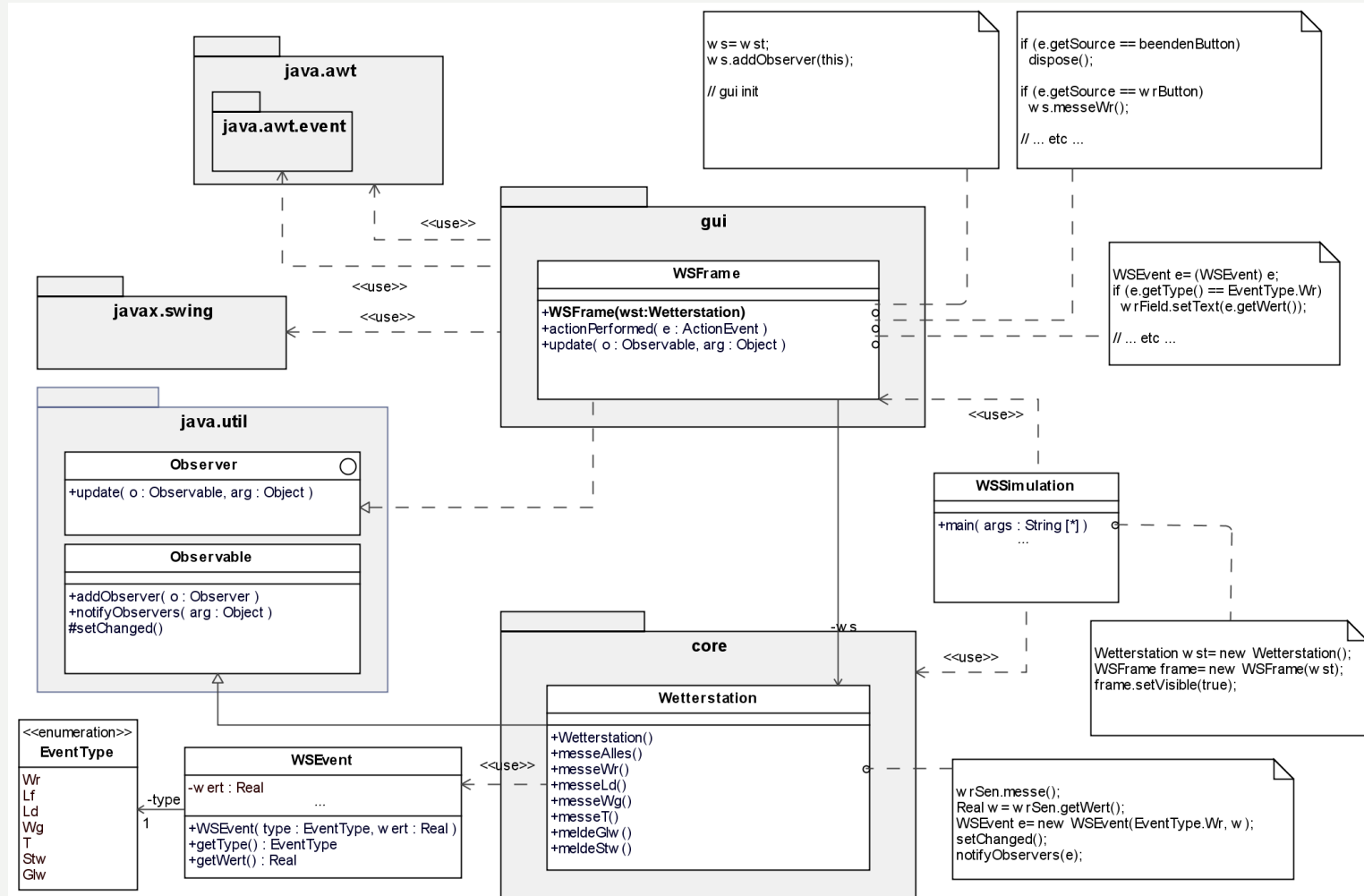
b) Implementierung des **Observables (MyApplication)**

- Subklasse von der Klasse `Observable` bilden
- Im geeigneten Moment: Events auf die Reise schicken (`setChanged()`, `notifyObservers()`).

c) Implementierung des **Observers (MyGUI)**

- Implementierung des Interface `Observer`
- Implementierung der Methode `update` (=> Reaktion auf Events...)
- Registrierung des Observers beim Observable (`addObserver()`, ggf. `removeObserver()`).





Übung 11

ENDE