

Lecture

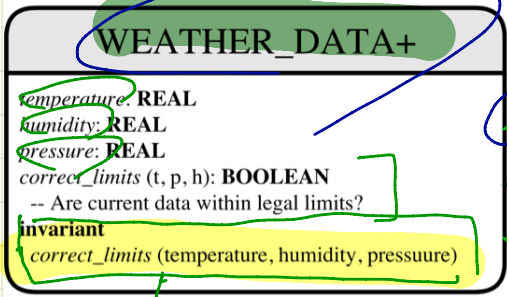
18

Monday

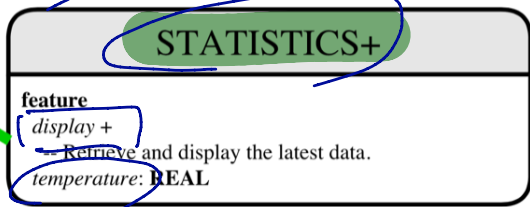
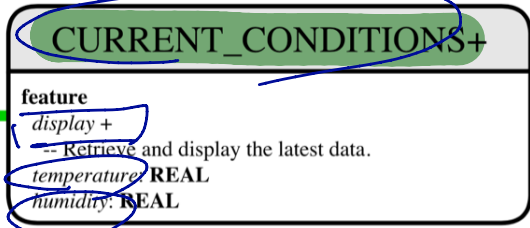
Nov. 13

supplier

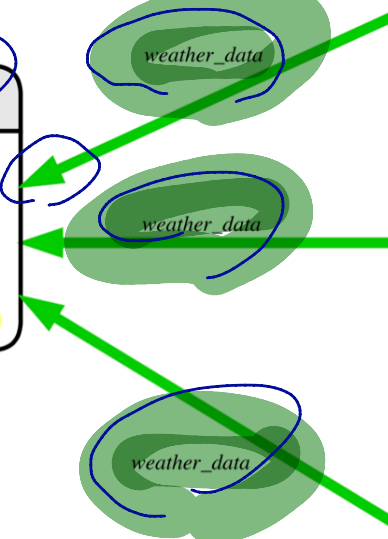
weather-data-temperature

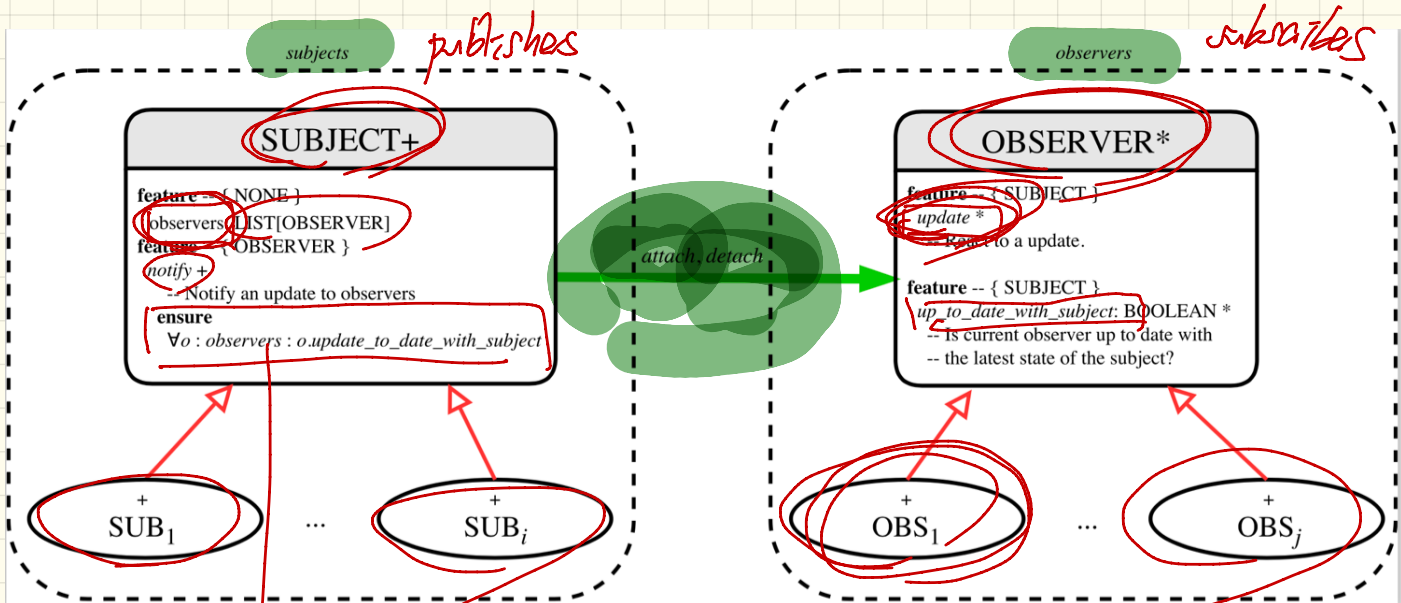


class invariant



clients





*notify*  
*ensure*  
 $\forall o : observers : o.up\_to\_date\_with\_subject$

subjects

observers

**SUBJECT+**

```

feature -- { NONE }
observers LIST(OBSERVER)
feature -- { OBSERVER }
notify +
-- Notify an update to observers
ensure
  ∀ o : observers : o.update_to_date_with_subject

```

ST.

attach, detach

**OBSERVER\***

```

feature -- { SUBJECT }
update *
-- React to a update.
feature -- { SUBJECT }
up_to_date_with_subject: BOOLEAN *
-- Is current observer up to date with
-- the latest state of the subject?

```

**WEATHER\_DATA+**

```

temperature: REAL
humidity: REAL
pressure: REAL
correct_limits (t, p, h): BOOLEAN
-- Are current data within legal limits?
invariant
  correct_limits (temperature, humidity, pressure)

```

+ FORECAST

+ CURRENT\_CONDITION

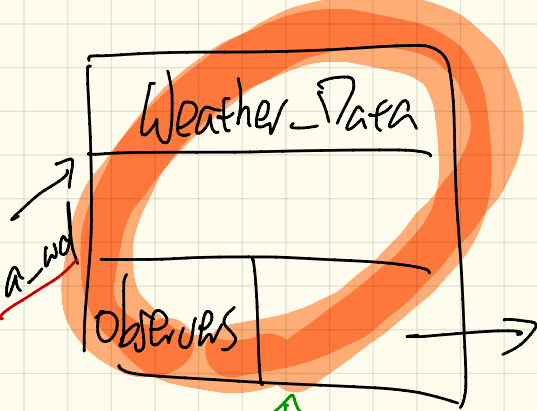
+ STATISTICS

wd

01: FORECAST  
02: STATISTICS

wd: WEATHER\_DATA  
wd.attach(01)  
wd.detach(02)

wd.notify\_observers [1].update  
observers [2].update



```
fd.wd = a_wd  
a_wd.observers[i] = fd
```



```
wd := a_wd  
wd.attach(fd)
```

