

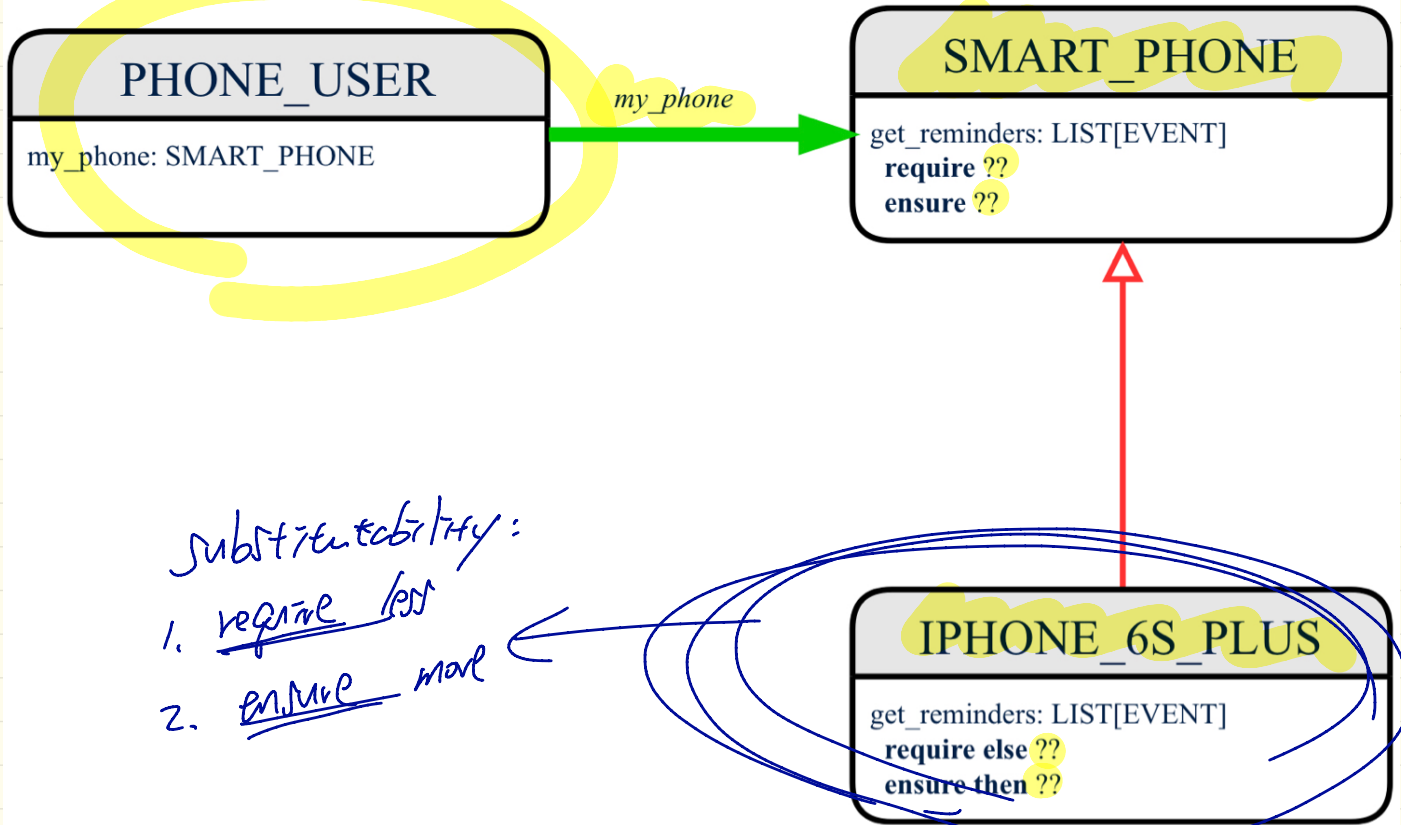
Thursday Nov. 15
Lecture 19

- Project

- Lab Test Nov. 28

- Lab 5 Next Wed.

Subcontracting: Architectural View



Substitutability:

1. require less
2. ensure more

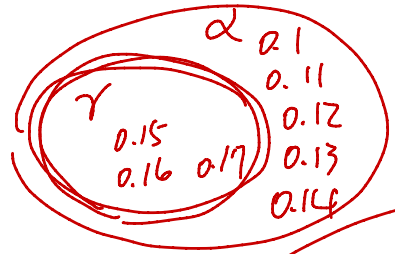
Subcontracting: Example (1)

$x \alpha \Rightarrow \gamma$
 $\alpha \Rightarrow \gamma$
 $\gamma \Rightarrow \alpha$

$x = 0.12$
 $\alpha: T$
 $\gamma: F$
 $\alpha \Rightarrow \gamma = T \Rightarrow F = F$
 $P \Rightarrow P \vee Q$
 δ
 β

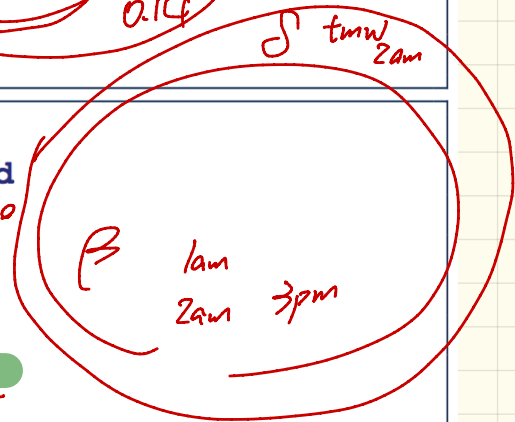
```

class SMART_PHONE
  get_reminders: LIST[EVENT]
  require
     $\alpha$ : battery_level  $\geq$  0.1 10%
  ensure
     $\beta$ :  $\forall e: \text{Result} \mid e \text{ happens today}$ 
end
  
```



```

class IPHONE_6S_PLUS
  inherit SMART_PHONE redefine get_reminders end
  get_reminders: LIST[EVENT]
  require else
     $\gamma$ : battery_level  $\geq$  0.15 15%
  ensure then
     $\delta$ :  $\forall e: \text{Result} \mid e \text{ happens today or tomorrow}$ 
  end
end
  
```



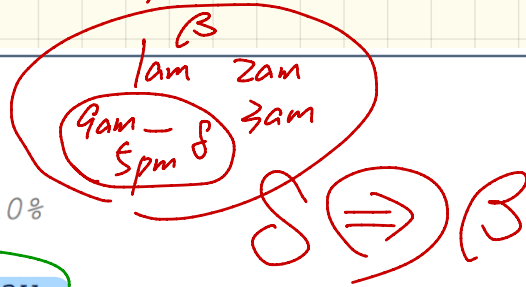
requiring more

Subcontracting: Example (2)

appropriate design of precond: $\alpha \Rightarrow \gamma$

```

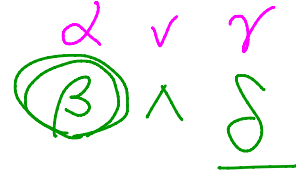
class SMART_PHONE
  get_reminders: LIST[EVENT]
  require
     $\alpha$ : battery_level  $\geq$  0.1 -- 10%
  ensure
     $\beta$ :  $\forall e$ : Result | e happens today
end
  
```



```

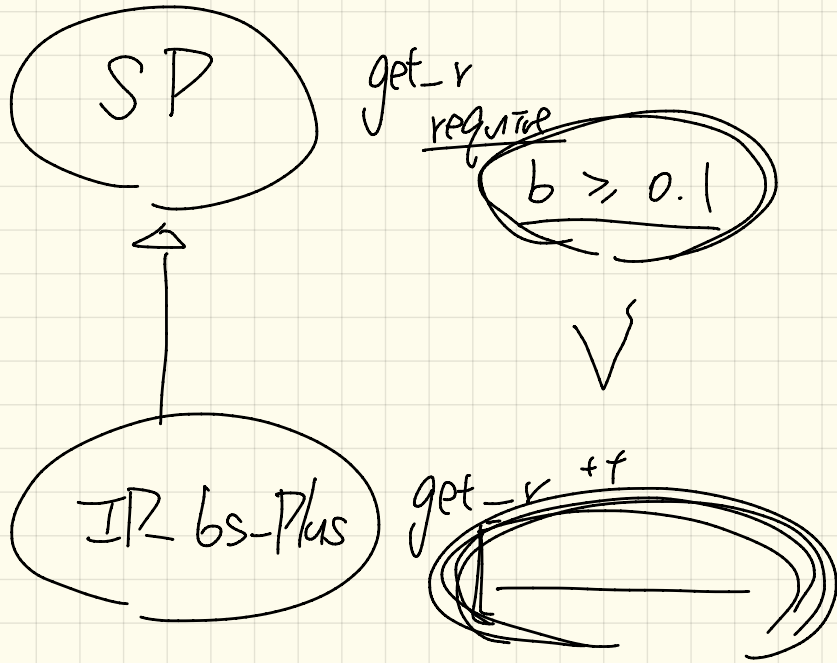
class IPHONE_6S_PLUS
  inherit SMART_PHONE redefine get_reminders end
  get_reminders: LIST[EVENT]
  require else
  require battery_level  $\geq$  0.5
  ensure then
     $\delta$ :  $\forall e$ : Result | e happens today between 9am and 5pm
  end
end
  
```

At runtime.



require else false

require else true



① Design Appropriate? Yes \because no precondition specified for descendant

② At runtime: $b \geq 0.1 \vee \text{false} \equiv b \geq 0.1$

parent_require
T

∨

child_require ✓

≡ T

parent_ensure
F

∧

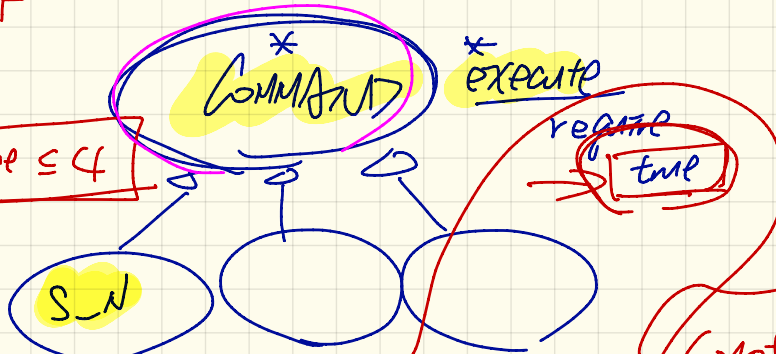
child_ensure

≡ F

7. At runtime:

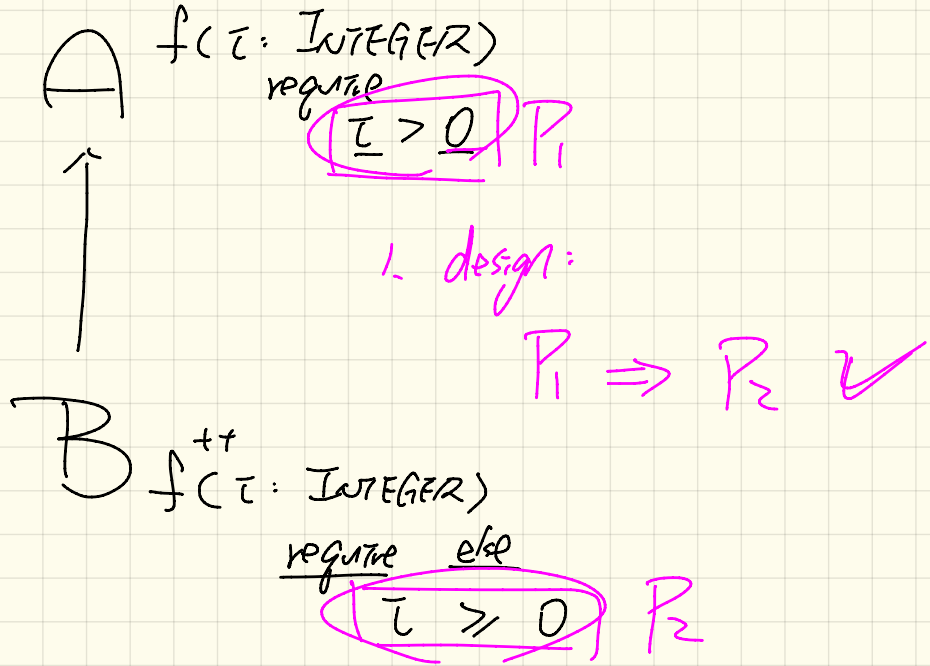
true (V) | $1 \leq \text{value} \leq 4$

≡ (True)



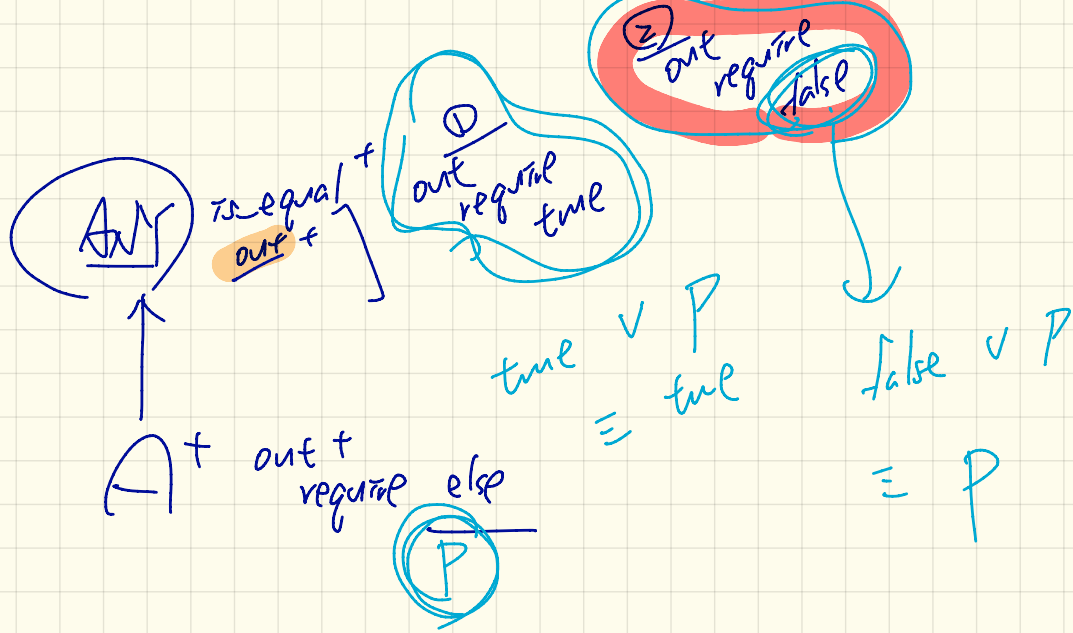
execute +
require else
1 ≤ value ≤ 4

7. 1. Not a good design ∴ requiring more

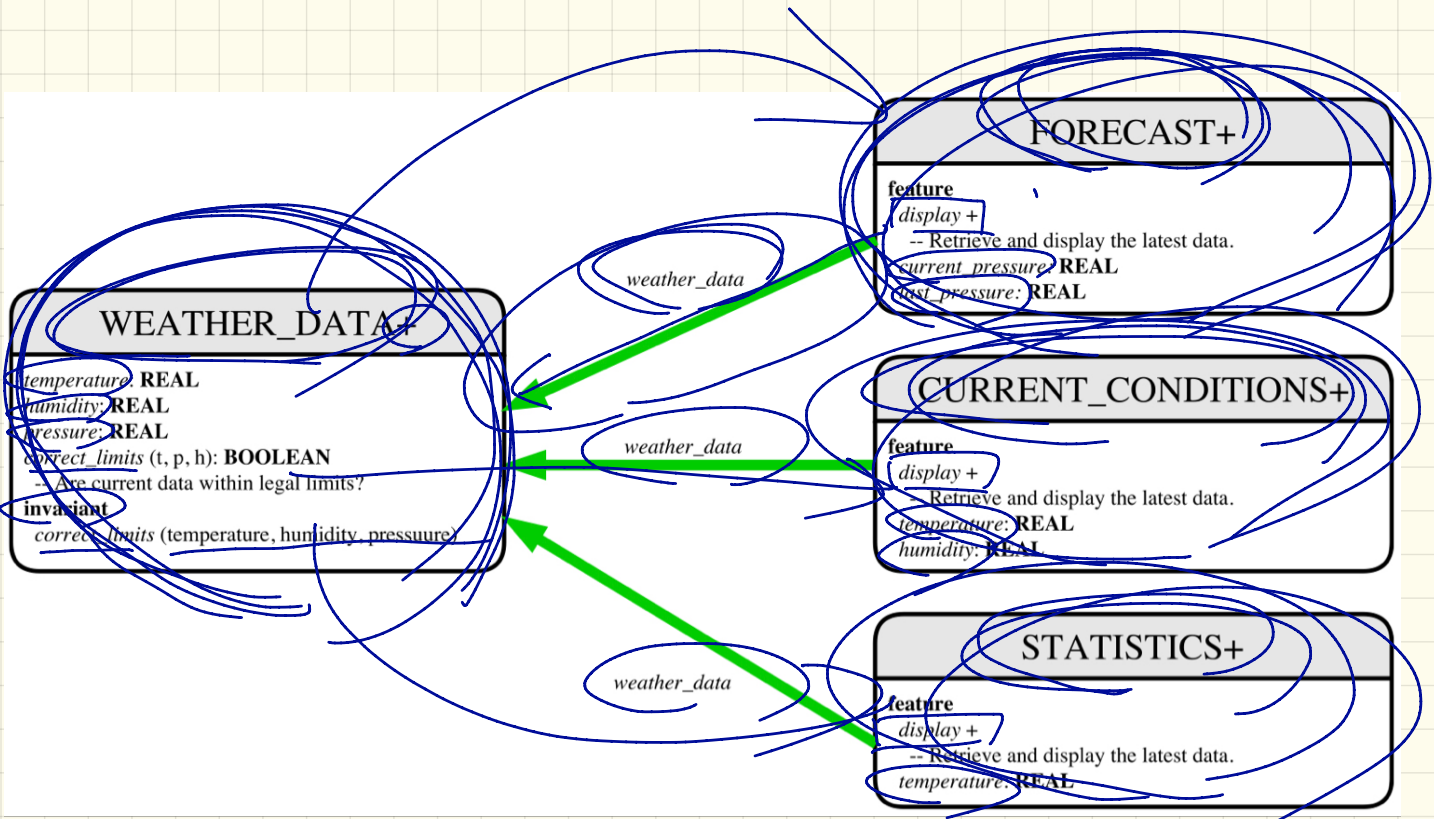


2. At runtime:

$$\underline{i > 0} \quad \vee \quad \underline{i \geq 0} \quad \equiv \quad i \geq 0$$



Weather Station: 1st Design



Weather Station: 1st Implementation

```
class WEATHER_DATA create make
feature -- Data
  temperature: REAL
  humidity: REAL
  pressure: REAL
feature -- Queries
  correct_limits(t,p,h: REAL): BOOLEAN
  ensure
    Result implies -36 <= t and t <= 60
    Result implies 50 <= p and p <= 110
    Result implies 0.8 <= h and h <= 100
feature -- Commands
  make (t, p, h: REAL)
  require
    correct_limits(temperature, pressure, humidity)
  ensure
    temperature = t and pressure = p and humidity = h
invariant
  correct_limits(temperature, pressure, humidity)
end
```

```
class FORECAST create make
feature -- Attributes
  current_pressure: REAL
  last_pressure: REAL
  weather_data: WEATHER_DATA
feature -- Commands
  make(wd: WEATHER_DATA)
  ensure weather_data = a.weather_data
  update
  do last_pressure := current_pressure
     current_pressure := weather_data.pressure
  end
  display
  do update
```

```
class CURRENT_CONDITIONS create make
feature -- Attributes
  temperature: REAL
  humidity: REAL
  weather_data: WEATHER_DATA
feature -- Commands
  make(wd: WEATHER_DATA)
  ensure weather_data = wd
  update
  do temperature := weather_data.temperature
     humidity := weather_data.humidity
  end
  display
  do update
```

```
class STATISTICS create make
feature -- Attributes
  weather_data: WEATHER_DATA
  current_temp: REAL
  max, min, sum_so_far: REAL
  num_readings: INTEGER
feature -- Commands
  make(wd: WEATHER_DATA)
  ensure weather_data = a.weather_data
  update
  do current_temp := weather_data.temperature
     -- Update min, max if necessary.
  end
  display
  do update
```

Weather Station: Testing 1st Design

```

class WEATHER_STATION create make
feature -- Attributes
  cc: CURRENT_CONDITIONS ; fd: FORECAST ; sd: STATISTICS
  wd: WEATHER_DATA
feature -- Commands
make
do
  create wd.make (9, 75, 25)
  create cc.make (wd) ; create fd.make (wd) ; create sd.make (wd)
  wd.set_measurements (15, 60, 30.4)
  cc.display ; fd.display ; sd.display
  cc.display ; fd.display ; sd.display
  wd.set_measurements (11, 90, 20)
  cc.display ; fd.display ; sd.display
end
end
  
```

```

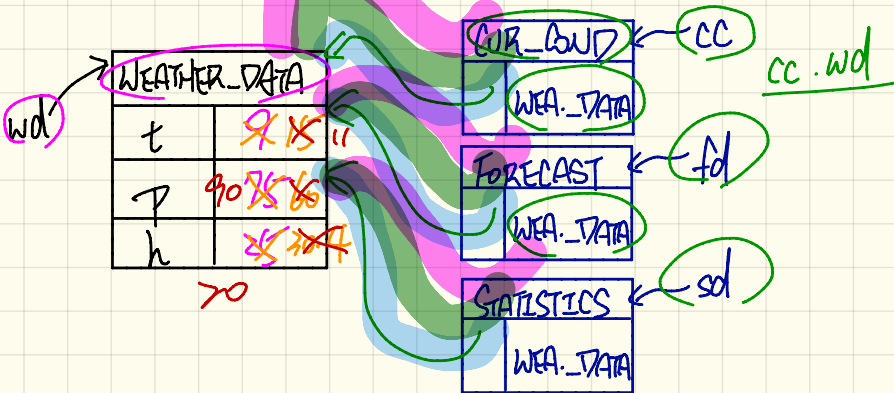
class FORECAST create make
feature -- Attributes
  current_pressure: REAL
  last_pressure: REAL
  weather_data: WEATHER_DATA
feature -- Commands
make (wd: WEATHER_DATA)
  ensure weather_data = a.weather_data
update
  do last_pressure := current_pressure
     current_pressure := weather_data.pressure
  end
display
  do update
  
```

```

class CURRENT_CONDITIONS create make
feature -- Attributes
  temperature: REAL
  humidity: REAL
  weather_data: WEATHER_DATA
feature -- Commands
make (wd: WEATHER_DATA)
  ensure weather_data = wd
update
  do temperature := weather_data.temperature
     humidity := weather_data.humidity
  end
display
  do update
  
```

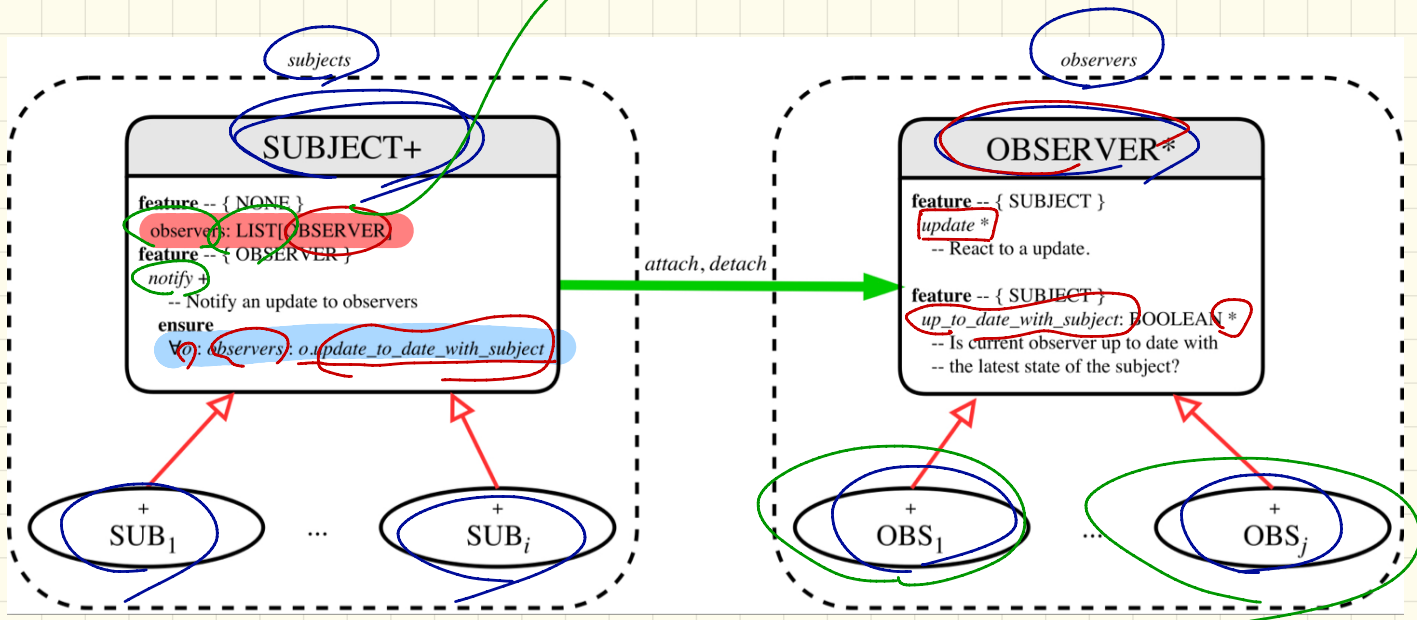
```

class STATISTICS create make
feature -- Attributes
  weather_data: WEATHER_DATA
  current_temp: REAL
  max, min, sum_so_far: REAL
  num_readings: INTEGER
feature -- Commands
make (wd: WEATHER_DATA)
  ensure weather_data = a.weather_data
update
  do current_temp := weather_data.temperature
     -- Update min, max if necessary.
  end
display
  do update
  
```



The Observer Pattern

Static Type: OBSERVER



Weather Station: Applying the Observer Pattern

