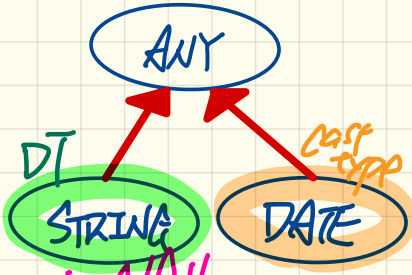
EECS3311 Software Design (Fall 2020)

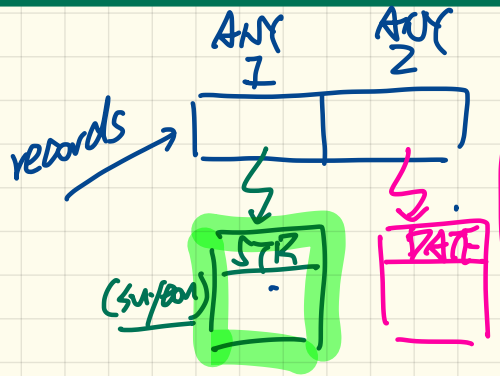
Q&A - Lecture Series W8

Tuesday, November 10

General Book: Retrieval from Polymorphic Arrary

```
1  birthday: DATE; phone_number: STRING
2  b: BOOK; is_wednesday: BOOLEAN
3  create {BOOK} b.make
4  phone_number := "416-677-1010"
5  b.add ("SuYeon", phone_number)
6  create {DATE} birthday.make(1975, 4, 10)
7  b.add ("Yuna", birthday)
```





FALSE ∵ DT STR cannot fulfill expect of DATE.

b.get("SuYeon")

ST: ANY DT: STR.

b.get("Yuna")

ST: ANY DT: DATE

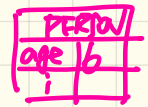
```
check attached {DATE} b.get("SuYeon") as suyeon_bday then
  is_wednesday := suyeon_bday.get_day_of_week = 4
end
```

Multiple Constructors

```
class PERSON
create make_with_name, make_with_age
feature -- Commands
  make_with_name (n: STR)
    do ... end
  make_with_age (a: INT)
    do ... end
end
```

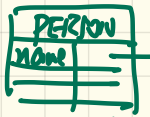
Java: ① constructor names must match class names ✗
② overloading is allowed ✗

```
class A {
  A() { . . }
  A(int i) { . . }
}
```



Eiffel.

p: PERSON



"Juna"

create p.make_with_name("Juna")

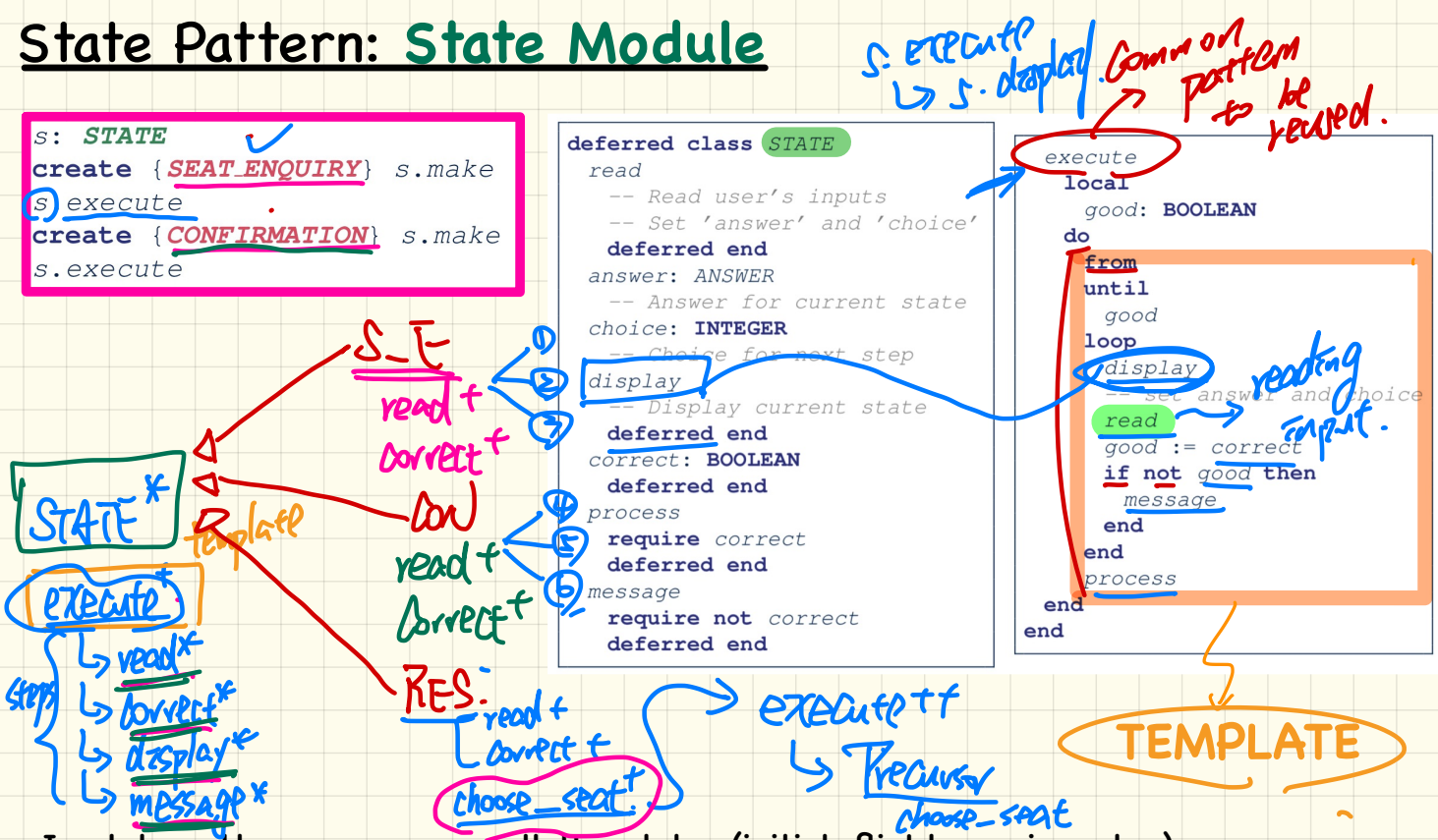
create p.make_with_age(6)

State Pattern: State Module

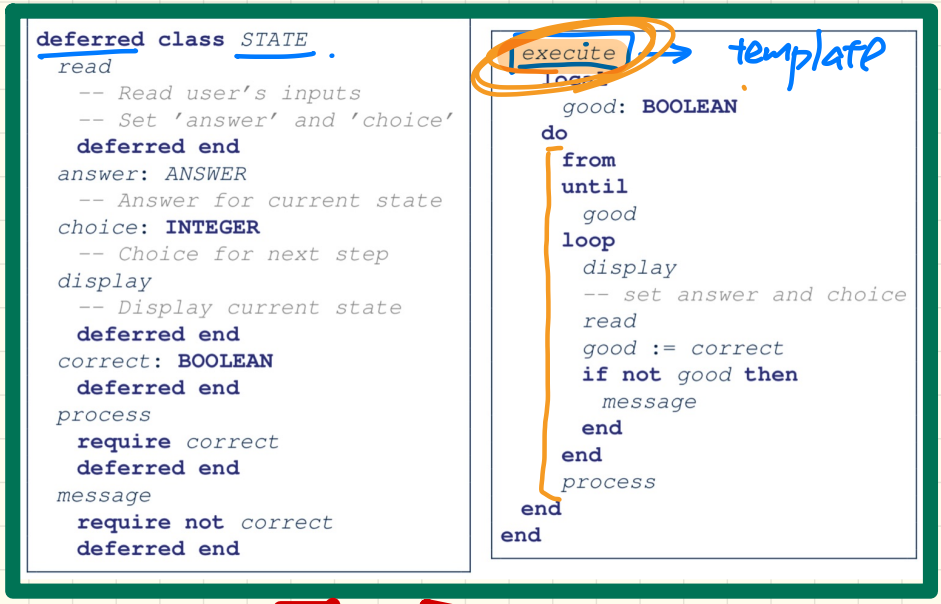
```
s: STATE
create {SEAT_ENQUIRY} s.make
s.execute
create {CONFIRMATION} s.make
s.execute
```

```
deferred class STATE
  read
    -- Read user's inputs
    -- Set 'answer' and 'choice'
    deferred end
  answer: ANSWER
    -- Answer for current state
  choice: INTEGER
    -- Choice for next step
  display
    -- Display current state
    deferred end
  correct: BOOLEAN
    deferred end
  process
    require correct
    deferred end
  message
    require not correct
    deferred end
```

```
execute
  local
    good: BOOLEAN
  do
    from
    until
      good
    loop
      display
      -- Set answer and choice
      read
      good := correct
      if not good then
        message
      end
    end
    process
  end
end
```



In state pattern we assume all the states (initial, flight enquiry, etc.)
all have sub steps display, read, message, process, etc.
what if different steps have different sub steps?

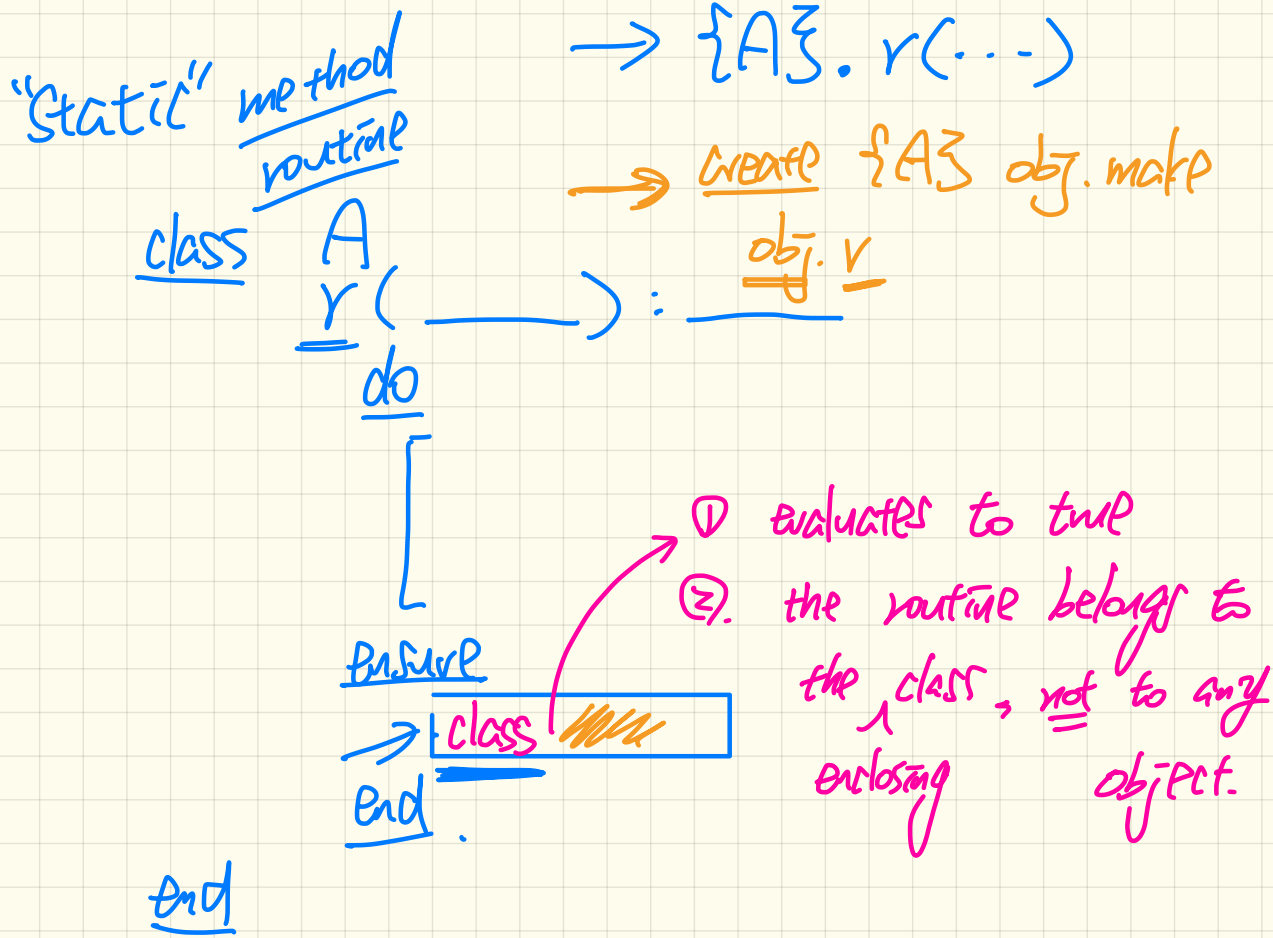


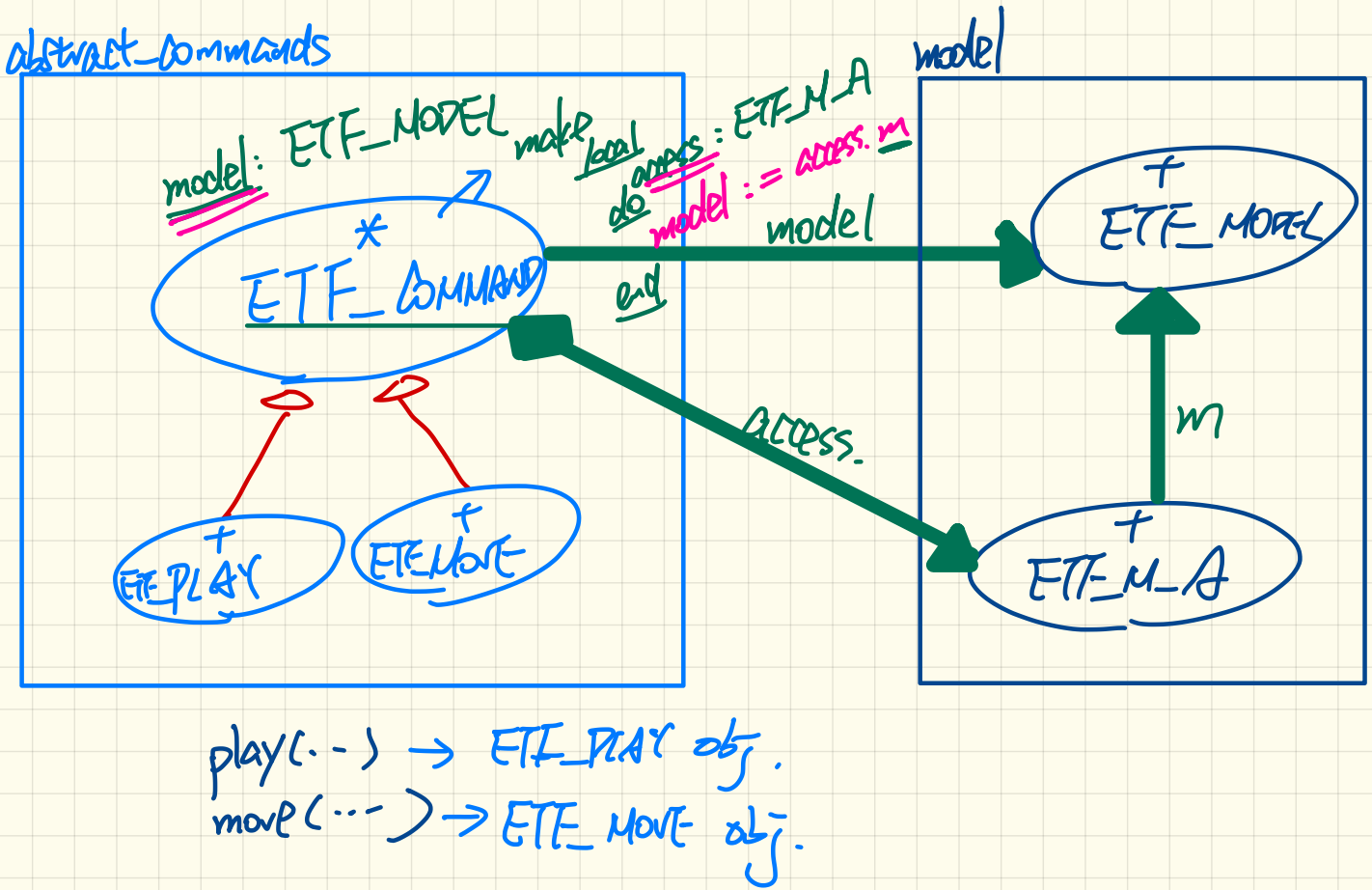
② execute ++
do
precursor
choose_window
end

① execute
reused completely.
+
SEAT_ENQUIRER

trigger to RES.
choose_window +
+
RESERVATION

① execute ++
do
choose_window
end


"static" method routine
→ {A}.r(...)
→ create {A} obj.make
obj.r
class A
r( ) :
do
ensure
class
end.
end
① evaluates to true
② the routine belongs to the class, not to any enclosing object.


abstract_commands
model
model: ETF_MODEL
make
local
do
model := access.m
end
model
access
m
play(...) → ETF_PLAY obj.
move(...) → ETF_MOVE obj.