EXAM REVIEW III

THURSDAY DECEMBER 13

Solution to Proving (1.2)

We first calculate the wp for the loop body to maintain the LI:

$$
\begin{aligned}
& wp(\texttt{if a[i] > Result then Result := a[i] end; i := i + 1},\ \boxed{\forall j \mid a.lower \le j \le i - 1 \bullet a.lower \le j \land j \le a.upper \land \textbf{Result} \ge a[j]})\\
= \ & \{wp \texttt{ rule for seq. comp. }\}\\
& wp(\texttt{if a[i] > Result then Result := a[i] end},\ \boxed{wp(\texttt{i := i + 1},\ \boxed{\forall j \mid a.lower \le j \le \overset{(i+1)-1}{i} \bullet a.lower \le j \land j \le a.upper \land \textbf{Result} \ge a[j]})})\\
= \ & \{wp \texttt{ rule for assignment}\}\\
& wp(\texttt{if a[i] > Result then Result := a[i] end},\ \boxed{\forall j \mid a.lower \le j \le i \bullet a.lower \le j \land j \le a.upper \land \textbf{Result} \ge a[j]})\\
= \ & \{wp \texttt{ rule for conditional}\}\\
& a[i] > \textbf{Result} \implies wp(\texttt{Result := a[i]},\ \boxed{\forall j \mid a.lower \le j \le i \bullet a.lower \le j \land j \le a.upper \land \textbf{Result} \ge a[j]})\\
& \land\\
& a[i] \le \textbf{Result} \implies wp(\texttt{Result := Result},\ \boxed{\forall j \mid a.lower \le j \le i \bullet a.lower \le j \land j \le a.upper \land \textbf{Result} \ge a[j]})\\
= \ & \{wp \texttt{ rule for assignment, twice}\}\\
& a[i] > \textbf{Result} \implies \forall j \mid a.lower \le j \le i \bullet a.lower \le j \land j \le a.upper \land \textbf{a[i]} \ge a[j]\\
& \land\\
& a[i] \le \textbf{Result} \implies \forall j \mid a.lower \le j \le i \bullet a.lower \le j \land j \le a.upper \land \textbf{Result} \ge a[j]
\end{aligned}
$$

$\checkmark\ a[i]$

$\cdot P$

$\land$

$\cdot Q$

We then prove that the precondition (i.e., $\neg$(exit condition) and LI) is no weaker than the above calculated wp:

- To prove:

$\neg B$ $\quad$ LI

$\neg B \land LI \Rightarrow (P \land Q)$

$$
\neg(i > a.upper) \land (\ \forall j \mid a.lower \le j \le i - 1 \bullet a.lower \le j \land j \le a.upper \land \textbf{Result} \ge a[j]\ ) \implies a[i] > \textbf{Result} \implies \boxed{\forall j \mid a.lower \le j \le i \bullet a.lower \le j \land j \le a.upper \land a[i] \ge a[j]}
$$

$(\neg B \land LI \Rightarrow P) \land (\neg B \land LI \Rightarrow Q)$

Proof:

$$
\begin{aligned}
& \boxed{\forall j \mid a.lower \le j \le i \bullet a.lower \le j \land j \le a.upper \land a[i] \ge a[j]}\\
\equiv \ & \{\texttt{split range: } \forall j \mid a.lower \le j \le i \bullet P(j) \equiv (\forall j \mid a.lower \le j \le i - 1 \bullet P(j)) \land P(i)\}\\
& (\forall j \mid a.lower \le j \le \mathbf{i - 1} \bullet a.lower \le j \land j \le a.upper \land a[i] \ge a[j]) \land (a.lower \le \mathbf{i} \land \mathbf{i} \le a.upper \land a[i] \ge a[\mathbf{i}])\\
\equiv \ & \{\texttt{antecedent: } a[i] > \texttt{Result}\texttt{; and RHS of precond: } \forall j \mid a.lower \le j \le i - 1 \bullet a.lower \le j \land j \le a.upper \land \texttt{Result} \ge a[j]\}\\
& true \land (a.lower \le i \land i \le a.upper \land a[i] \ge a[i])\\
\equiv \ & \{\texttt{LHS of precond: } \neg(i > a.upper) \texttt{ and } a[i] \ge a[i] \equiv true\}\\
& true
\end{aligned}
$$

Given.

$wp(x := e, \ldots)$

$wp(\text{if} \ldots, \ldots)$

$wp(S_1 ; S_2, \ldots)$

$\Rightarrow$ POs. X

$\{Q\} \, S \, \{R\}$

$\equiv \quad Q \Rightarrow wp(S, R)$

```
{x > 0 ∧ y > 0}
if x > y then
  bigger := x ; smaller := y
else
  bigger := y ; smaller := x
end
{bigger ≥ smaller}
```

(bigger → b, smaller → s)

$x \geq y \Rightarrow x > y$ ✗

$x \leq y \Rightarrow y \geq x$ ✓

$wp(b := x ; s := y, b \geq s)$

$= \{$ rule for ; $\}$

$wp(b := x, wp(s := y, b \geq s))$

$= \{$ rule for := $\}$

$wp(b := x, b \geq y)$

$= \{$ rule for := $\}$

$x \geq y$

0. $\{x > 0 \wedge y > 0\}$ if $x > y$ then $b := x ; s := y$ else $b := y ; s := x$ $\{b \geq s\}$

② $x > 0 \wedge y > 0 \Rightarrow T$

1. $wp($if $x > y$ then $b := x ; s := y$ else $b := y ; s := x, b \geq s)$

$= \{$ wp rule for if... $\}$

T $\boxed{x > y \Rightarrow wp(b := x ; s := y, b \geq s)}$ ($x \geq y$)

$\wedge$

T $\boxed{\neg(x > y) \Rightarrow wp(b := y ; s := x, b \geq s)}$ ($\neg(x > y)$: $x \leq y$; $b \geq s$: $y \geq x$)

$x > y \Rightarrow x \geq y$

$\equiv x > y \Rightarrow x > y \vee x = y$

$$\forall x \mid 1 \leq x \leq 5 \cdot x^2 \geq 3$$

$$\equiv (1^2 \geq 3 \wedge 2^2 \geq 3 \wedge 3^2 \geq 3 \wedge 4^2 \geq 3) \wedge 5^2 \geq 3$$

$$\equiv \underbrace{(\forall x \mid 1 \leq x \leq 4 \cdot x^2 \geq 3)}_{F} \wedge \underbrace{5^2 \geq 3}_{T}$$

$$\underline{\hspace{6em}}$$

$$F$$

$$(\forall x \mid i \le x \le j \cdot P(x))$$

$$\equiv (\forall x \mid i \le x \le j-1 \cdot P(x)) \wedge P(j)$$

$$(\exists x \mid i \le x \le j \cdot P(x))$$

$$\equiv (\exists x \mid i \le x \le j-1 \cdot P(x)) \vee P(j)$$

Given that the loop is not ready to exit, and that the LI has been maintained by previous iterations, the current iteration maintains the LI.

$\{\neg B \wedge LI\}\ S_{body}\ \{LI\}.$

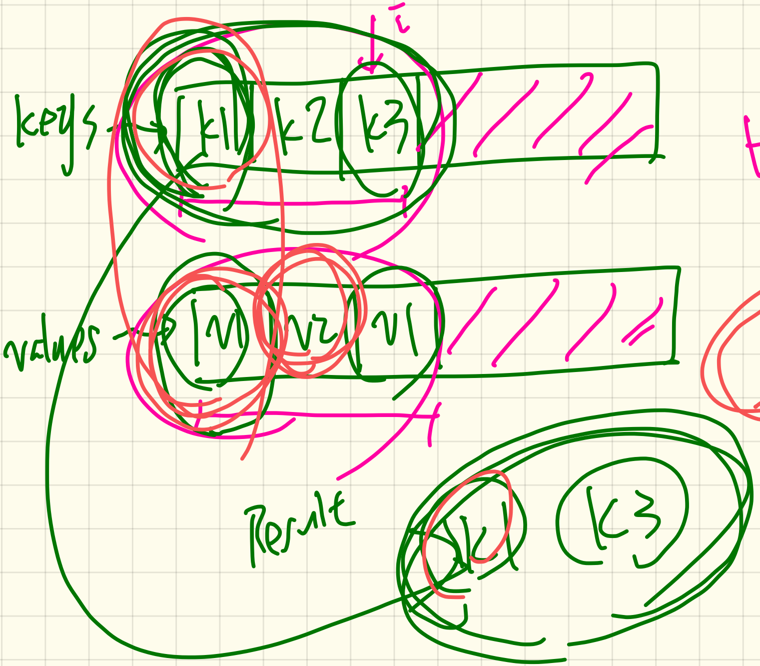
from
S_{init}
invariant
LI
until
B
loop
S_{body}
variant
V
end

Postcondition.

get_keys(v1).

result_valid: $\forall k \mid k \in$ **Result** $\bullet$ *model.item(k)* $\sim$ *v*

no_missing_keys: $\forall k \mid k \in$ *model.domain* $\bullet$ *model.item(k)* $\sim$ *v* $\Rightarrow k \in$ **Result**



keys[j] values[j]

$\forall j \mid 1 \le j \le i \bullet$ (or $i-1$.)

values[j] $\sim$ v1

and

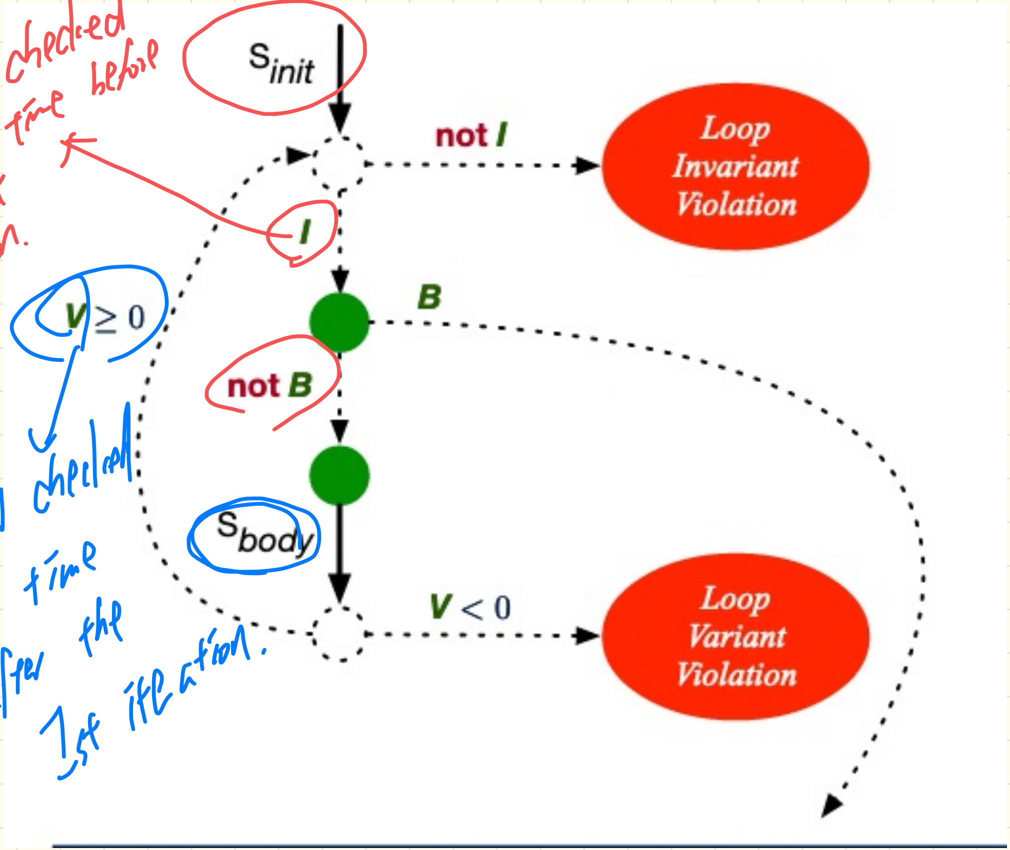
implies

Result.has(keys[j])

ITERABLE

across.

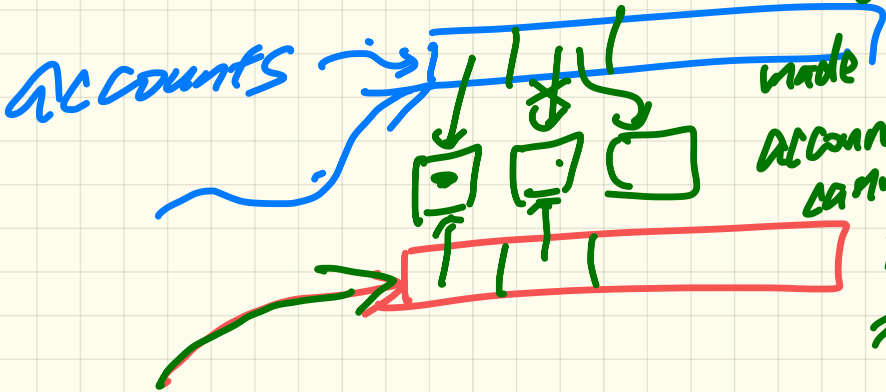
LI checked first time before the 1st iteration.

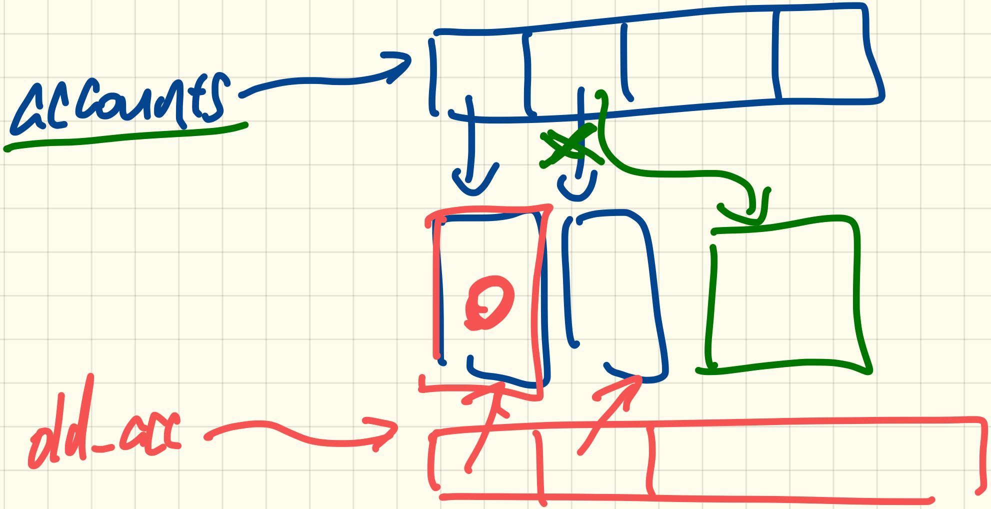


LV checked 1st time after the 1st iteration.

- `accounts = old accounts` T
- `accounts = old accounts.twin` F

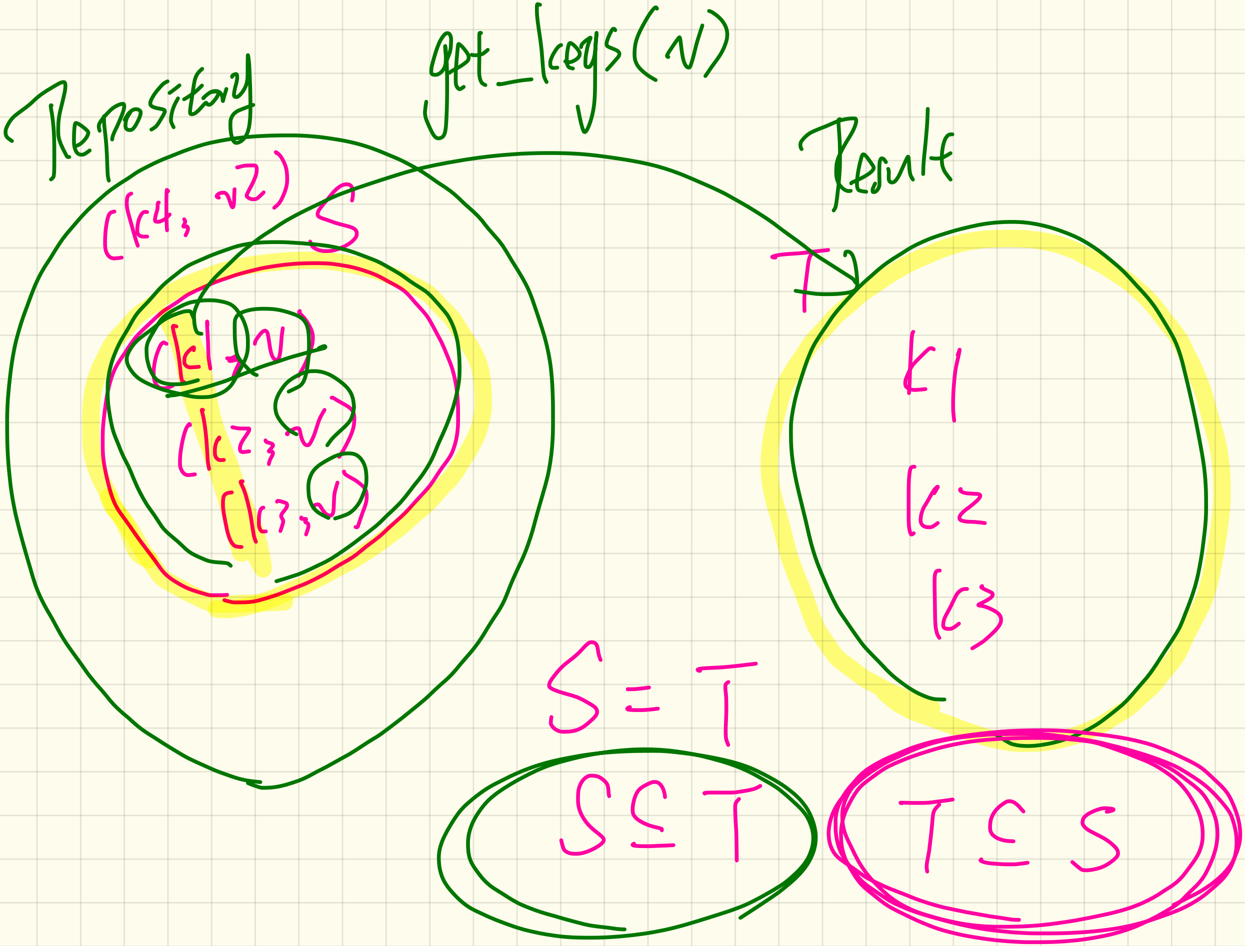
accounts ~ old accounts.twin. not appr ∵ changes made on account cannot be identified.



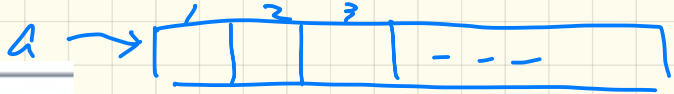


① accounts ~ old accounts.twin

② accounts ~ old accounts.deep_twin



Repository
get_keys(v)
Result
(k4, v2)
S
(k1, v)
(k2, v)
(k3, v)
T
k1
k2
k3
S = T
S ⊆ T
T ⊆ S



$\{ \text{True} \}$

$i := a.lower$

Result $:= a[i]$ $\quad \leq i$

$\{ \forall j \mid a.lower \leq j < i \bullet Result \geq a[j] \}$

$$\forall x \mid F \bullet P(x) \equiv T$$

1. $wp(\, i := a.lower \,;\, Result := a[i],\ \forall j \mid a.lower \leq j < i \bullet Result \geq a[j])$

$= \{ \text{rule for } ; \}$

$wp(\, \underline{i} := \underline{a.lower} \,;\, wp(\underline{R} := \underline{a[i]},\ \forall j \mid a.lower \leq j < \cancel{i} \bullet \cancel{R} \geq a[j]))$

$a.lower \quad a[i] \quad a.lower$

$= \{ \text{rule for } ; \}$

$$\boxed{\forall j \mid \underbrace{a.lower \leq j < a.lower}_{F} \bullet a[i] \geq a[j]} \equiv T$$

$a.lower$

$\cancel{a.lower} \leq j \wedge j < \cancel{a.lower}$

EVENT

wd.change_on_temperature.subscribe (~~agent~~ update_temperature)

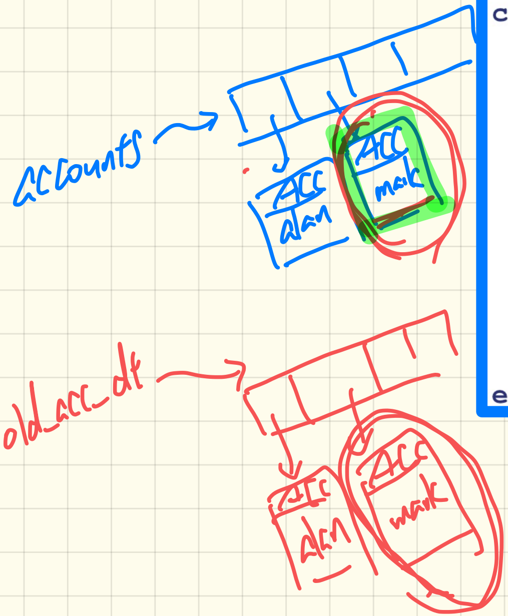
wd.change_on_humidity.subscribe (agent update_humidity)

Type? void

&f

*

Testing of Postcondition: Exercise



alan

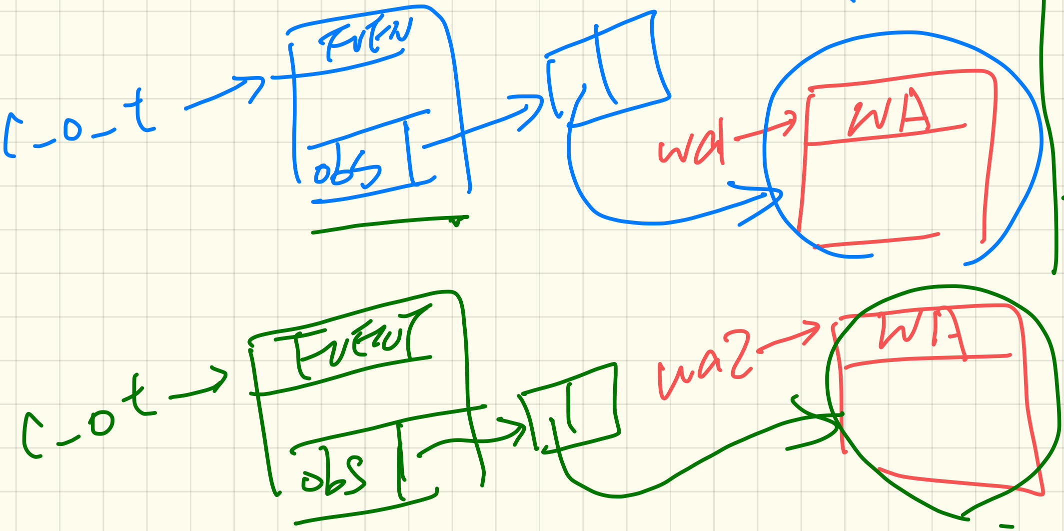
```
class BANK
  deposit_on_v5 (n: STRING; a: INTEGER)
    do ... -- Put Correct Implementation Here.
    ensure
      ... balance = old balance
      others_unchanged :
        across old accounts.deep_twin as cursor
        all cursor.item.owner /~ n implies
              cursor.item ~ account_of (cursor.item.owner)
        end
    end
end
```

TEST

```
class BAD_BANK_DEPOSIT
inherit BANK redefine deposit end
feature -- redefined feature
  deposit_on_v5 (n: STRING; a: INTEGER)
    do Precursor (n, a)                2
      accounts[accounts.lower].deposit(a)
    end
end
```

```
change_on_temperature : EVENT[TUPLE[REAL]] once create Result end
change_on_humidity : EVENT[TUPLE[REAL]] once create Result end
change_on_pressure : EVENT[TUPLE[REAL]] once create Result end
```

do



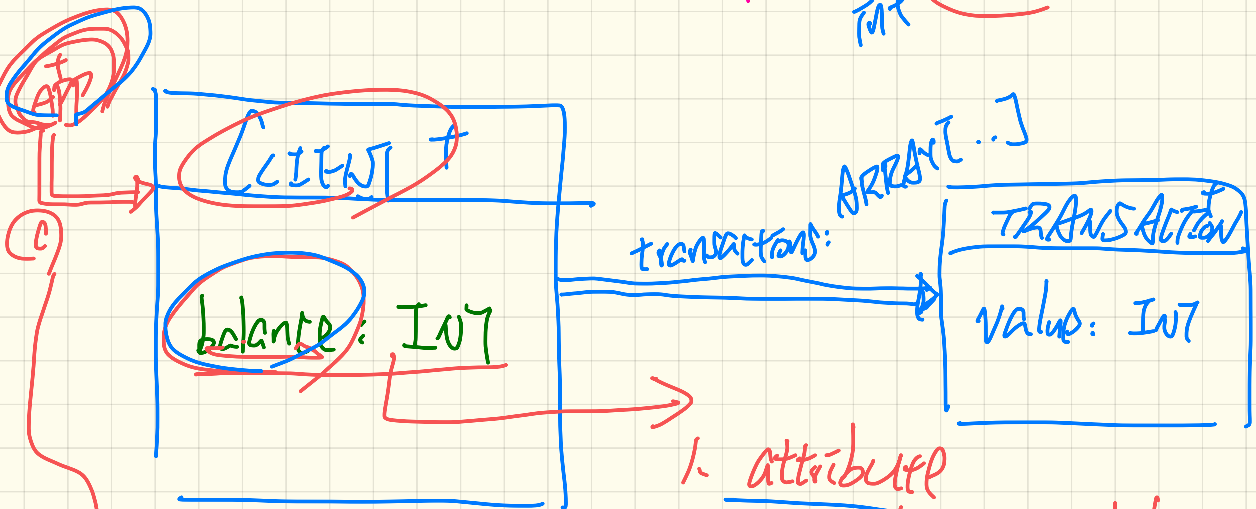
class Foo
create
default_create
feature
i : INT.

obj : Foo
create obj.d_c
create obj.

Uniform Access Principle

int balance

int balance()



1. attribute

balance := balance + - -

c: CLIENT

~~c.balance~~ := 200

2. query

do across ts is

end