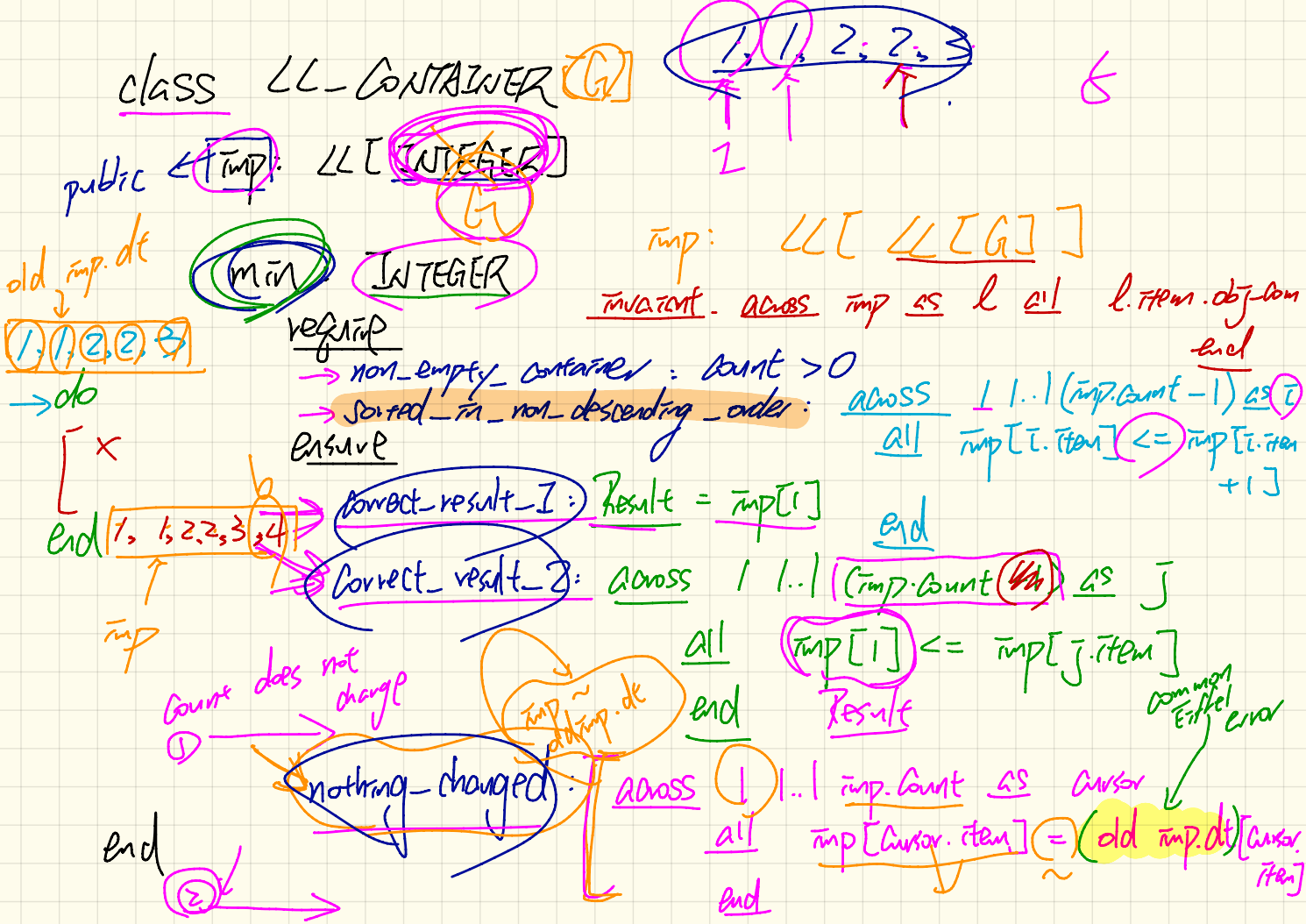
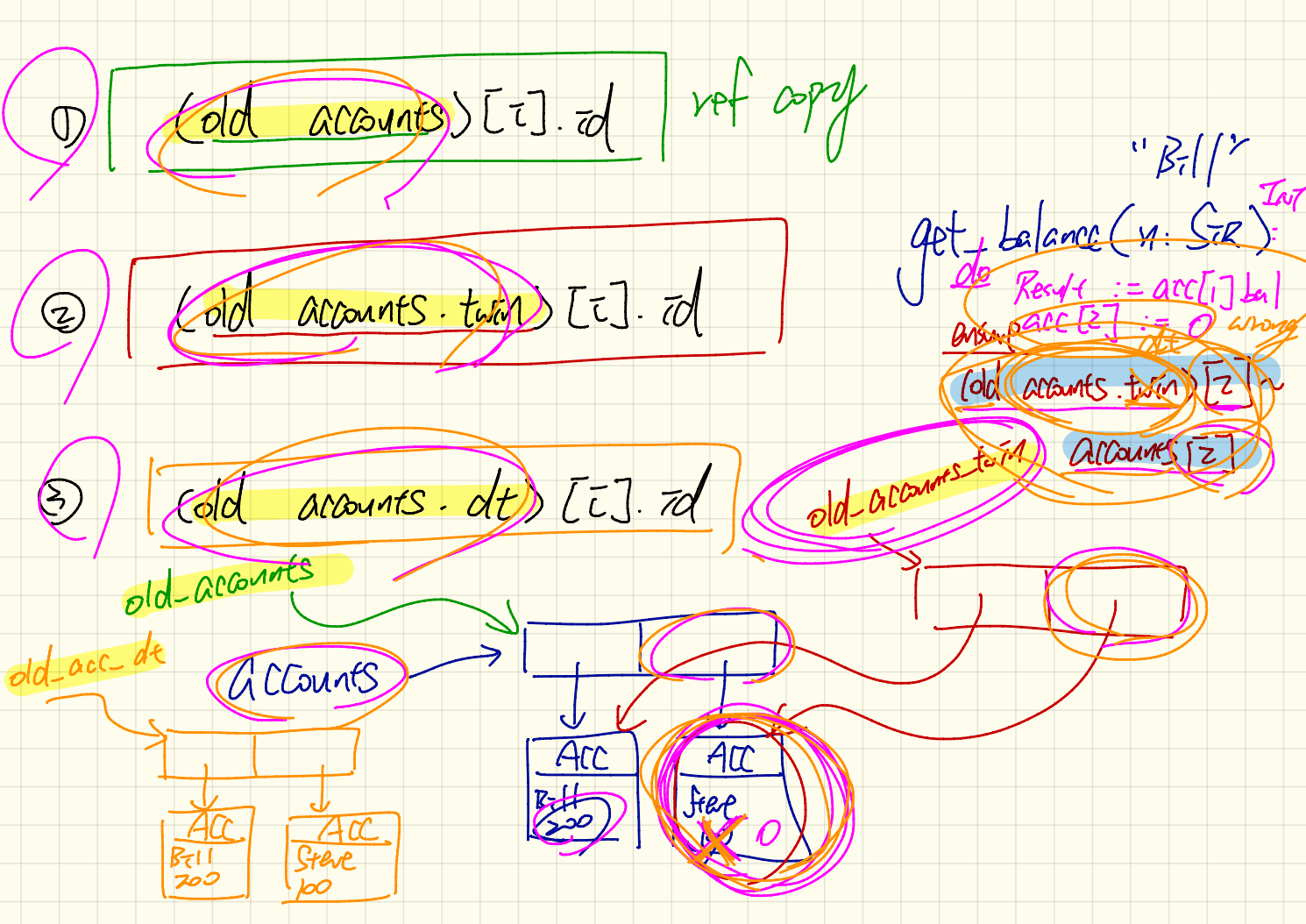
Friday Dec. 7

Exam Review Session 2

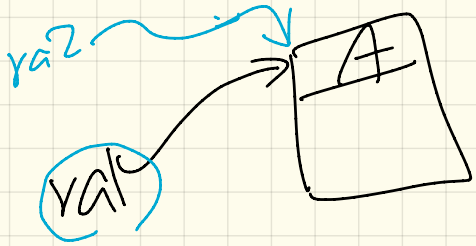

class LL_CONTAINER [G]
public tmp: LL [INTEGER]
G
min
INTEGER
1, 1, 2, 2, 3
1
tmp: LL [LL [G]]
invariant across tmp as l all l.item.obj_count end
old tmp.dt
1, 1, 2, 2, 3
do
require
non_empty_container: count > 0
sorted_in_non_descending_order:
across 1 |..| (tmp.count - 1) as i all tmp[i.item] <= tmp[i.item + 1] end
ensure
correct_result_1: Result = tmp[1]
correct_result_2: across 1 |..| (tmp.count) as j all tmp[1] <= tmp[j.item] end
Result
1, 1, 2, 2, 3, 4
tmp
end
count does not change
nothing_changed:
tmp ~ old tmp.dt
across 1 |..| tmp.count as cursor all tmp[cursor.item] ~ (old tmp.dt)[cursor.item] end
common Eiffel error
end


1
(old accounts)[i].id
ref copy
2
(old accounts.twin)[i].id
3
(old accounts.dt)[i].id
"Bill"
get_balance(x: STR): INT
do
Result := acc[i].bal
acc[z] := 0
(old accounts.twin)[z]
accounts[z]
old_accounts_twin
old_accounts
old_acc_dt
accounts
ACC
Bill
200
ACC
Steve
100
ACC
free
0

class A

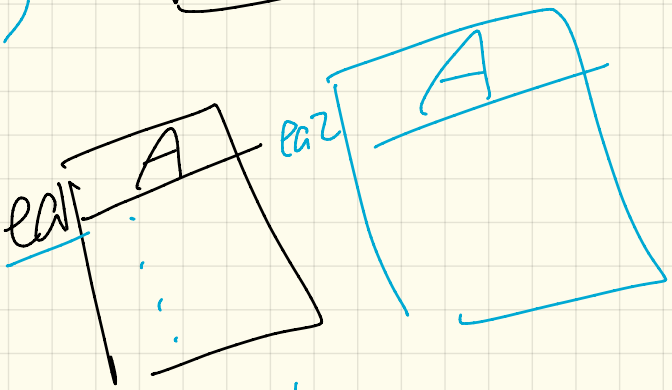
end

ra2 := ra1




ra2, ra1 : A

ea2, ea1 : expanded A




ea2 := ea1

Is this program correct?

1. Formulate the program as a Hoare Triple

```
x: Int
y: Int
swap
do
    x := x + y
    y := x - y
    x := x - y
ensure
    - x = old y
    - y = old x
end
```

$\{True\}\ x := x+y;\ y := x-y;\ x := x-y\ \{x = y_0 \wedge y = x_0\}$

2. $wp(x := x+y;\ y := x-y;\ x := x-y,\ x = y_0 \wedge y = x_0)$

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

$wp(x := x+y,\ wp(y := x-y;\ x := x-y,\ x = y_0 \wedge y = x_0))$

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

$wp(x := x+y,\ wp(y := x-y,\ wp(x := x-y,\ x = y_0 \wedge y = x_0)))$

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

$wp(x := x+y,\ wp(y := x-y,\ x - y = y_0 \wedge y = x_0))$

3. $True \Rightarrow$ ?

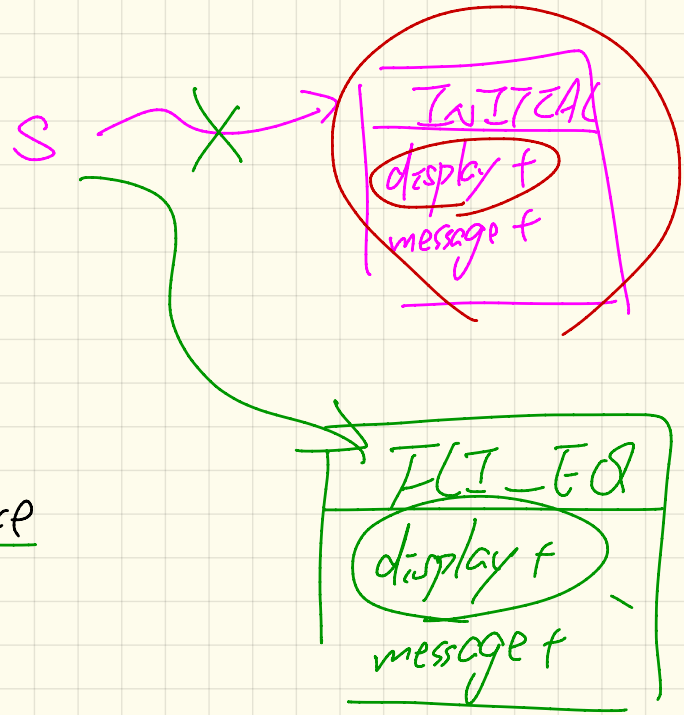
S : STATE

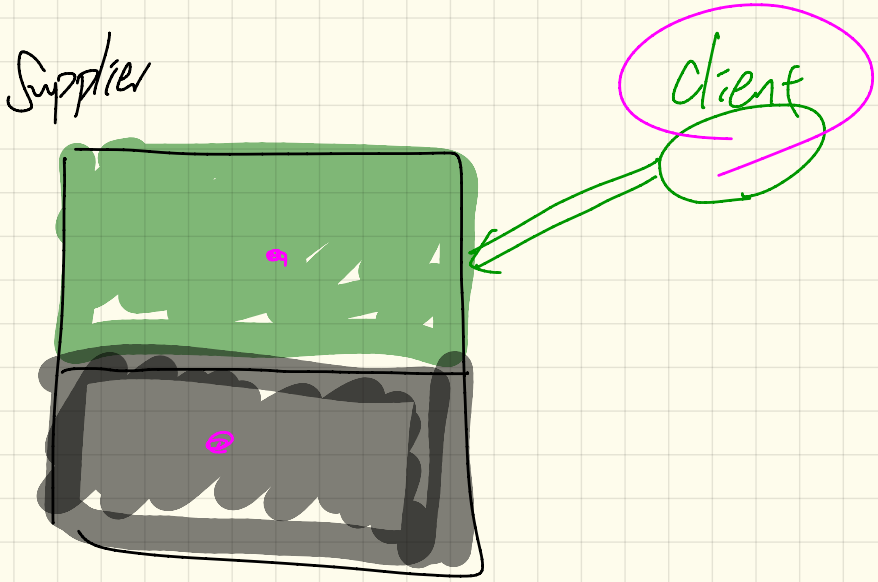
create {INITIAL} s.make

s.execute ← S.display

create {FLIGHT_ENQ} s.make

s.execute ← S.display




Supplier
client