Review Lecture

Thursday August 23

MaxSum

maxSum

23

0	2	-4	3	7	6	-2	-4	
1	20	1	-2	4				
2	7	1	1	1	1	2	2	3

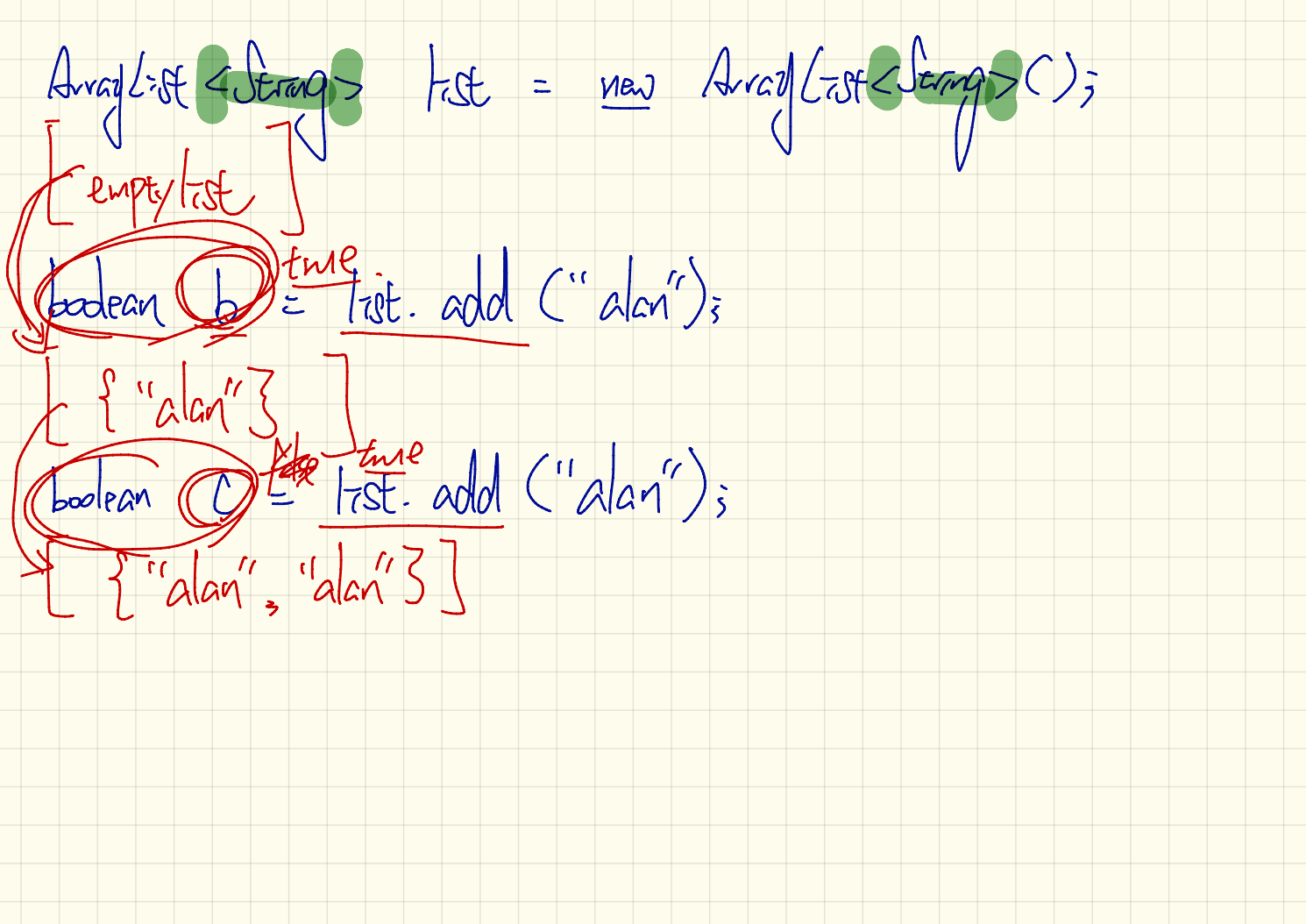
Analogy - find max in a 1D array.

-4	6	-2	20	20	100	-200

maxOfA

100

```
int max(int[] a) {
    int maxOfA;   -> int value ?
    [
    return maxOfA;
}
```


ArrayList<String> list = new ArrayList<String>();
[empty list]
boolean b = list.add("alan");
true
[{"alan"}]
boolean c = list.add("alan");
true
[{"alan", "alan"}]

Short Circuit

- evaluate < to R | Logic
- In case &&:
  - if < is F, overall is F anyway
  - skip evaluating R

boolean b1
boolean b2

conjunction ("and")

b1	b2	b1 and b2
T	T	
T	F	
F	T	
F	F	

guarding condition

b1 (&&) b2

"conjunction"

0 F

crash

e.g. x != 0 && y / x > 2

say: y is 4, x is 0

y / x > 2 (&&) x != 0

Say: x is 0, y is 4

`x == 0 || y/x > 2`

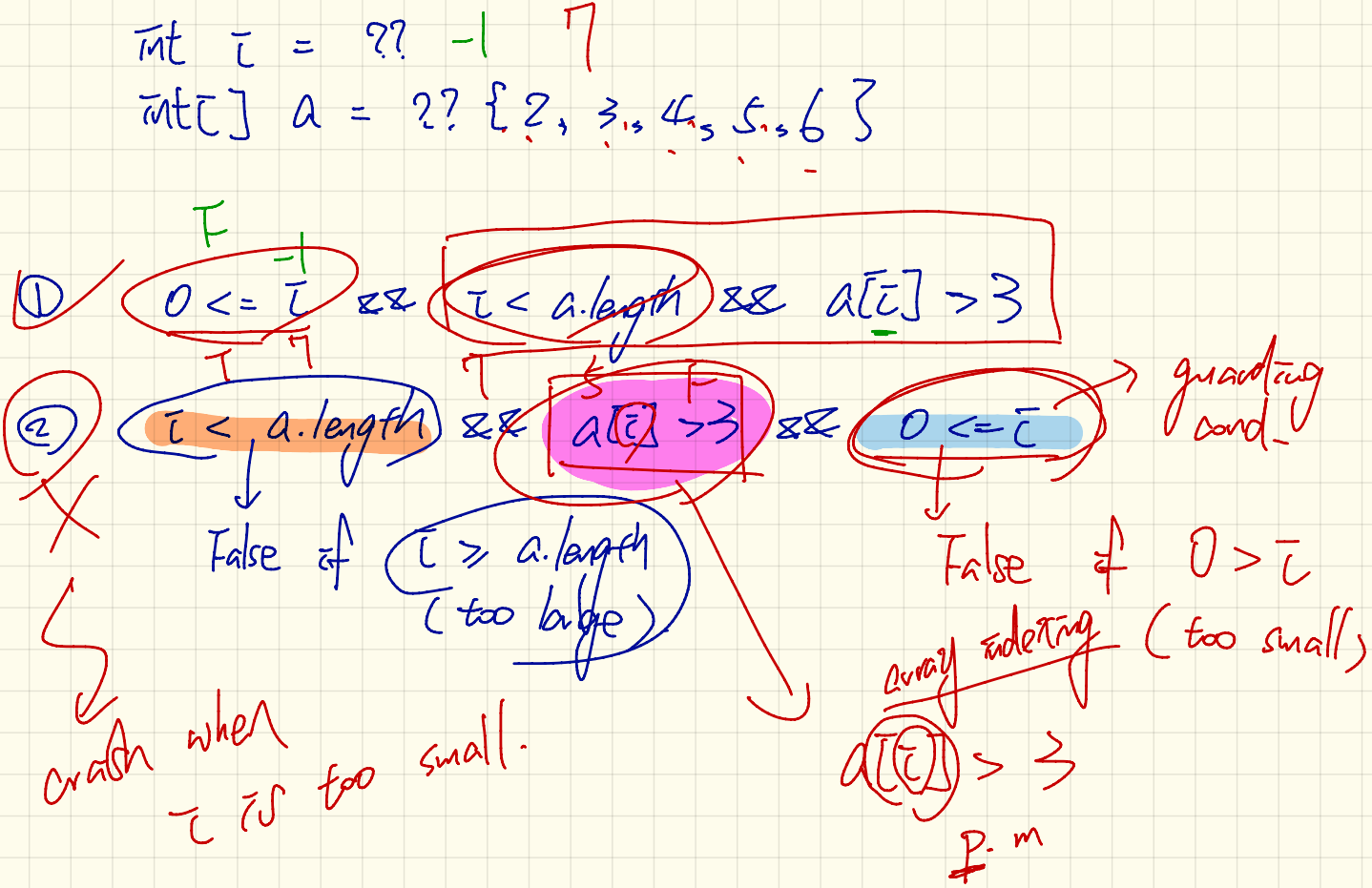
`x == 0` → T

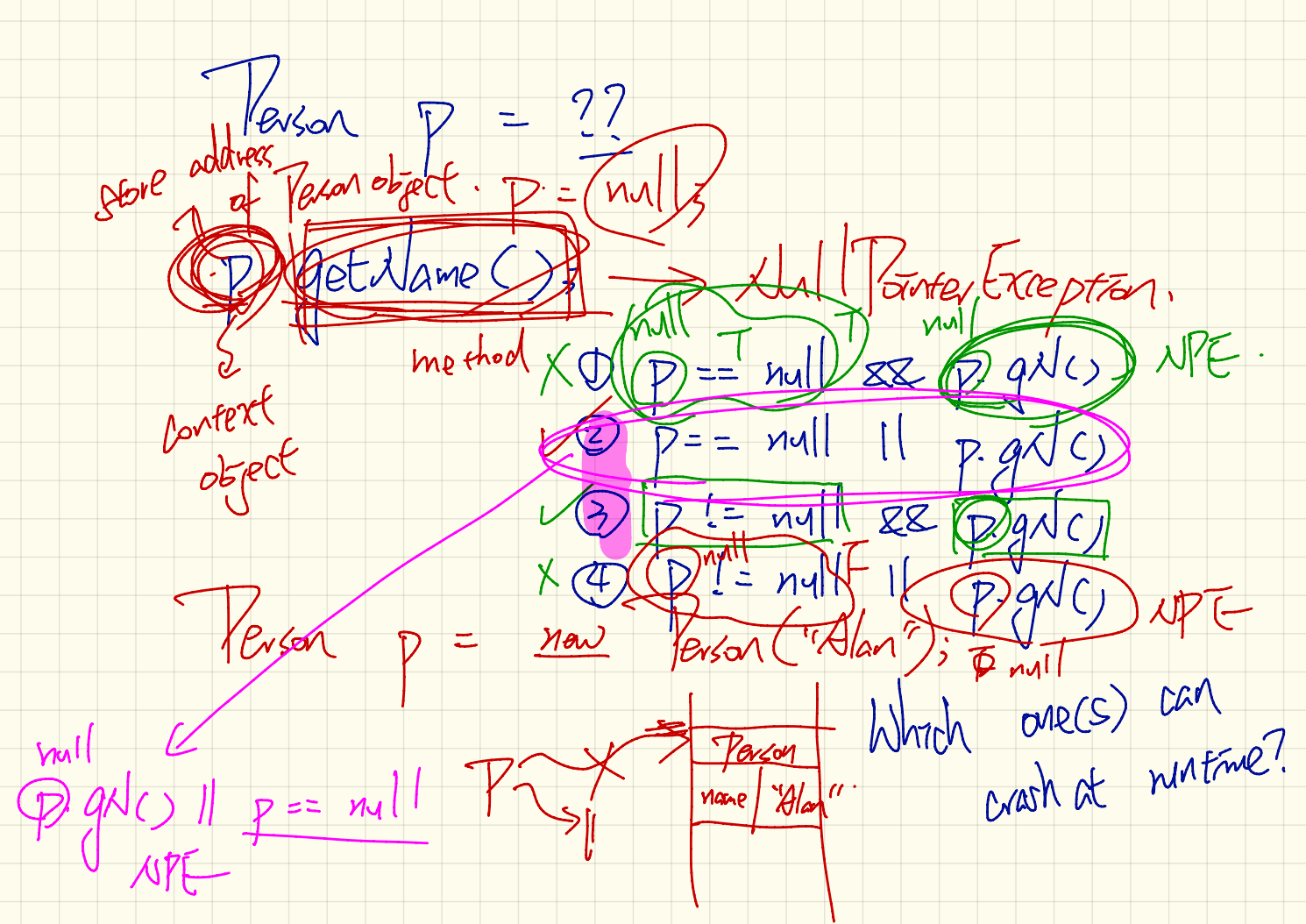
Avoid evaluating this: `y/x > 2`

`y/x > 2 || x == 0`

`y/x` → 0

b1	b2	b1 or b2
T	T	
T	F	
F	T	
F	F	


int i = ?? -1
int[] a = ?? {2, 3, 4, 5, 6}
①
0 <= i && i < a.length && a[i] > 3
②
i < a.length && a[i] > 3 && 0 <= i
guarding cond.
False if i >= a.length (too large)
False if 0 > i (too small)
crash when i is too small.
array indexing
a[i] > 3
P. m


Person p = ??
store address of Person object.
p = null;
p
getName();
context object
method
null Pointer Exception.
X ① p == null && p.gN()
NPE.
② p == null || p.gN()
③ p != null && p.gN()
X ④ p != null || p.gN()
NPE
Person p = new Person("Alan");
null
p.gN() || p == null
NPE
Person
name "Alan"
Which one(s) can crash at runtime?

Short Circuit

1. division — `x != 0 && y/x > 2`
2. array indexing — `i >= 0 && i < a.length && a[i] > 2`
3. method call — `p != null && p.g()`

```
class Person {
    String name;
    Person spouse;
}
```

reference type

① `Person jim = new Person( );`
`jim.setName("Jim");`

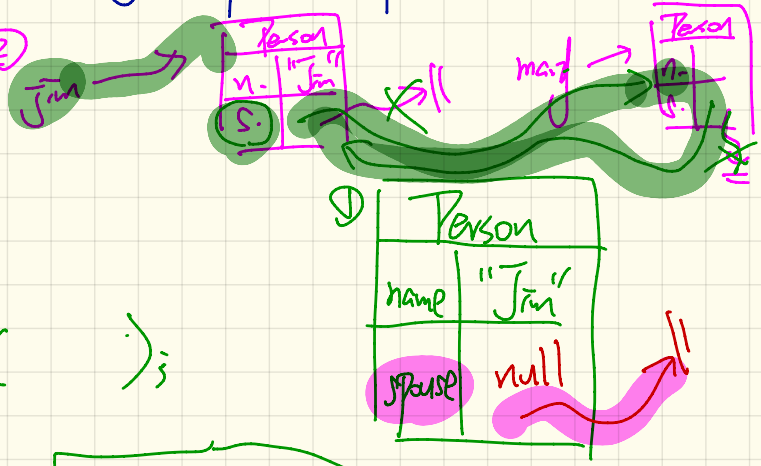
`jim.spouse.spouse.name`

context object — attribute

for name

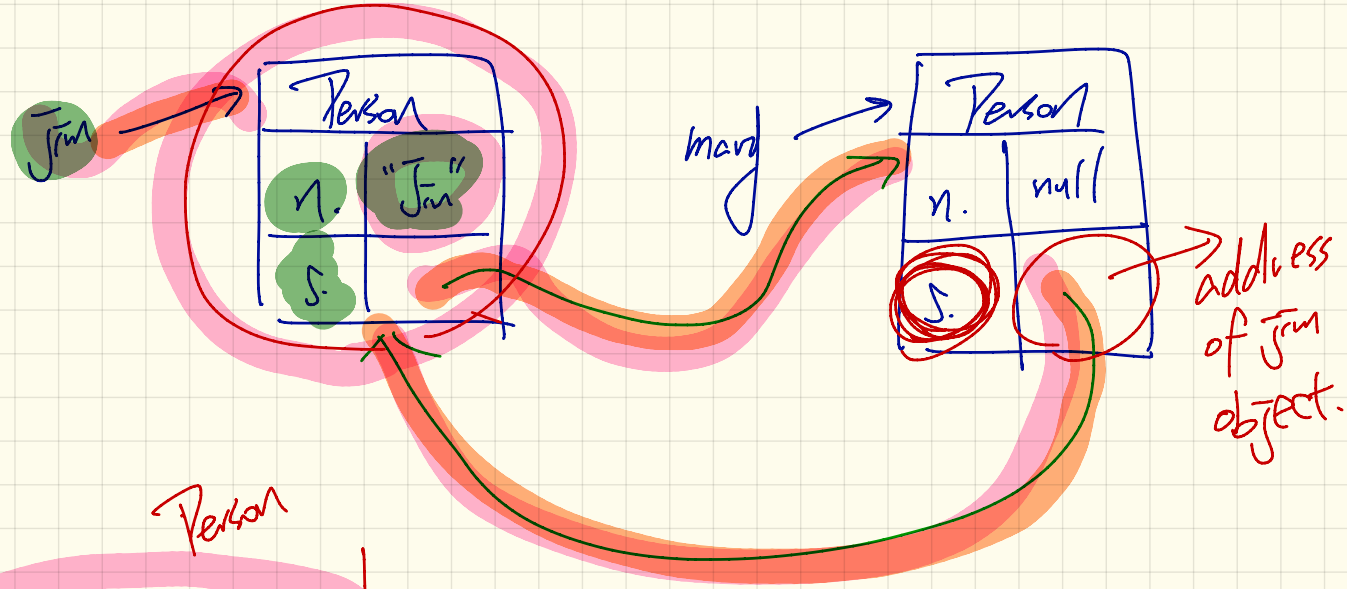
`jim.spouse` → `spouse` → `null` → NPE.

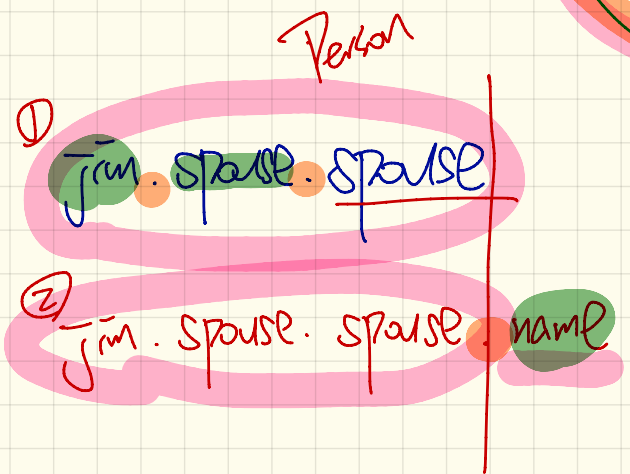
jim.spouse.spouse.name



② `Person jim = new Person( );`
`jim.setName("Jim");`
`Person mary = new Person( );`
`jim.spouse = mary;`
`mary.spouse = jim;`

jim.spouse.spouse

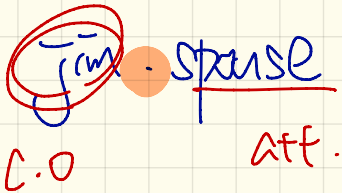


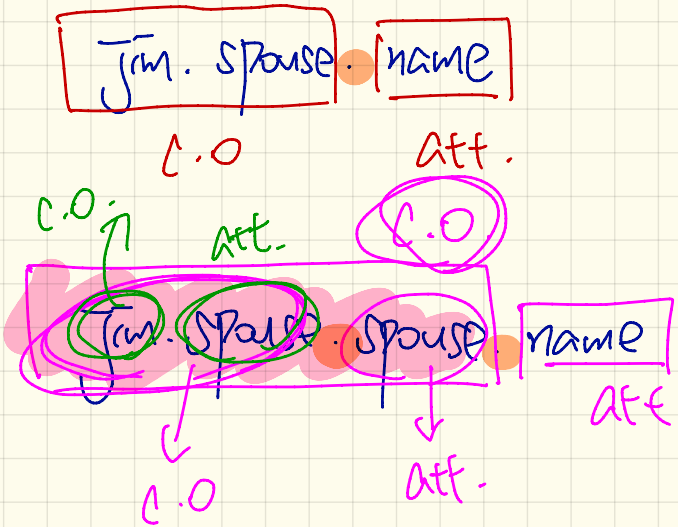


Jim . spouse . spouse . spouse . name

Jim . spouse

Jim . spouse . spouse


Jim
spouse
c.o
att.


Jim. spouse
name
c.o
att.
c.o
att.
c.o
Jim
spouse
spouse
name
att
c.o
att.

String answer = nextLine();

boolean continue = answer.equals("Y");

while (continue) {

answer = nextLine();

continue = answer.equals("Y");

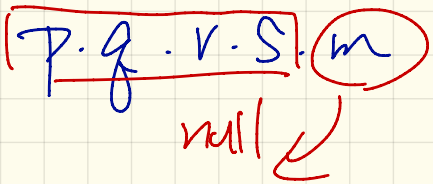
}

"Y"

"N"

"N"

NPE : when the context object is null



AIOBE : a[i]

i < 0 || i >= a.length.

Person[] persons = {p1, p2};

0 1

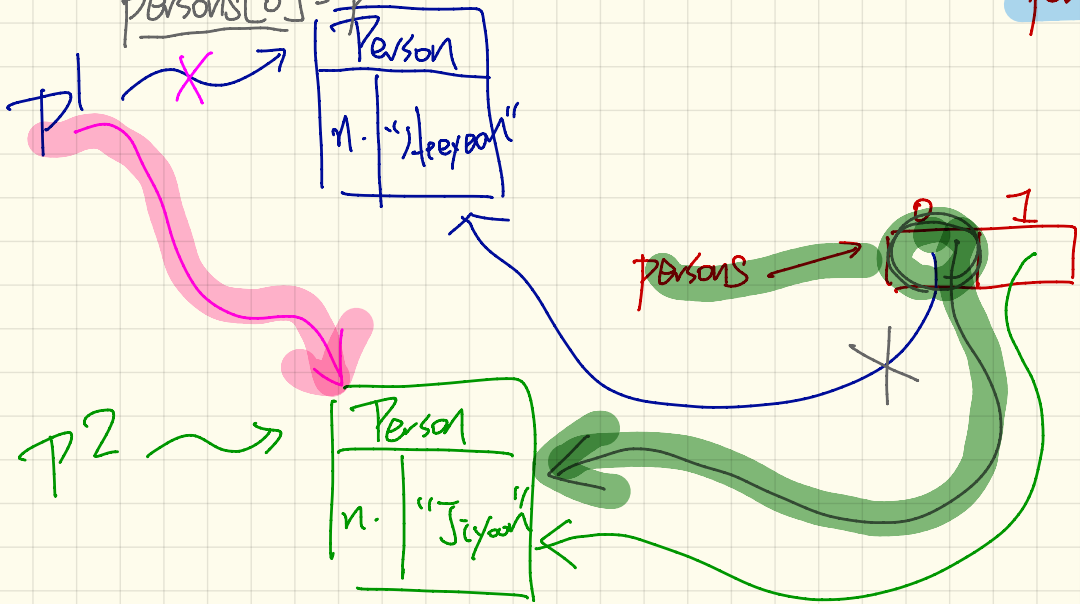
Person[] persons = new Person[2];

persons[0] = p1;

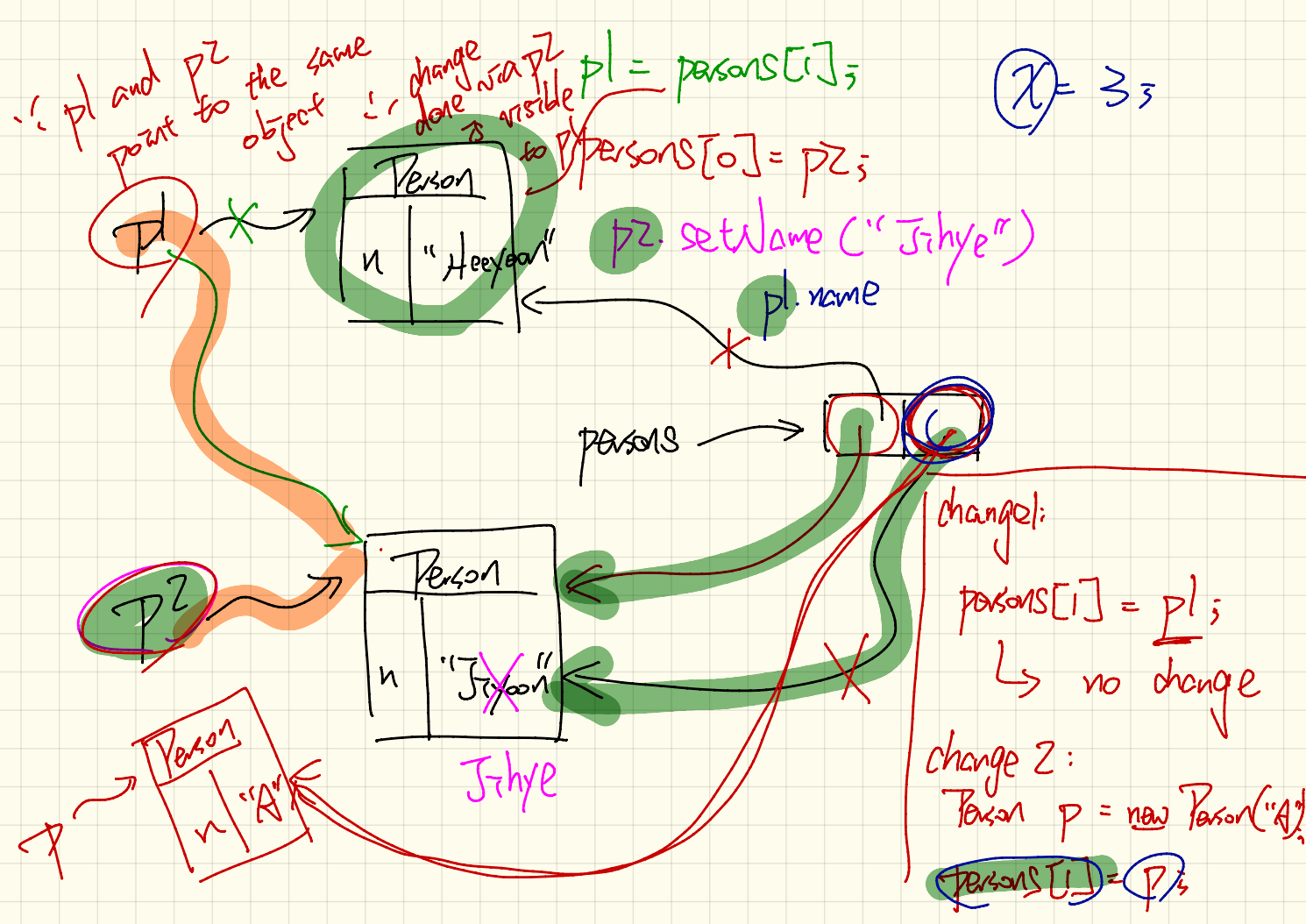
persons[1] = p2;

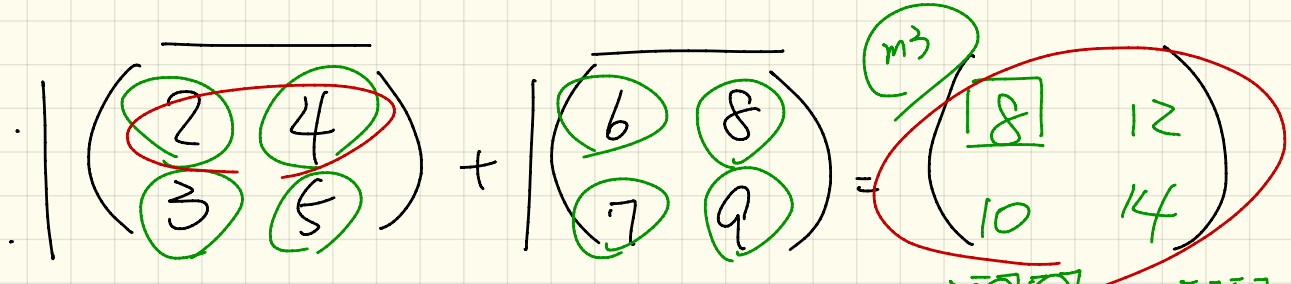
p1 = p2;

persons[0] = p2;

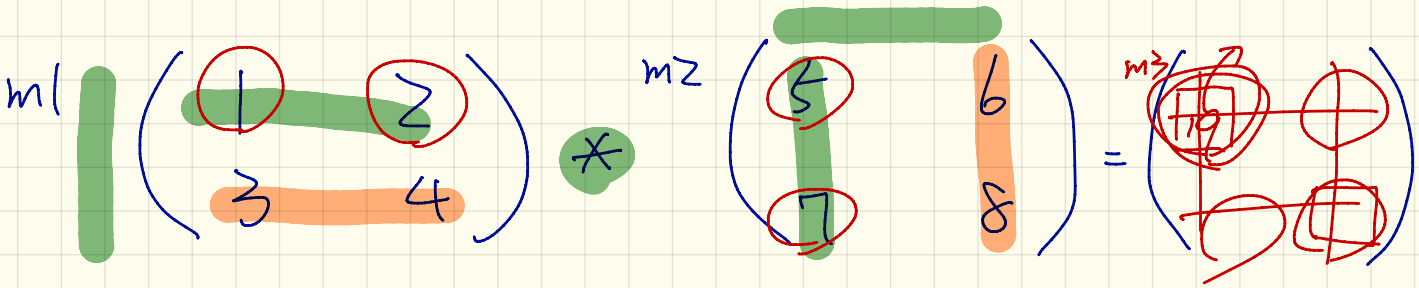


persons[0] == p1 true


∴ p1 and p2 point to the same object
∴ change done via p2 is visible to p1
p1 = persons[1];
persons[0] = p2;
x = 3;
p2.setName("Jihye")
p1.name
Person
n
"Heeyeon"
p1
p2
persons
Person
n
"Jiyoon"
Jihye
Person
n
"A"
p
change1:
persons[1] = p1;
↳ no change
change 2:
Person p = new Person("A");
persons[1] = p;




```
void add(int[][] m1, int[][] m2){
    int[][] m3 = new int[m1.length][m1[0].length];
    for(int r=0; ---){
        for(int c=0; ---){
            m3[r][c] = m1[r][c] + m2[r][c];
        }
    }
    return m3;
}
```



$1*5 + 2*7 = 19$

$3*6 + 4*8 = 50$

m3[0][0]

row 0 of m1

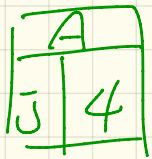
column 0 of m2

m3[1][1]

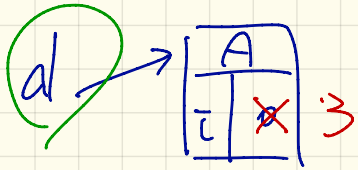
```
class A {
    int i;
    static int j;
    A() { };
}
```

non-static

```
A a1 = new A();
A a2 = new A();
```

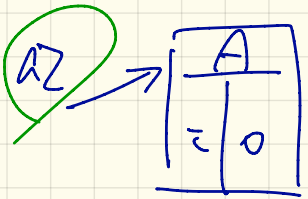



```
a1.i = 3;
```



① println(a1.i); 3

② println(a2.i); 0



A.j
a1.j = 4;

println(a1.j) 4

println(a2.j) 4