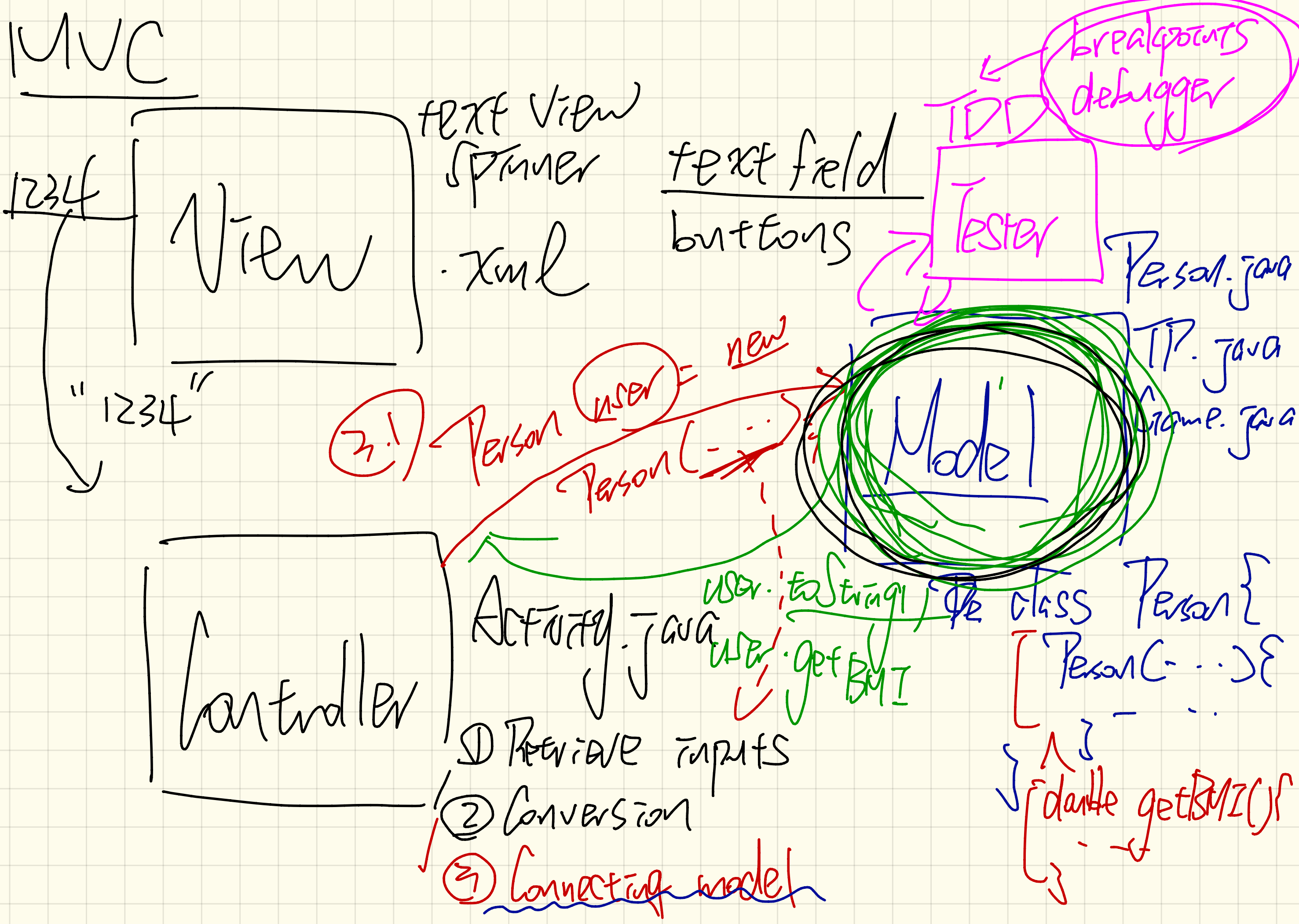
Monday Jan. 29

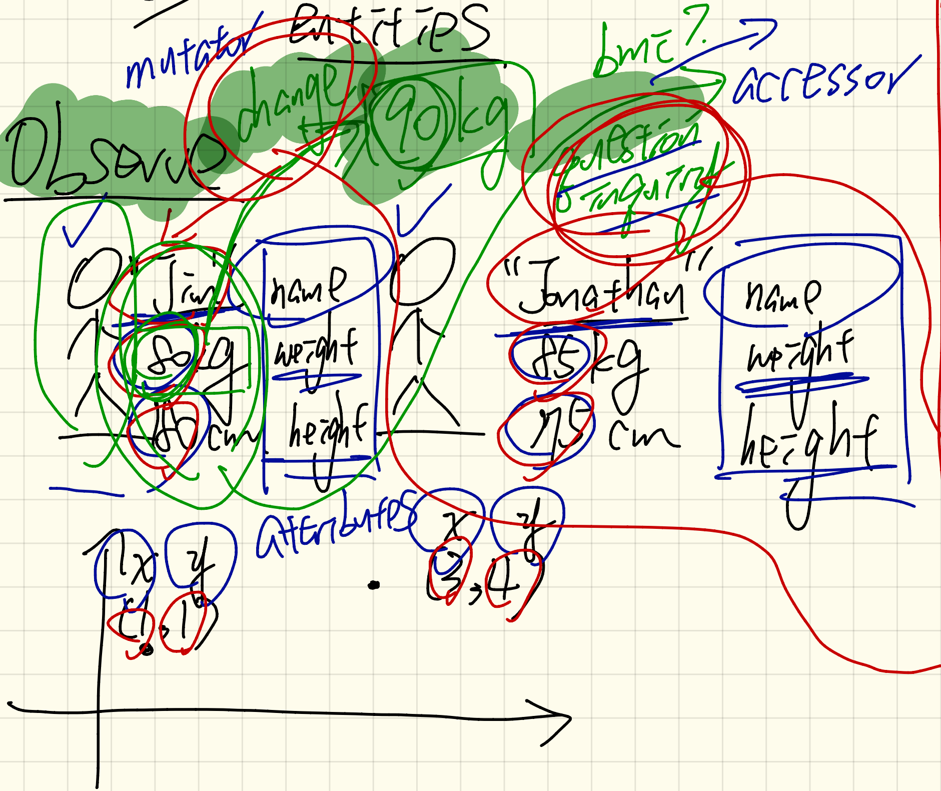
Lecture 4

MVC


1234
View
text View
spinner
.xml
text field
buttons
"1234"
IDE
breakpoints
debugger
Tester
Person.java
TP.java
Game.java
Model
3.1
Person user = new
Person(...)
Controller
Activity.java
user.toString()
user.getBMI
pc class Person{
Person(...){
...
}
double getBMI(){
...
}
}
① Retrieve inputs
② Conversion
③ Connecting model

Object Orientation (OO)

Observe entities



Model

class ___ {

attributes

constructors

accessors

mutators

}

Execute

- allocate space in mem. for new objects
- manipulate objects (call methods)

```
class Point{
    double x;
    double y;
   Point(double nx; double ny){
        x = nx;
         y = ny;
     }
}
```

Scope of variables

```
class Foo {
    int i
    void m1(int i, int j) {
        ✓ int k = i * j;
    }
    void m2( ) {
        int k = i * j;
    }
}
```

attribute
↓
class-level variable

?

input parameter i shadows attribute i.

?

scope of m1

scope of m2

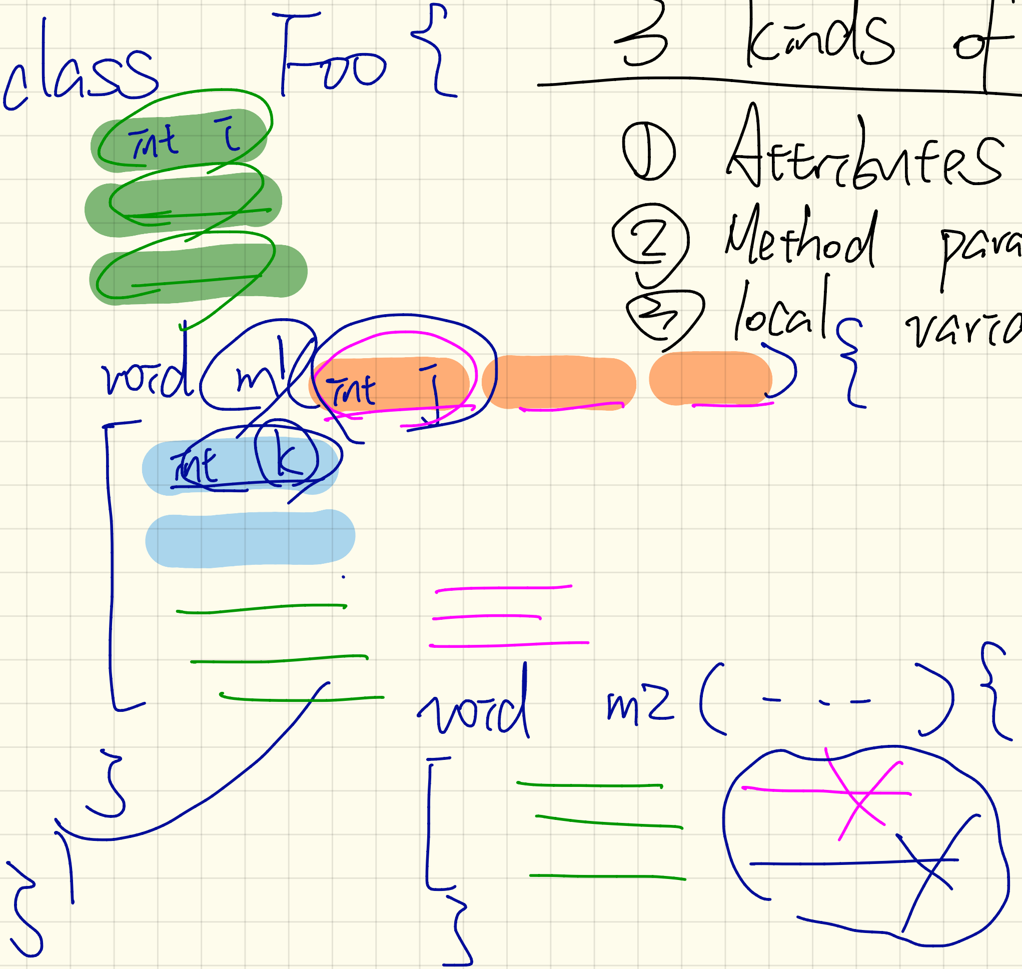
does not compile.

3 kinds of variables

① Attributes (class-level)

② Method parameters

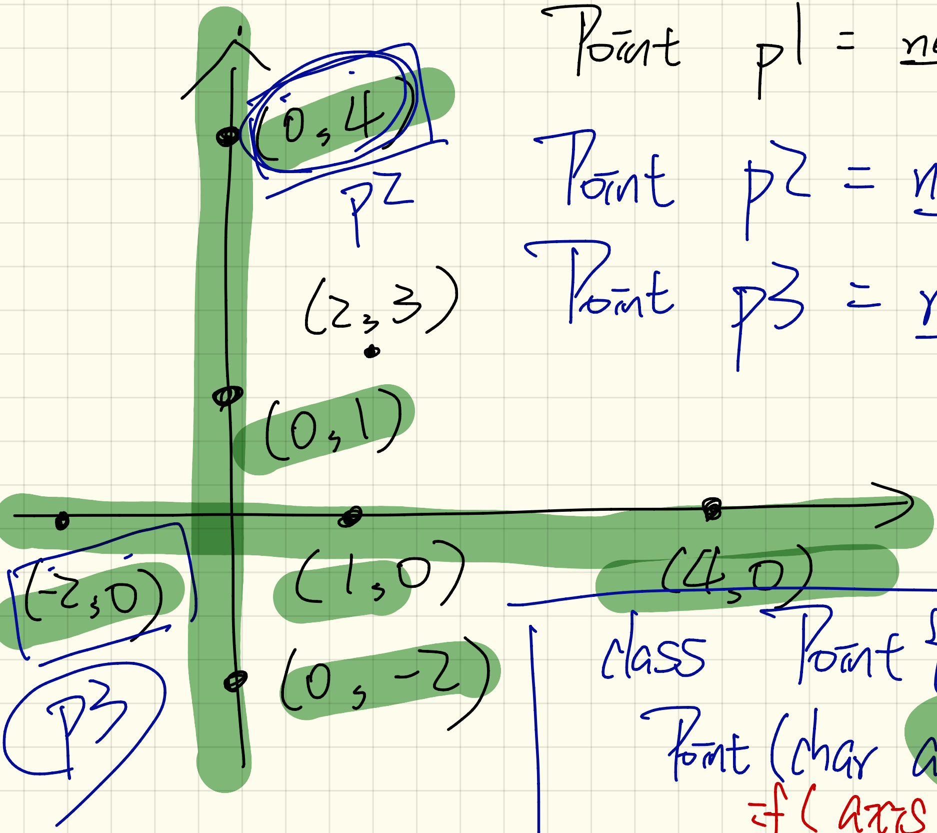
③ locals variables.



```
class Foo {
    int a1;
    void m1() {
        int i = 23;
        if ( i > 24) {
            int j = i * 2;
        }
        else {
            int k = i * 4;
            j = 46;
        }
        j = 46; X
    }
}
```

visible to the entire "m1"

visible to the entire "if" branch



```
Point p1 = new Point(2, 3);
Point p2 = new Point('y', 4);
Point p3 = new Point('x', -2);
```

y

x

```
class Point {
  Point(char axis, double val) {
    if (axis == 'x') {this.x = val;}
    else if (axis == 'y') {this.y = val;}
} }
```

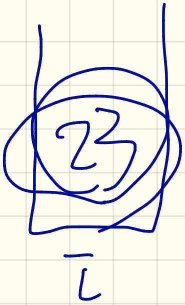
✓ double

```
int i = 23;
```

all-lower case

⇓

primitive type (value)



```
class Point {
  double x;
  double y;
  Point( --- ) { --- }
}
```

```
Point p = new Point(2, 3);
```

Capitalized

⇓

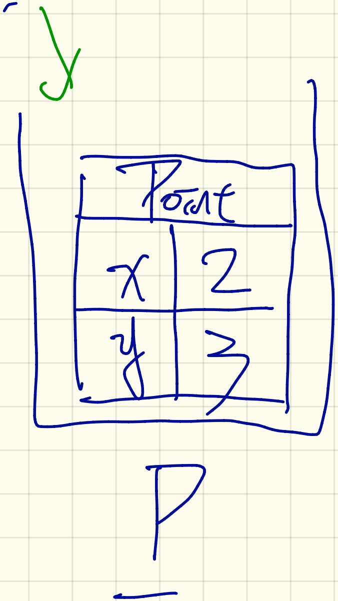
reference type

(address of composite structure)

stores address of some Point object.

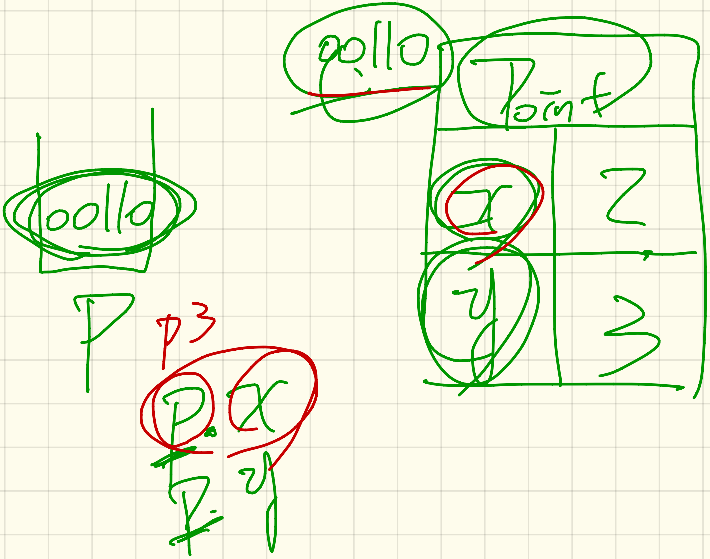
Point p = new Point(2, 3);

Wrong

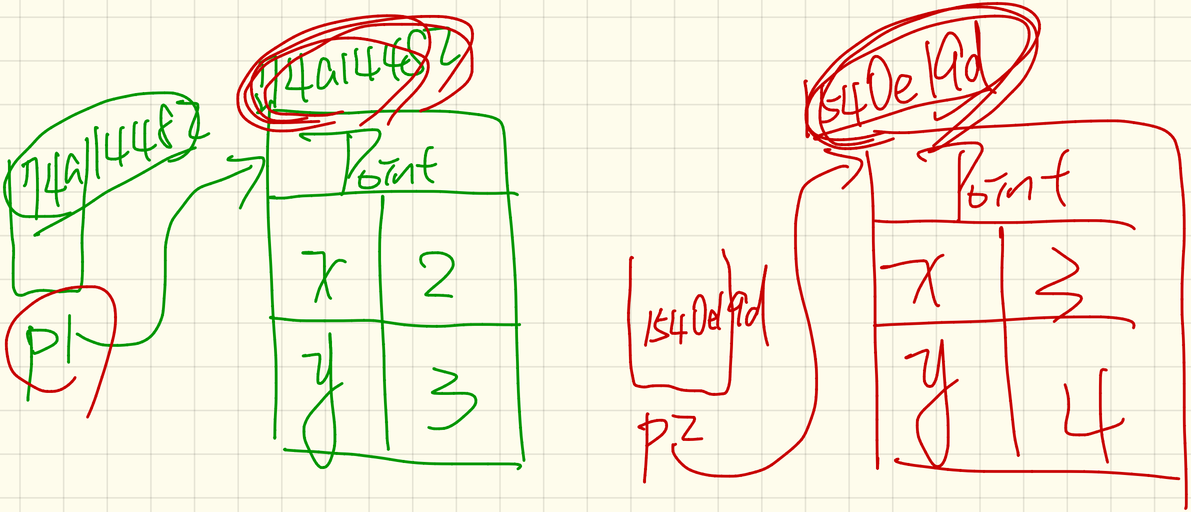


Correct

address (EECS2021)



```
Point p1 = new Point(2, 3);
Point p2 = new Point(3, 4);
```


74a14482
p1
74a14482
Point
x
2
y
3
1540e19d
p2
1540e19d
Point
x
3
y
4