

# 1. give a complete description of

650610795

650610821

## AirPurifier class

- what should be instance variables?
- what should be instance methods?
- what should be class variables?
- what should be class methods?

### instance variables

```
- private String model;  
- private String serialNo;  
- boolean power = false; → default ให้เป็น false ไว้  
- private String mode = "auto"; → default ให้เป็น auto ไว้  
- private int speedFan = 1; → default ให้เป็น 1 ไว้  
- int pm = (int) (Math.random() * 370); → assume ให้ค่า pm ที่ sensor รับเข้ามา มีค่า random [0,370)
```

### class variable

```
- static ArrayList<String> allModel = new ArrayList<>();  
- static ArrayList<Integer> numberOfEachModel = new ArrayList<>();  
- static int numberOfAPF = 0; → ให้ค่าเริ่มต้นเท่ากับ 0
```

### instance methods

```
- void turnOn() { power = true; } → method turnOn จะให้ power เป็น true เพื่อเปิด airpurifier  
- void turnOff() { power = false; } → method turnOff จะให้ power เป็น off เพื่อปิด airpurifier  
- void toggleOnOff() { power = !power; } → method toggleOnOff ให้ toggle power  
- void setMode (String _mode) {  
    if (_mode.equals("auto") || _mode.equals("sleep") || _mode.equals("custom")) {  
        this.mode = _mode;  
    } else {  
        System.out.println("This mode does not exist.");  
        return;  
    }  
    this.setSpeedFan();  
} → method setMode จะรับ _mode ที่รับ auto หรือ sleep หรือ custom ให้ mode เท่ากับ _mode input ให้อยู่  
แค่ 3 mode ที่กำหนดไว้ print ออกไปว่า This mode does not exist. แล้วจะกลับไปรับใน setSpeedFan  
ตาม _mode นั้นๆ  
- String pmLV() {  
    String Lv = "Safe";  
    if (pm <= 64) Lv = "Safe";  
    if (pm > 64 && pm <= 124) Lv = "Moderate";  
    if (pm > 125) Lv = "Unhealthy";  
    return Lv;  
} → method pmLV จะใช้บอกว่า ค่า pm ณ ตอนนั้น มี Lv เป็นอย่างไร โดย ถ้า ค่า pm <= 64 จะเป็น Level Safe  
ถ้า ค่า pm มากกว่า 64 แต่ไม่เกิน 124 จะเป็น Level Moderate แต่ถ้า ค่า pm มากกว่า 125 จะเป็น  
Level Unhealthy
```



## instance methods (no)

```

void setSpeedFan() {
    if (this.mode.equals("sleep")) speedFan = 1;
    if (this.mode.equals("custom")) {
        Scanner s = new Scanner(System.in);
        System.out.println("[custom mode] Enter the fan speed:");
        int speed = s.nextInt();
        if (speed >= 1 && speed <= 3) {
            speedFan = speed;
        } else {
            System.out.println("SpeedFan should be between 1 to 3.");
            setSpeedFan();
        }
    }
    if (this.mode.equals("auto")) {
        if (pm <= 64) speedFan = 1;
        if (pm > 64 && pm <= 124) speedFan = 2;
        if (pm > 124) speedFan = 3;
    }
}

```

→ method setSpeedFan เป็นการ setSpeedFan ในแต่ละ mode ว่าในแต่ละ mode speed fan เป็นอย่างไรบ้าง ถ้า mode เป็น sleep speedFan จะถูก set เป็น 1 ถ้า mode เป็น custom จะสามารถ setSpeedFan เองได้โดยต้องมีค่าระหว่าง 1 ถึง 3 และถ้า mode เป็น auto จะนำค่า pm มาดูว่าค่า pm มีค่าเท่าไรอยู่ในช่วงไหน เพื่อ setSpeedFan ตามช่วงที่กำหนด

```

void showDisplay()

```

→ method นี้ใช้ show ว่ามีอะไรบ้างจะแสดงอะไรบ้าง จะแสดง serial No, model, ค่า pm, pmLV, status vol power, mode และ speedFan



## class methods

```

-static int modelCount() { return allModel.size(); }
    → จำนวนของ model ทั้งหมดจะมีค่าเท่ากับขนาดของ array list allModel
-static String mostPopularModel() {
    int maxNumber = Collections.max(numberOfEachModel);
    int index = numberOfEachModel.indexOf(maxNumber);
    return allModel.get(index);
}
    → หา model ที่ popular มากที่สุด จะ return ออกมาเป็น model ที่ popular มากที่สุด
-static int numberOfThisModel(String model) {
    int index = allModel.indexOf(model);
    return numberOfEachModel.get(index);
}
    → หาจำนวนของ model หนึ่ง แล้ว return ออกมาเป็นจำนวนของ model ที่ต้องการทราบ

```

2. for each method proposed above, how do you plan to implement it?

- give an overview of the computation
  - what's needed as inputs?
  - what's to be returned?

```
- void turnOn() { power = true; } → ไม่มี input และ คืนค่า return ออก  
- void turnOff() { power = false; } → ไม่มี input และ คืนค่า return ออก  
- void toggleOnOff() { power = !power; } → ไม่มี input และ คืนค่า return ออก ใช้ toggle power เมื่อถูกป้อน  
- void setMode(String _mode) {  
    if (_mode.equals("auto") || _mode.equals("sleep") || _mode.equals("custom")) {  
        this.mode = _mode;  
    } else {  
        System.out.println("This mode does not exist.");  
        return;  
    }  
    this.setSpeedFan();  
}
```

→ input คือ String \_mode เมื่อ set mode จะต้องมี 3 mode เท่านั้นที่ set ได้ ได้แก่ auto, sleep, custom หากไม่มีก็จะ print ออกไปว่า This mode does not exist. จากนั้นจะทำการ setSpeedFan แล้วจึง return ค่าออกไป

```
- String pmLv() {  
    String Lv = "Safe";  
    if (pm <= 64) Lv = "Safe";  
    if (pm > 64 && pm <= 124) Lv = "Moderate";  
    if (pm > 125) Lv = "Unhealthy";  
    return Lv;  
}
```

→ ไม่มี input เข้ามา โดยค่า pm อยู่ในช่วงไหนก็จะ return String ออกมา เช่น pm = 30 จะ return ออกมา เป็น Safe

```
- void showDisplay() {  
    → ไม่มี input and return ออกไป ให้แสดงรายละเอียดต่างๆ จะแสดง serial No, model, ค่า pm, pmLv, status ของ power, mode และ speed Fan
```

```
- static int modelCount() { return allModel.size(); }
```

→ ไม่มี input return จำนวนทั้งหมดของ model ว่ามีกี่ชิ้น

```
- static String mostPopularModel() {  
    int maxNumber = Collections.max(numberOfEachModel);  
    int index = numberOfEachModel.indexOf(maxNumber);  
    return allModel.get(index);  
}
```

→ ไม่มี input return ออกมาเป็น model ที่ popular มากที่สุด

```
- static int numberOfThisModel(String model) {  
    int index = allModel.indexOf(model);  
    return numberOfEachModel.get(index);  
}
```

→ input เป็น model ที่ต้องการทราบ โดย return ออกมาเป็นจำนวนของ model ที่ต้องการทราบ

```
void setSpeedFan () {
```

```
    if (this.mode.equals("sleep")) speedFan = 1;
```

```
    if (this.mode.equals("custom")) {
```

```
        Scanner s = new Scanner(System.in);
```

```
        System.out.println("[custom mode] Enter the fan speed:");
```

```
        int speed = s.nextInt();
```

```
        if (speed >= 1 && speed <= 3) {
```

```
            speedFan = speed;
```

```
        } else {
```

```
            System.out.println("SpeedFan should be between 1 to 3.");
```

```
            setSpeedFan();
```

```
        }
```

```
    }
```

```
    if (this.mode.equals("auto")) {
```

```
        if (pm <= 64) speedFan = 1;
```

```
        if (pm > 64 && pm <= 124) speedFan = 2;
```

```
        if (pm > 124) speedFan = 3;
```

```
    }
```

```
}
```

→ ไม่มี input and return ให้ setSpeedFan ตาม mode ต่างๆ ถ้า mode เป็น sleep speedFan จะถูก set เป็น 1 ถ้า mode เป็น custom จะส่งมาที่ setSpeedFan เราได้ ต้องมีค่าระหว่าง 1 ถึง 3 และถ้า mode เป็น auto จะนำค่า pm มาดูว่าค่า pm มีค่าเท่าไรอยู่ในช่วงไหน เพื่อ setSpeedFan ตามช่วงที่กำหนด