

วัตถุประสงค์

A. เพื่อเข้าใจหลักการ polymorphism

กิจกรรมที่ 1

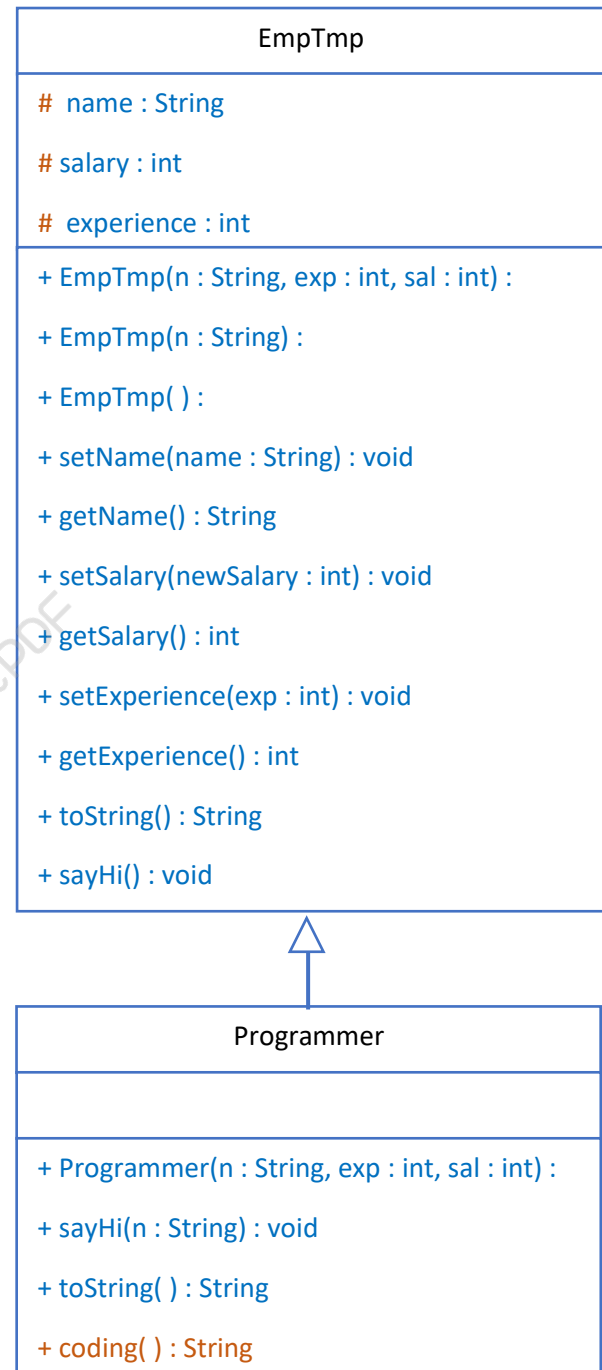
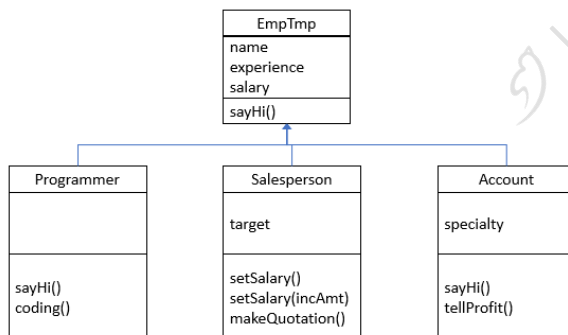
ยังอยู่ใน packA

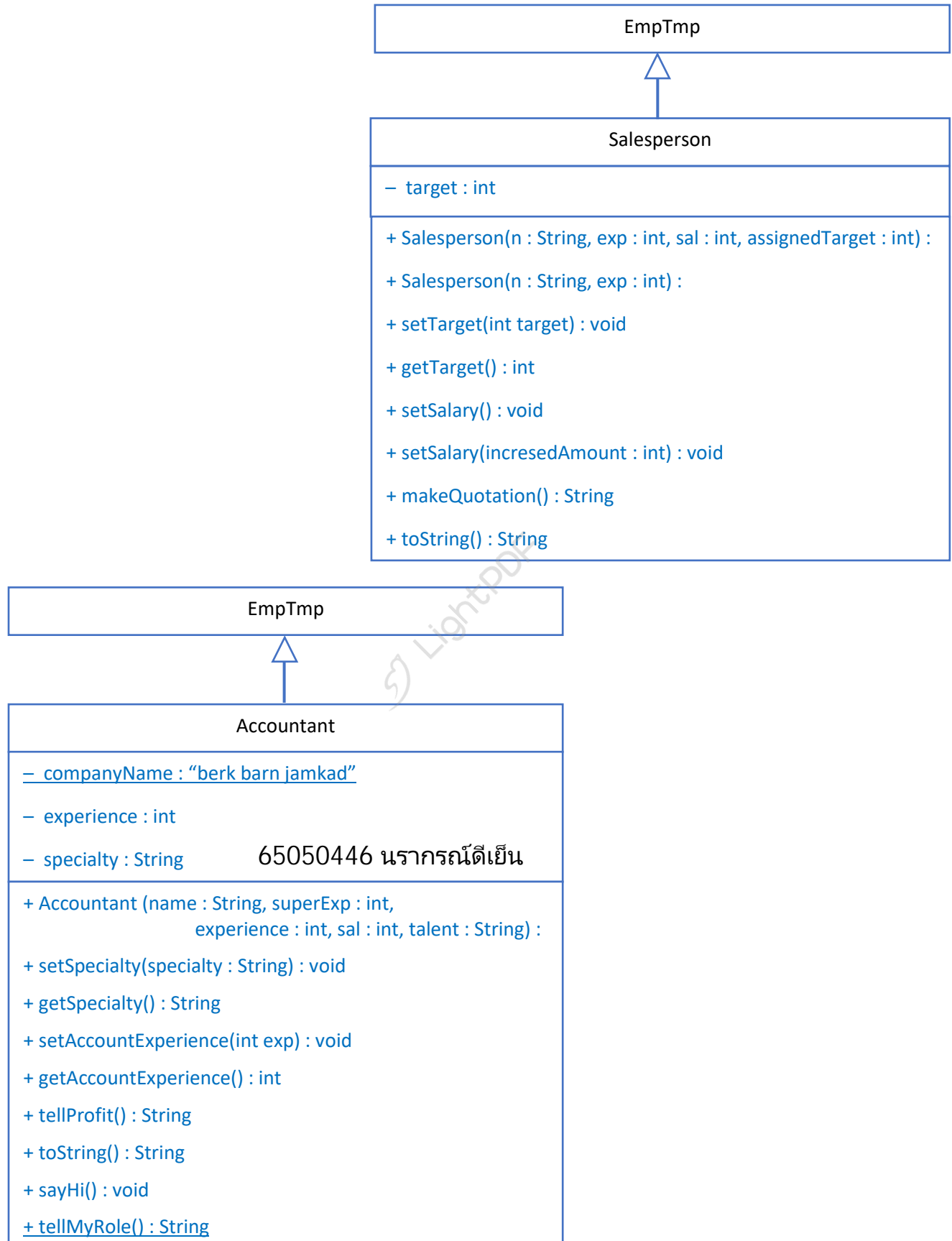
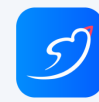
1.1 สร้าง EmpTmp.java ให้ Programmer, Salesperson, Accountant extends จาก EmpTmp นี้

Programmer override sayHi() และมี coding()

Salesperson override setSalary(int incAmt) และ overload setSalary() และมี makeQuotation()

Account override sayHi() และมี tellProfit()





## กิจกรรมที่ 2

```
9      static void q1() {
10         ArrayList<EmpTmp> aList = new ArrayList<>();
11         aList.add(new Accountant(name: "goodAtmyJob", superExp: 7, exp: 6, sal: 390, talent: "sing"));
12         aList.add(new Accountant(name: "canRap", superExp: 4, exp: 9, sal: 480, talent: "rap"));
13         aList.add(new Salesperson(name: "mr.salesperson", exp: 5, sal: 150, assignedTarget: 5000));
14         aList.add(new Salesperson(name: "mr.kayan", exp: 3, sal: 260, assignedTarget: 9000));
15         aList.add(new Programmer(n: "Keng", exp: 2, sal: 300));
16         aList.add(new EmpTmp(n: "invisible", exp: 9, sal: 120));
17
18         System.out.println(x: "Upcasting");
19         for (EmpTmp e : aList)
20             // as an EmpTmp reference, its methods are of EmpTmp
21             // but the method invoked is of the actual object type
22             e.sayHi();
23         System.out.println(x: "Downcasting");
24         for (EmpTmp e : aList) { // downcasting
25             String str;
26
27
28
29
30             Q1
31
32
33
34
35             System.out.println(str);
36         }
37     }
```

2.1 สังเกตว่าความสัมพันธ์ของ parent-child ดีกว่าให้ Salesperson และ Accountant สืบทอดจาก Programmer และสังเกตว่าระบบสามารถสร้าง object type EmpTmp ได้

2.2 ตอบ Q1 โดย หากเป็น programmer ให้เรียก coding() หากเป็น Salesperson ให้เรียก makeQuotation() และ หากเป็น Accountant ให้เรียก tellProfit() (หาก e เป็น EmpTmp ไม่ต้องทำอะไร)

2.2 เขียนผลลัพธ์จาก line 35

```
(Emp) hi from invisible
Downcasting
berk barn jamkad's profit is 1931. My salary is 390
berk barn jamkad's profit is 844. My salary is 480
Dear value customer, 6441 is my best offer.
Dear value customer, 1078 is my best offer.
Hello World
PS C:\Users\ann.dear\OneDrive\เอกสาร\OOP\Unsend\Lab3
```

สรุปหลักการ polymorphism พอสังเขป

```
System.out.println(x: "Downcasting");
for(EmpTmp e : aList){
    String str;
    if(e instanceof Programmer){
        Programmer p = (Programmer)e;
        str=p.coding();
    } else if(e instanceof Salesperson){
        Salesperson s = (Salesperson)e;
        str = s.makeQuotation();
    } else if(e instanceof Accountant){
        Accountant a = (Accountant)e;
        str = a.tellProfit();
    } else continue;

    System.out.println(str);
}
```

## Polymorphism

คือการที่ แม้มียุคคลหลายคนหรือเป็นการสร้างคลาสย่อยหลายๆคลาสที่อ้างอิงมาจากคลาสใหญ่ซึ่งจะทำให้เราสามารถ สร้างคลาสย่อยที่มีประเภทเป็นคลาสใหญ่ที่สามารถเรียกใช้เมธอดที่มีอยู่ในคลาสใหญ่ได้ และหากต้องการเรียกใช้เมธอดจากคลาสย่อยนั้นต้องระบุประเภทของคลาสย่อยให้ java รู้ก่อนจากนั้นจึงเรียกใช้

กำหนดส่ง TBA

65050446 นรากรณดีเย็น