

วัตถุประสงค์

A. เพื่อเข้าใจหลักการ polymorphism

กิจกรรมที่ 1

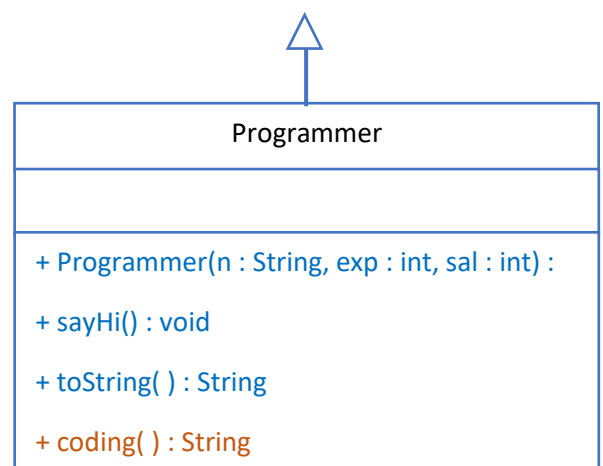
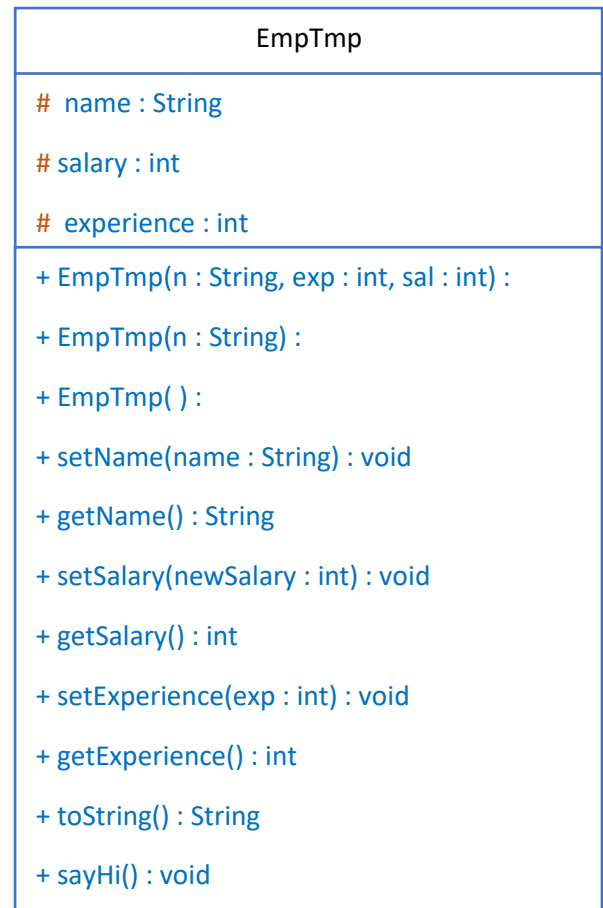
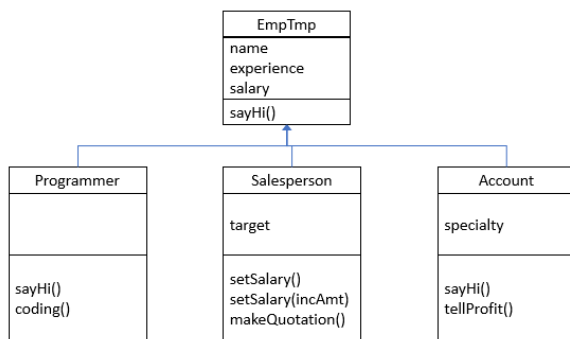
ยังอยู่ใน packA

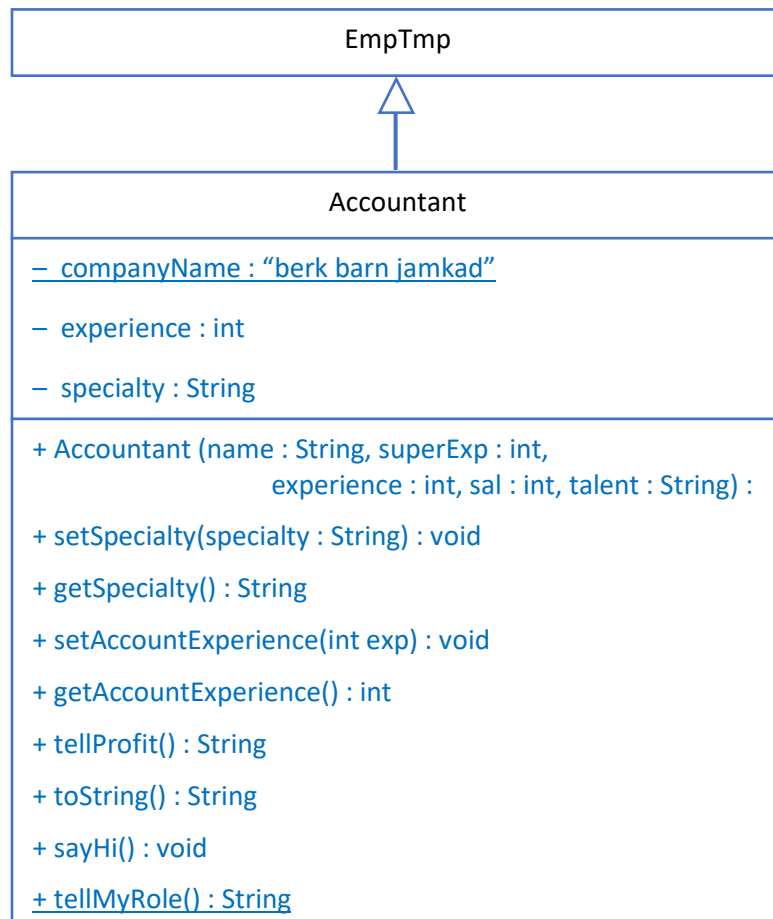
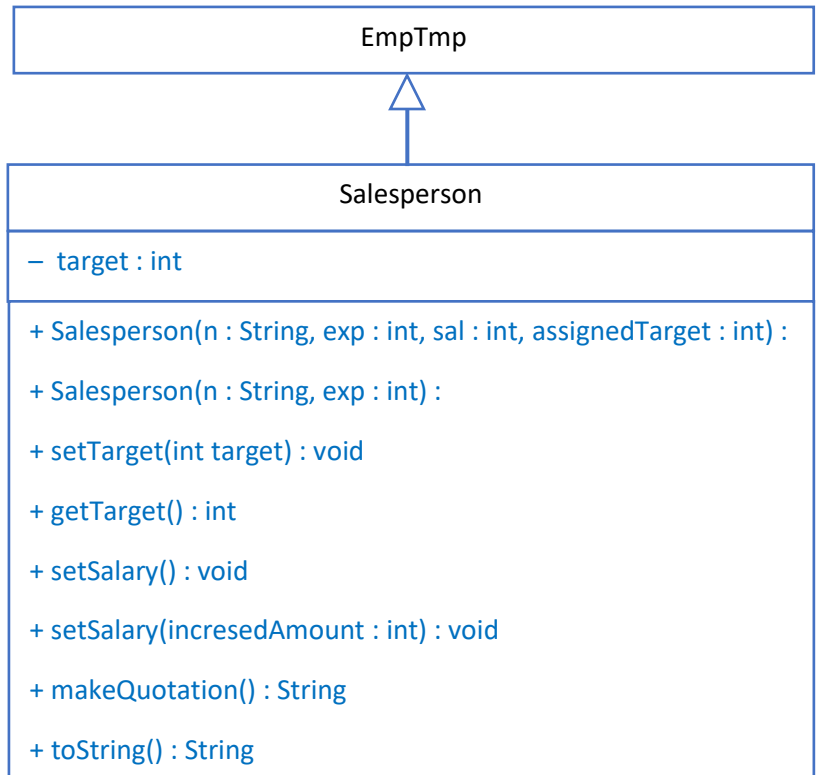
1.1 สร้าง EmpTmp.java ให้ Programmer, Salesperson, Accountant extends จาก EmpTmp นี้

Programmer override sayHi() และมี coding()

Salesperson override setSalary(int incAmt) และ overload setSalary() และมี makeQuotation() (จะเรียก sayHi() ของ EmpTmp)

Account override sayHi() และมี tellProfit()





กิจกรรมที่ 2

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)

```

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             // Upcasting = an EmpTmp reference is generic to its children,
21             //             it sees the structure of the object as defined in EmpTmp
22             //             but the method invoked is of the actual object type
23             e.sayHi();
24         System.out.println(x: "Downcasting");
25         for (EmpTmp e : alist) { // downcasting = "casting" a generic reference so the object
26                                 // becomes its actual type.
27             String str;
28
29
30
31
32
33
34
35
36
37             System.out.println(str);
38         }
39     }

```

Q1

```

if (e instanceof Accountant)
    str = ((Accountant)e).tellProfit();
else if (e instanceof Programmer)
    str = ((Programmer)e).coding();
else if (e instanceof Salesperson)
    str = ((Salesperson)e).makeQuotation();
else
    continue;

```

2.2 เขียนผลลัพธ์จาก line 35

berk ban jamkad's profit is 190. My salary is 390.
 berk ban jamkad's profit is 124. My salary is 480.
 Deer value customer, 599 is my best offer.
 Deer value customer, 952 is my best offer.
 Coding from programmer.

สรุปหลักการ polymorphism พอสังเขป

การเขียนโปรแกรมนี้ให้สอดคล้องกับ objects เช่น class ต่างกัน โดยใช้ method เดียวกัน
 โดย method ของแต่ละ class ก็จะมีการทำงานต่างกัน

กำหนดส่ง TBA