05506004

วัตถุประสงค์

A. เพื่อเข้าใจหลักการ polymorphism

กิจกรรมที่ 1

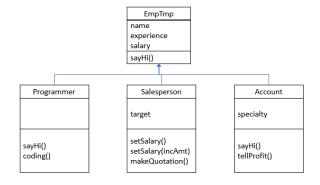
ยังอยู่ใน packA

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()



```
# name : String
# salary : int
# experience : int
+ EmpTmp(n : String, exp : int, sal : int) :
+ EmpTmp(n : String) :
+ EmpTmp() :
+ setName(name : String) : void
+ getName() : String
+ setSalary(newSalary : int) : void
+ getSalary() : int
+ setExperience(exp : int) : void
+ getExperience() : int
+ toString() : String
+ sayHi() : void
```



- + Programmer(n : String, exp : int, sal : int) :
- + sayHi(n : String) : void
- + toString(): String
- + coding(): String

Salesperson - target: int + Salesperson(n: String, exp: int, sal: int, assignedTarget: int): + Salesperson(n: String, exp: int): + salesperson(n: String, exp: int): + setTarget(int target): void + getTarget(): int + setSalary(): void + setSalary(incresedAmount: int): void + makeQuotation(): String

+ toString(): String

Accountant - companyName: "berk barn jamkad" - experience: int - specialty: String + Accountant (name: String, superExp: int, experience: int, sal: int, talent: String): + setSpecialty(specialty: String): void + getSpecialty(): String + setAccountExperience(int exp): void + getAccountExperience(): int + tellProfit(): String + toString(): String + sayHi(): void + tellMyRole(): String

กิจกรรมที่ 2

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

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

2.2 เขียนผลลัพธ์จาก line 35

```
Upcasting
berk barn jamkad's profit is 729. My salary is 390
berk barn jamkad's profit is 221. My salary is 480
hi from mr.salesperson
hi from kayan
hi from invisible
Downcasting
berk barn jamkad's profit is 60. My salary is 390
berk barn jamkad's profit is 245. My salary is 480
Dear value customer, 768 is my best offer.
Dear value customer, 815 is my best offer.
Coding
```

สรุปหลักการ polymorphism พอสังเขป

การสร้าง Objects ที่อกจาก class ที่มี super class เดียวกัน โดยใน sub class มีการกำหนดการ ทำงานในม่ให้กับ Method ใน้อเรงกับอังกฤประสาคของ class นั้น หรือการ override method หลังจากขึ้นสามารถ ใช้ super class สำหรับการประกาศ ตัวแปรของ object c class instance) ที่สร้าง object จาก sub class ได้