

วัตถุประสงค์

A. เพื่อเข้าใจหลักการ package

B. เพื่อเข้าใจหลักการ Inheritance

กิจกรรมที่ 1

1.1 สร้างแพมย่อย packA

1.2 เขียน Programmer.java ใน packA

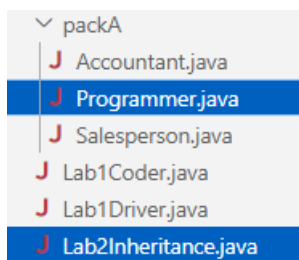
(บรรทัดแรก ประกาศว่าเป็น packA ...

สามารถ save as จาก Lec1Coder มา  
แปลงได้)

1.3 access modifier # ใช้ keyword

\_\_\_\_\_ กับ

1.4 เขียน Lab2Inheritance.java เหนือ  
packA



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

1.5 หากหน้า class Programmer { } ไม่ได้กำกับ public ไว้ main จะเห็น Programmer หรือไม่ \_\_\_\_\_

1.6 implement

sayHi() {

System.out.println("hi  
from " + name);

}

```

1  import packA.*;
2
3  public class Lab2Inheritance {
    Run | Debug
4      public static void main(String[] args) {
5          q1();
6          // q2_Salesperson();
7          // q3_Accountant();
8      }
9
10     static void q1() {
11         Programmer p1 = new Programmer(n: "ber1", exp: 2, sal: 500);
12         System.out.println(p1); // Programmer [name=ber1, salary=500, experience=2]
13     }
14 }

```

## กิจกรรมที่ 2

## 2.1 เขียน Salesperson.java

## 2.2 เรียก constructor ของ parent ด้วย super()

## 2.3 สามารถเรียก super() ที่ไม่ใช่บรรทัดแรกของ Salesperson() ได้หรือไม่ \_\_\_\_\_

## 2.4 การอ้างถึง attribute / method ของ parent class ใช้ keyword \_\_\_\_\_ กำกับ

## 2.5 implement makeQuotation() ตาม q2\_Salesperson() โดยใช้ Math.random()

## 2.6 setSalary(int newSalary) ของ Salesperson ให้ newSalary หมายถึงเงินเพิ่ม (จาก salary เดิม)

## 2.7 การ implement method ให้ต่างจาก implementation ของ parent class เช่น setSalary(int increasedAmount) เรียกว่า \_\_\_\_\_

```

15  static void q2_Salesperson() {
16      Salesperson p2 = new Salesperson(name: "mr.salesperson", exp: 5, sal: 150, assignedTarget: 5000);
17      Salesperson p3 = new Salesperson(name: "mr.kayan", exp: 4, sal: 260, assignedTarget: 9000);
18      System.out.println("example of inherited method " + p2.getName());
19      System.out.println(p2.makeQuotation());
20      System.out.print(s: "another example of inherited method ");
21      p3.sayHi();
22      System.out.println(x: "notice the result of overridden setSalary(int increasedAmount) below");
23      System.out.print(p2.getName() + "'s salary was " + p2.getSalary() + " -> ");
24      p2.setSalary(increasedAmount: 100);
25      System.out.println(p2);
26      p2.setSalary();
27      System.out.println(p2);
28      System.out.println(p3);
29      // example of inherited method mr.salesperson
30      // Dear value customer, 898 is my best offer.
31      // another example of inherited method hi from mr.kayan
32      // notice the result of overridden setSalary(int increasedAmount) below
33      // mr.salesperson's salary was 150 -> Salesperson [target=5000 Programmer [name=mr.salesperson, salary=250, experience=5] ]
34      // Salesperson [target=5000 Programmer [name=mr.salesperson, salary=275, experience=5] ]
35      // Salesperson [target=9000 Programmer [name=mr.kayan, salary=260, experience=4] ]
36      // ]
37  }

```

## 2.8 setSalary() หมายถึง salary ใหม่เป็น 110% ของเงินเดือนเดิม

## 2.9 setSalary() เป็น overload หรือ override \_\_\_\_\_

## 2.10 เขียน q2\_Salesperson()

## 2.11 attribute salary นั้นเป็น protected เราสามารถอ้างถึง salary ในคลาส Salesperson ได้หรือไม่ \_\_\_\_\_



## กิจกรรมที่ 3

## 3.1 เขียน Accountant.java

3.2 ใน Account.java มี static attribute  
ชื่อ \_\_\_\_\_

3.3 กำหนดค่า companyName เป็น  
"berk barn jamkad"

3.4 ใน Account.java มี static method  
ชื่อ \_\_\_\_\_

3.5 implement tellProfit() ตาม  
q3\_Accountant() โดยใช้  
Math.random()

3.6 Override sayHi() ตาม  
q3\_Accountant()

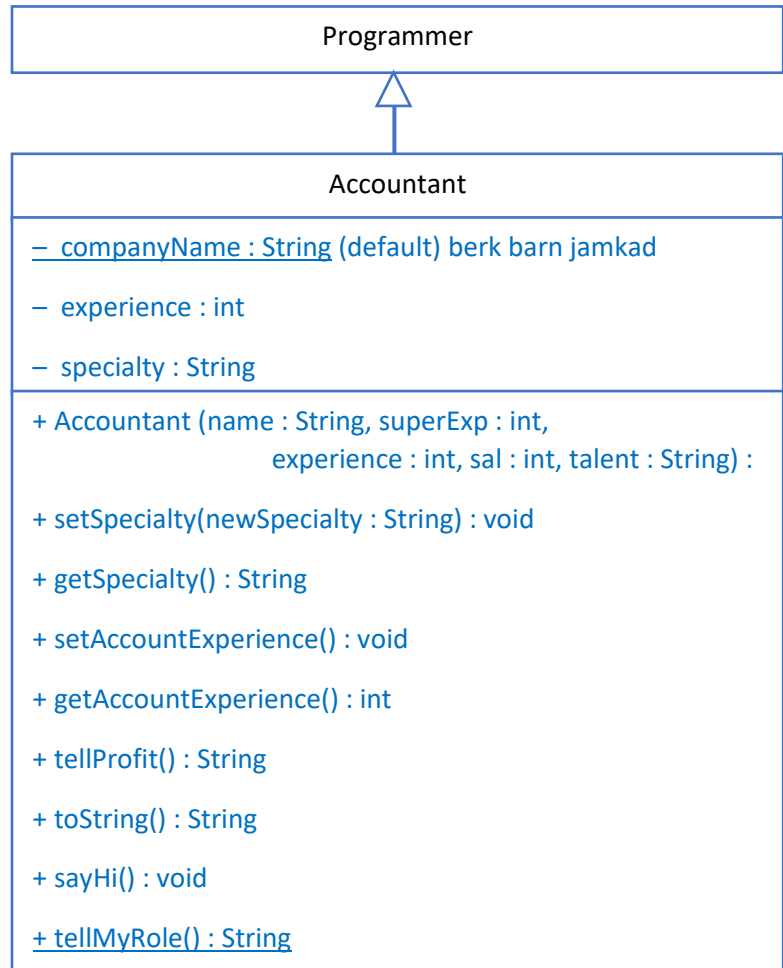
3.7 implement static String  
tellMyRole() {  
    System.out.println(  
        "I am an accountant at "  
        + companyName);  
}

3.8 เนื่องจาก Accountant มี experience  
ของตัวเอง setExperience() ผูกกับ  
experience ของ Programmer หรือ  
Accountant

3.9 วิธีอ้างถึง experience ที่ได้รับสืบทอดมาคือ \_\_\_\_\_

สรุปหลักการ inheritance พอสังเขป

กำหนดส่ง TBA



```

static void q3_Accountant() {
    Accountant p4 = new Accountant(name: "goodAtmyJob", superExp: 7, exp: 6, sal: 390, talent: "sing");
    Accountant p5 = new Accountant(name: "canRap", superExp: 4, exp: 9, sal: 480, talent: "rap");
    System.out.println(p4.tellProfit());
    p5.sayHi();
    System.out.println(Accountant.tellMyRole());
    System.out.println(p4);
    System.out.println(p5);
    // berk barn jamkad's profit is 850. My salary is 390
    // berk barn jamkad's profit is 890. My salary is 480
    // I am an accountant at berk barn jamkad
    // goodAtmyJob 7 6 can sing
    // canRap 4 9 can rap
}
  
```