

วัตถุประสงค์

A. เพื่อเข้าใจหลักการ package

B. เพื่อเข้าใจหลักการ Inheritance

กิจกรรมที่ 1

1.1 สร้างแพ็คเกจ packA

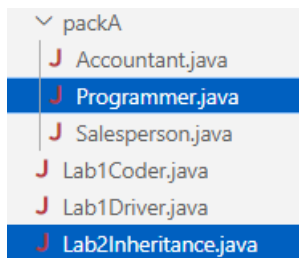
1.2 เขียน Programmer.java ใน packA

(สามารถ save as จาก Lec1Coder มาแปลงได้) บรรทัดแรก ประกาศว่าเป็น packA ด้วย `package packA;`

1.3 access modifier # ใช้ keyword

protected กำกับ

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 ไม่

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 ได้2.4 การอ้างถึง attribute / method ของ parent class ใช้ keyword super กำกับ

2.5 implement makeQuotation() ตาม q2_Salesperson() โดยใช้ Math.random()

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

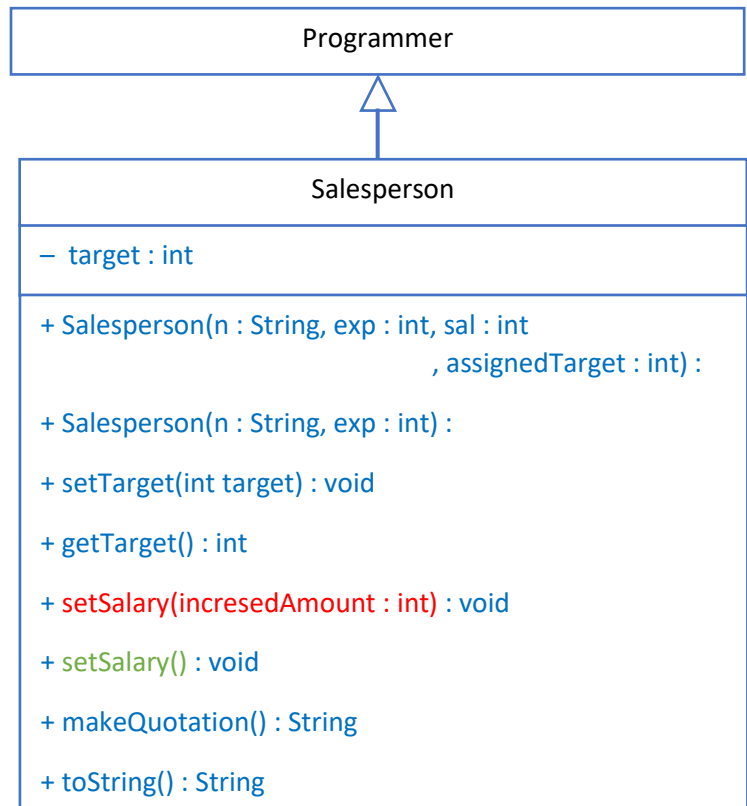
2.7 การ implement method ให้ต่างจาก implementation ของ parent class เช่น setSalary(int increasedAmount) เรียกว่า override

2.8 setSalary() หมายถึง salary ใหม่เป็น 110% ของเงินเดือนเดิม

2.9 setSalary() เป็น overload หรือ override

overload

2.10 เขียน q2_Salesperson()

2.11 attribute salary นั้นเป็น protected เราสามารถอ้างถึง salary ในคลาส Salesperson ได้หรือไม่ ได้

```

static void q2_Salesperson() {
    Salesperson p2 = new Salesperson(name: "mr.salesperson", exp: 5, sal: 150, assignedTarget: 5000);
    Salesperson p3 = new Salesperson(name: "mr.kayan", exp: 4, sal: 260, assignedTarget: 9000);
    System.out.println("example of inherited method " + p2.getName());
    System.out.println(p2.makeQuotation());
    System.out.print(s: "another example of inherited method ");
    p3.sayHi();
    System.out.println(x: "notice the result of overridden setSalary(int increasedAmount) below");
    System.out.print(p2.getName() + "'s salary was " + p2.getSalary() + " -> ");
    p2.setSalary(increasedAmount: 100);
    System.out.println(p2);
    p2.setSalary();
    System.out.println(p2);
    System.out.println(p3);
    // example of inherited method mr.salesperson
    // Dear value customer, 898 is my best offer.
    // another example of inherited method hi from mr.kayan
    // notice the result of overridden setSalary(int increasedAmount) below
    // mr.salesperson's salary was 150 -> Salesperson [target=5000 Programmer [name=mr.salesperson, salary=250, experience=5] ]
    // Salesperson [target=5000 Programmer [name=mr.salesperson, salary=275, experience=5] ]
    // Salesperson [target=9000 Programmer [name=mr.kayan, salary=260, experience=4] ]
}
  
```

กิจกรรมที่ 3

3.1 เขียน Accountant.java

3.2 ใน Account.java มี static attribute

ชื่อ companyName3.3 กำหนดค่า companyName เป็น
"berk barn jamkad"

3.4 ใน Account.java มี static method

ชื่อ tellMyRole()

3.5 implement tellProfit() ตาม

q3_Accountant() โดยใช้
Math.random()

3.6 Override sayHi() ตาม

q3_Accountant()

3.7 implement static String

tellMyRole() {
 return "I am an accountant at "
 + companyName
}

3.8 เนื่องจาก Accountant มี experience

ของตัวเอง setExperience() ผูกกับ
experience ของ Programmer หรือAccountant Programmer3.9 วิธีอ้างถึง experience ที่ได้รับสืบทอดมาคือ super.getExperience()

สรุปหลักการ inheritance พอสังเขป

มรดกต่อ attributes และ methods จาก class หนึ่งไปยังอีก class หนึ่ง

กำหนดส่ง TBA

