

ชื่อ-นามสกุล นภย ภานุการ ภานุเดช รหัสประจำตัวนักศึกษา 6604062630021 ปีการศึกษา 2567

แบบฝึกหัดปฏิบัติการภาคที่ 7: Polymorphism

คำสำคัญ

1. ให้ศึกษาหลักการ Polymorphism ต่อไปนี้

1. ความสัมพันธ์แบบการสืบทอดคุณสมบัติ อนุญาตให้คลาสสามารถสืบทอดคุณสมบัติจากคลาสแม่ (Superclass) ได้
2. คลาสลูก (Subclass) สามารถเพิ่มคุณสมบัติใหม่ ๆ ได้ โดย Subclass จะเป็นคลาสที่เฉพาะเจาะจงไปจากคลาสแม่
3. ทุก ๆ วัตถุที่สร้างจาก Subclass จะถือว่าเป็น instance ของ superclass ด้วย เช่น
 - a. ทุก ๆ วัตถุที่สร้างจากคลาส circle ถือเป็นวัตถุของคลาส Object
 - b. แต่ทุก ๆ วัตถุของคลาส Object ไม่ได้เป็นวัตถุชนิด circle
4. กรณีที่ในเมธอดมีการประกาศรับพารามิเตอร์เป็นแบบ instance ของ superclass เราสามารถส่ง instance ของ subclass ไปเป็นพารามิเตอร์ของ superclass ได้
5. ตัวแปรอ้างอิงของซูเปอร์คลาสสามารถใช้อินสแตนท์ของชั้บคลาส
6. แต่ตัวแปรอ้างอิงของชั้บคลาสไม่สามารถใช้อินสแตนท์ของซูเปอร์คลาสหรืออินสแตนท์ของคลาสที่มีซูเปอร์คลาสเดียวกันได้ ถ้าจะลองนำไปชี้คลาสอื่น ๆ ที่ไม่มีความเกี่ยวข้องกันเลยยังไม่ได้
7. การที่ตัวแปรของซูเปอร์คลาสสามารถใช้อินสแตนท์ของชั้บคลาสได้ เป็นการเพิ่มความยืดหยุ่นในการเขียนโปรแกรม ลักษณะนี้เรียกว่า โพลิมอร์ฟิซึม

Dynamic Binding

- ▶ กำหนดให้วัตถุ o เป็นอินสแตนซ์ของคลาส $C_1 C_2 \dots C_{n-1} C_n$ เมื่อ C_1 เป็นคลาลูกของ C_2 และ C_2 เป็นคลาลูกของ $C_3 \dots$ และ C_{n-1} เป็นคลาลูกของ C_n ดังปรากฏในรูปกล่าวด้านล่าง
 - ▶ C_n จะเป็นคลาสที่มีความเป็นทั่วไปมากที่สุด (most general class)
 - ▶ C_1 เป็นคลาสที่มีความเฉพาะเจาะจงมากที่สุด (most specific class) โดยคลาส C_n ในภาษาจาวาคือคลาส Object
- ▶ ถ้า o เป็นอินสแตนซ์ของคลาส C_1 โดยที่ o เรียกเมธอด p แล้ว JVM จะทำการค้นหาเมธอดที่ตรงกับที่ต้องการมากที่สุดโดยจะค้นหาเมธอด p ในคลาส $C_1 C_2 \dots C_{n-1} C_n$ ตามลำดับจนกว่าจะพบ

Static Binding

คือการผูกตัวแปรอ้างอิงกับชนิดข้อมูลช่วง Compile time

ชื่อ-นามสกุล นิพัทธ์ กะจันทร์ หนูนิเดิร์ รหัสประจำตัวนักศึกษา 660406263004 ปีการศึกษา 2567

1. ให้ศึกษาและทดลองพิมพ์ Code ตัวอย่างการสืบทอดคลาสจากตัวอย่างต่อไปนี้

```
class A{
    void f(){
        System.out.println("A");
    }
}
class B extends A{
    void f(){
        System.out.println("B");
    }
}
class Test38{
    static void t(A a){
        a.f();
    }
    public static void main(String[] args){
        t(new A());
        t(new B());
    }
}
```

ให้อธิบายผลลัพธ์ของโปรแกรมพร้อมอธิบายเหตุใดจึงได้คำตอบดังกล่าว

A

B

เนื้อหาเกี่ยวกับ method Java
จะเรียกว่า method มาก class ที่รับ
อนามัย B จะเป็น A หรือ B อยู่
แล้วจะสามารถเข้าถึง

```
class A{
    int x=1;
    void inc(){x++;}
    int get(){return x;}
}
class B extends A{
    int x=2;
    void inc(){x++;}
    int get(){return x;}
}
class Test{
    public static void main(String[] args){
        B b =new B();
        A a= b;
        a.inc();
        b.inc();
        System.out.println(a.x+"," +b.x);
        System.out.println(a.get()+"," +b.get());
    }
}
```

ให้อธิบายถึงส่วนที่มีการทำงานเป็น Static binding และ Dynamic binding

1, 4

4, 4

ในการเรียก a.inc() จะออก
a ถูก reference ไปที่ b

แต่ใน x นี่ b จะเป็น (Dynamic Binding)

โดยส่วนทั่วไป

จะอธิบายว่าถ้าเราเรียก attribute
ของลูกคุณจะต้องเรียก attribute
ของลูก ซึ่งเป็นส่วนตัว x ของ superclass จะ static binding
มีลักษณะเช่นนี้ จะเรียก method แบบ dynamic

```
public class Test{
    public static void main(String[] args){
        Object fruit = new Fruit();
        Object apple= new (Apple)fruit;
    }
}
class Apple extends Fruit{ }
class Fruit{ }
```

จงอธิบายส่วนที่ผิดของโปรแกรม

binding

ก็จะ new แล้ว type-casting

ชื่อ-นามสกุล หมาย ภานุกร หะภีกุล รหัสประจำตัวนักศึกษา 6604062630021 ปีการศึกษา 2567

2. จงอธิบายผลลัพธ์ของ Output ต่อไปนี้

```
public class Test {
    public static void main(String[] args) {
        Object a1 = new A();
        Object a2 = new Object();
        System.out.println(a1);
        System.out.println(a2);
    }
}

class A {
    int x;
    public String toString() {
        return "A's x is " + x;
    }
}
```

*A's x is 0
java.lang...*

*a, คือปัจจุบัน instance
ในคลาส A หรือ Java ที่มี
method toString ที่ถูก
override*

```
public class Test {
    public static void main(String[] args) {
        Object a1 = new A();
        Object a2 = new A();
        System.out.println(a1.equals(a2));
    }
}

class A {
    int x;
    public boolean equals(A a) {
        return this.x == a.x;
    }
}
```

false

*เมื่อเรา default method
ของ equals รูปแบบ object
และเมื่อเรา a1, a2 คือปัจจุบัน
เป็น object ที่มีค่าต่างๆ
Method Java จะใช้ function
ของ object นั้น*

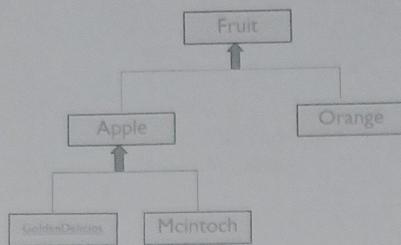
```
public class Test {
    public static void main(String[] args) {
        A a1 = new A();
        A a2 = new A();
        System.out.println(a1.equals(a2));
    }
}

class A {
    int x;
    public boolean equals(A a) {
        return this.x == a.x;
    }
}
```

true

*เมื่อเรา กำหนด ค่าปัจจุบัน
ให้ A หรือปัจจุบัน
method Java จะใช้
method ของ overload*

3. กำหนดให้ Fruit, Apple, Orange, Golden Delicious Apple, และ Macintosh Apple มีโครงสร้างการสืบทอดและการประกاثตัวแปรดังนี้ ให้เขียนโปรแกรมเพื่อจำลองการทำงานของโครงสร้างของคลาสดังกล่าวพร้อมตอบคำถามต่อไปนี้



กำหนดการทำงานภายใน main() มีการสร้าง Object ดังนี้
 Fruit fruit = new GoldenDelicious();
 Orange orange = new Orange(); ;

1. Is fruit instanceof Orange?
2. Is fruit instanceof Apple?
3. Is fruit instanceof GoldenDelicious?
4. Is fruit instanceof Macintosh?
5. Is orange instanceof Orange?
6. Is orange instanceof Fruit?
7. Is orange instanceof Apple?
8. Suppose the method makeApple() is defined in the Apple class. Can fruit invoke this method?
Can orange invoke this method?
9. Suppose the method makeOrangeJuice() is defined in the Orange class. Can orange invoke this method? Can fruit invoke this method?

4. พิจารณาส่วนของโปรแกรมต่อไปนี้

```

1 public class Test{
2     public static void main(String[] args){
3         Animal x= new Tiger();
4         System.out.println("1. x.news is "+ x.news);
5         System.out.println("2. ((Tiger)x).news is "+ ((Tiger)x).news);
6         System.out.println("3. x.smile() is "+ x.smile());
7         System.out.println("4. ((Tiger)x).smile() is "+ ((Tiger)x).smile());
8         System.out.println("5. x.getNews() is "+ x.getNews());
9         System.out.println("6. x.getMessage() is "+ x.getMessage() );
10
11     }
12
13 }
14 class Animal{
15     public String news= "Animal's news";
16     public String message="Animal's message";
17     public static String smile(){
18         return "smile from Animal";
19     }
20     public String getNews(){
21         return news;
22     }
23     public String getMessage(){
24         return message;
25     }
  
```

ชื่อ-นามสกุล..... นฤมล. กาญจน์ หลักสูตร รหัสประจำตัวนักศึกษา..... 660406263004 ปีการศึกษา 2567

```

26 }
27 class Tiger extends Animal{
28     public String news= "Tiger's news";
29     public String message="Tiger's message";
30     public static String smile(){
31         return "smile from Tiger";
32     }
33     //@override
34     public String getNews(){
35         return news;
36     }
37 }

```

ผลลัพธ์ที่จะเกิดขึ้นจากการใช้โปรแกรมต่อไปนี้คืออะไร

Animal's news	Tiger's news
Tiger's news smile from Animal	Animal's message
smile from Tiger	

อธิบายเหตุผลประกอบคำตอบ

- | | |
|---|---|
| 1) เช่น static binding เอกจาก superclasses | 5) เช่น override method บน Tiger |
| 2) casting, หักกับ เวลาซื้อ เฟรช果 ลดราคากันบ้าง | 6) Tiger Override method ด้วยร่องจาก superclass |
| 3) เช่น static Method เอกจาก superclasses | |
| 4) casting, หักกับ static method ทางเอกสารกันบ้าง | |

5. กำหนดความสัมพันธ์ของคลาส Employee และ Manager ในรูปของไดอะแกรม ดังนี้

