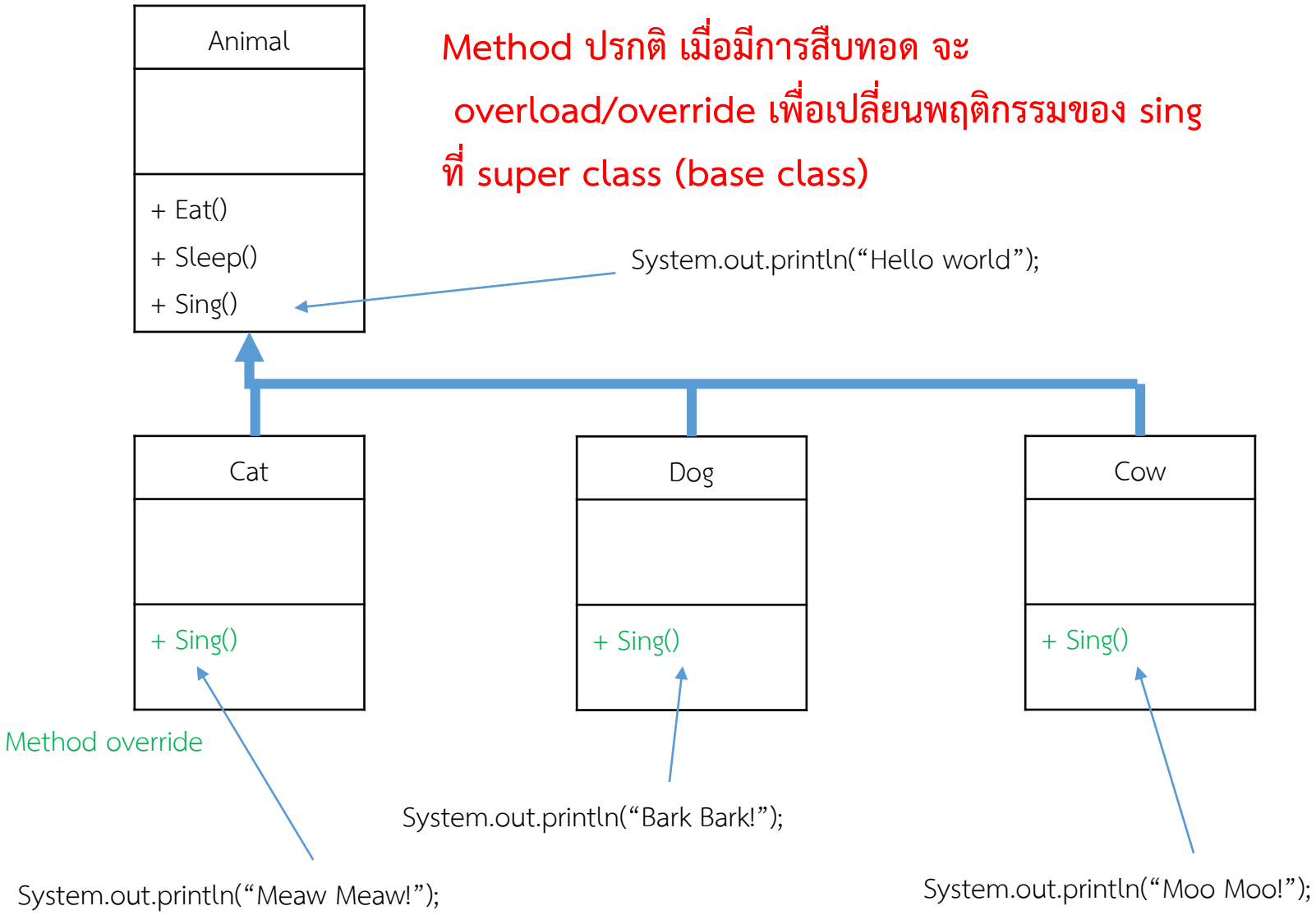
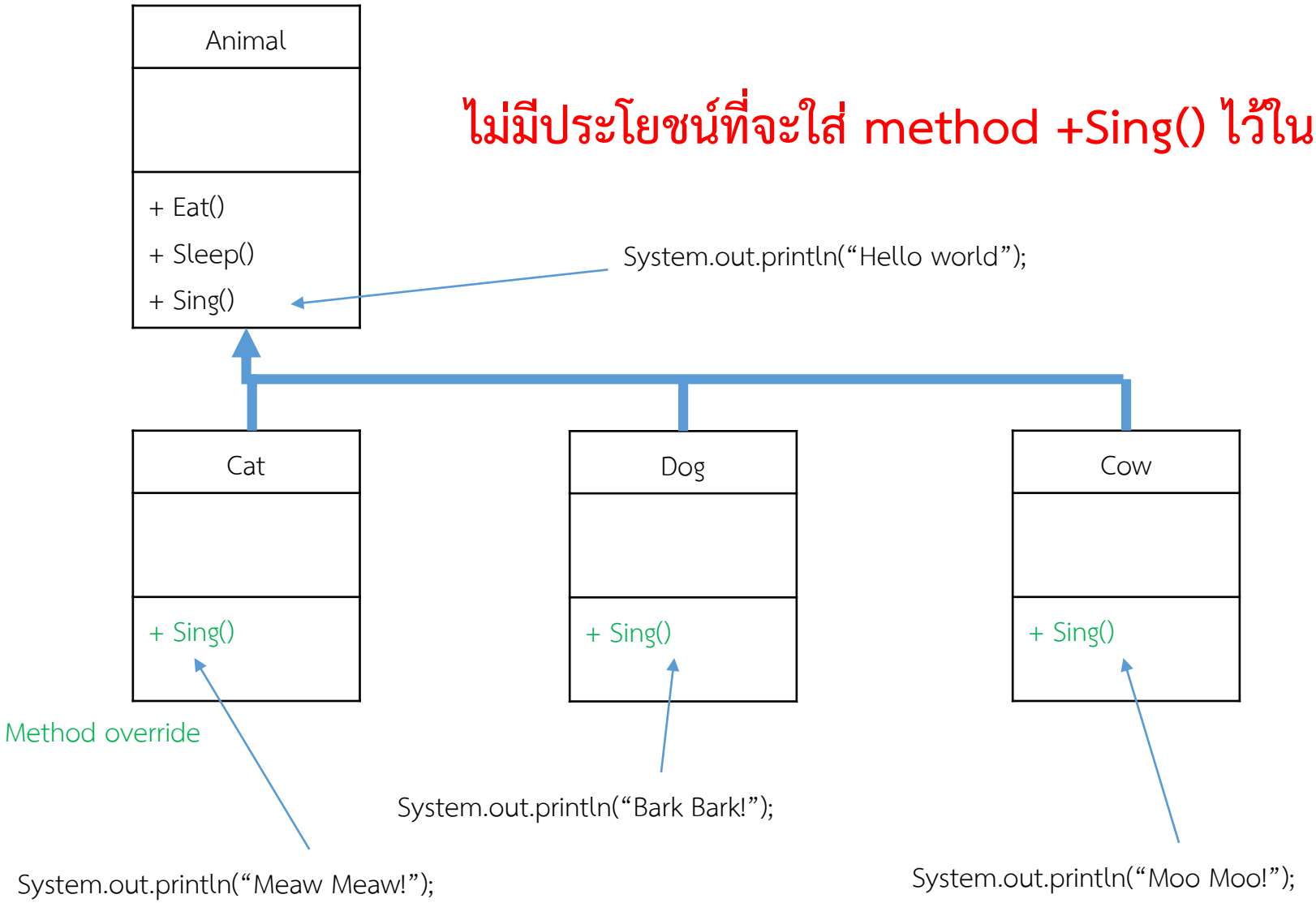
- CS117 OOP
- Chapter 4
- Abstraction

By Dr. Paween Khoenkaw
Computer Science MJU

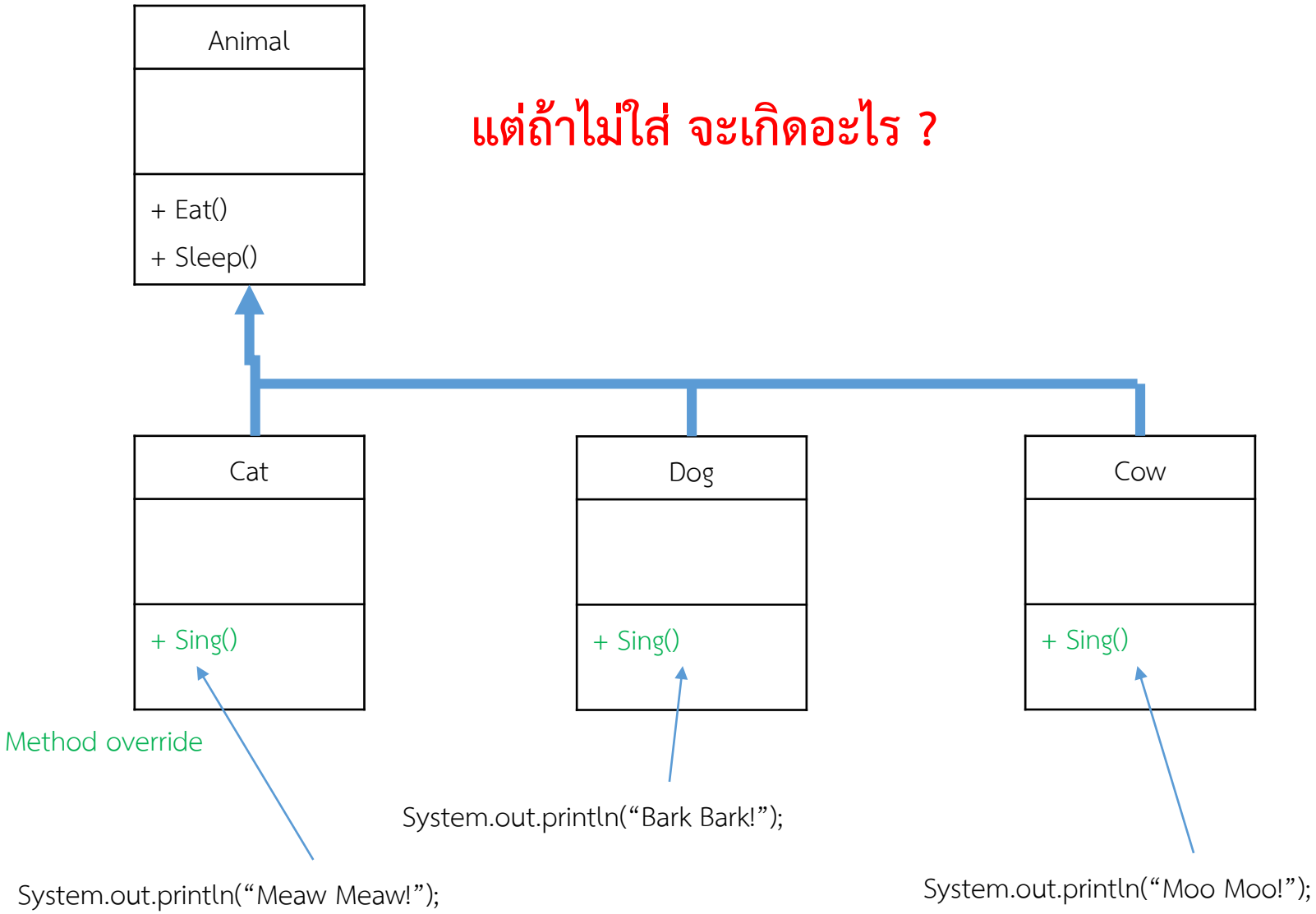
Abstraction



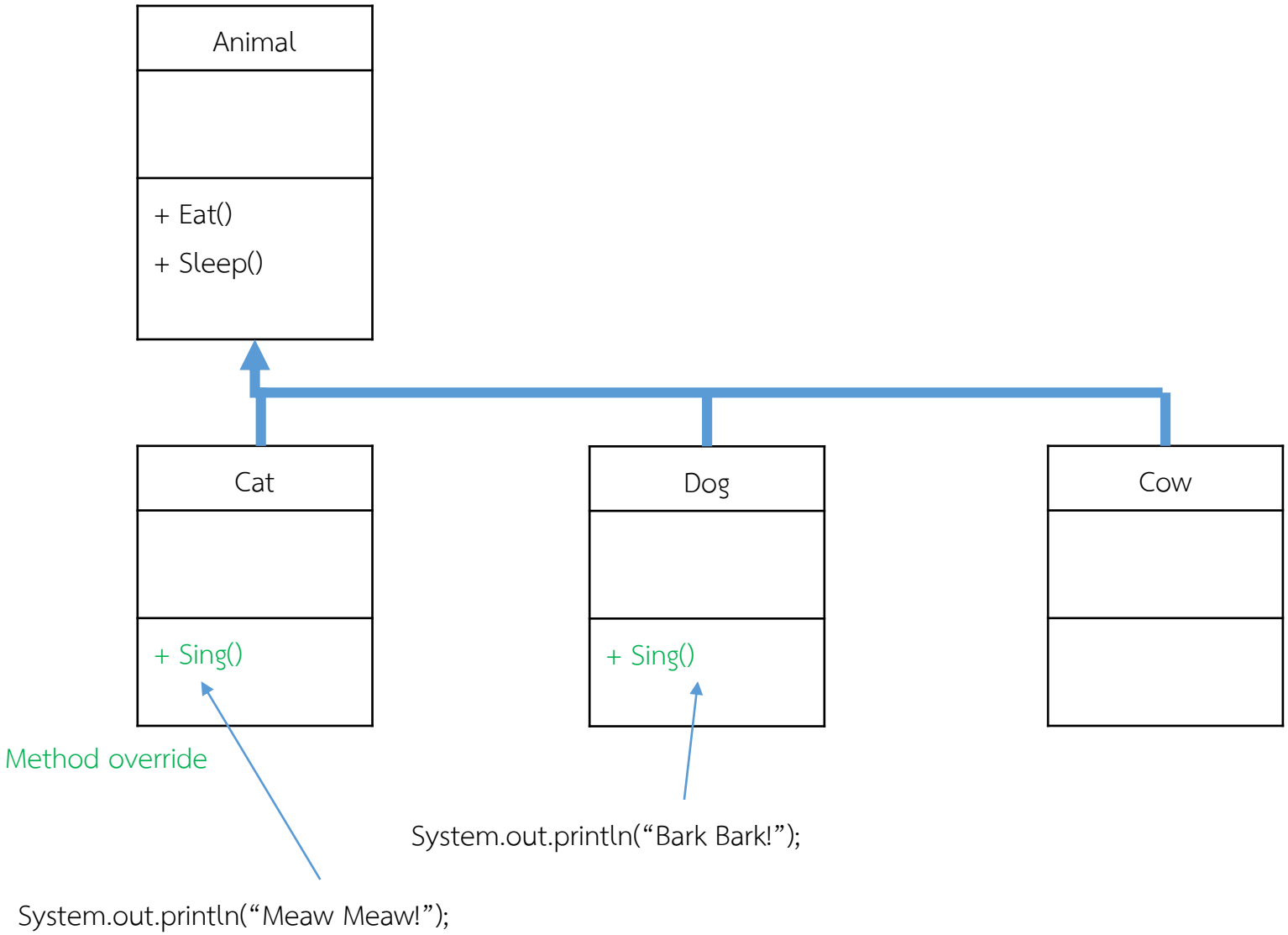
Abstraction



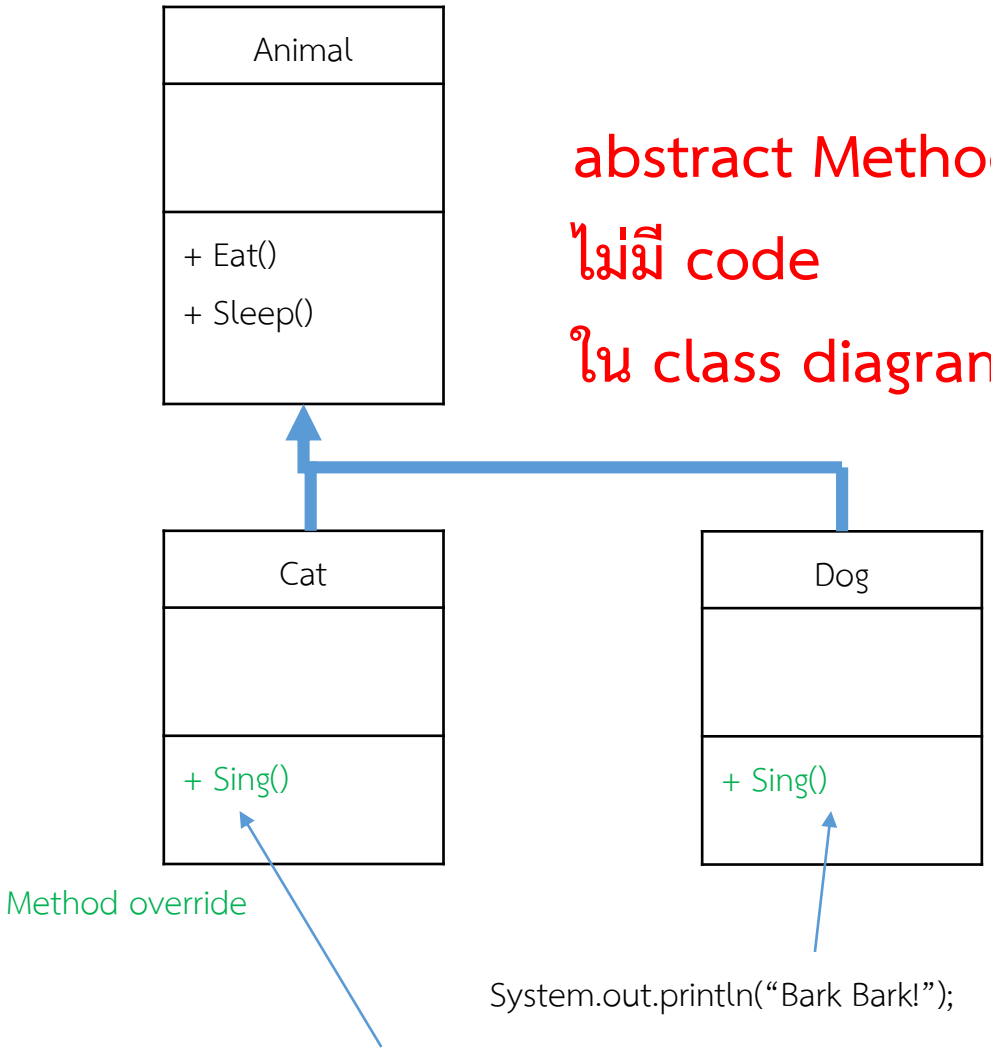
Abstraction



Abstraction



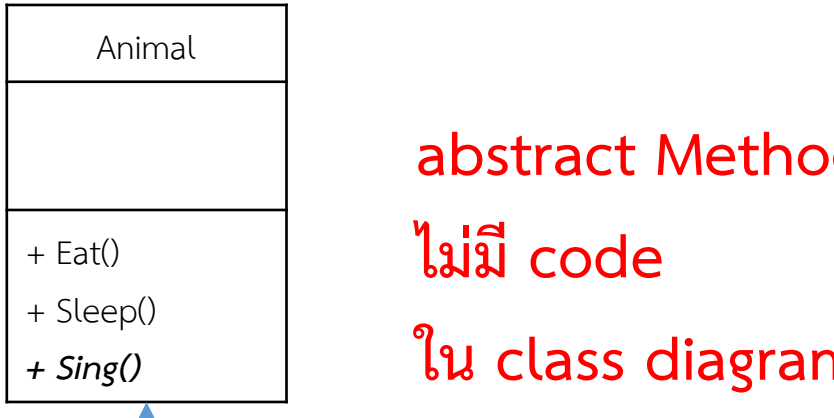
Abstraction



abstract Method คือ method เปล่าๆที่มีแต่โครง ไม่มี code

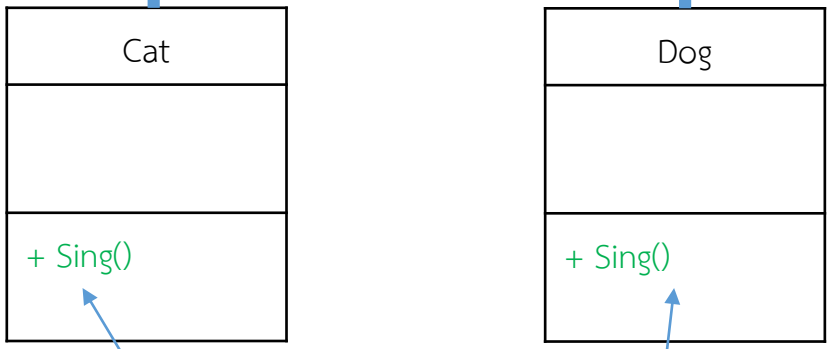
ใน class diagram จะแทนด้วย *ตัวเอียง*

Abstraction



abstract Method คือ method เปล่าๆที่มีแต่โครง ไม่มี code

ใน class diagram จะแทนด้วย *ตัวเอียง*

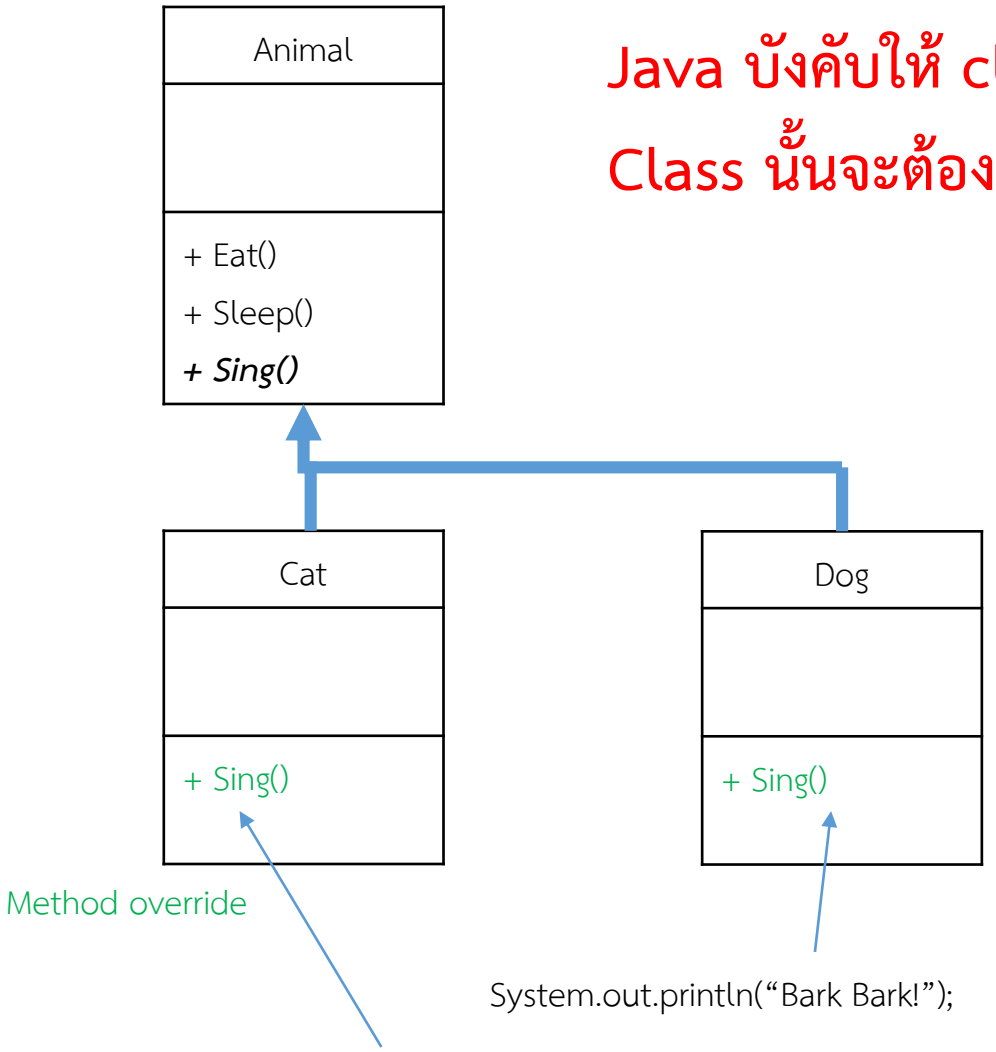


Method override

System.out.println(“Bark Bark!”);

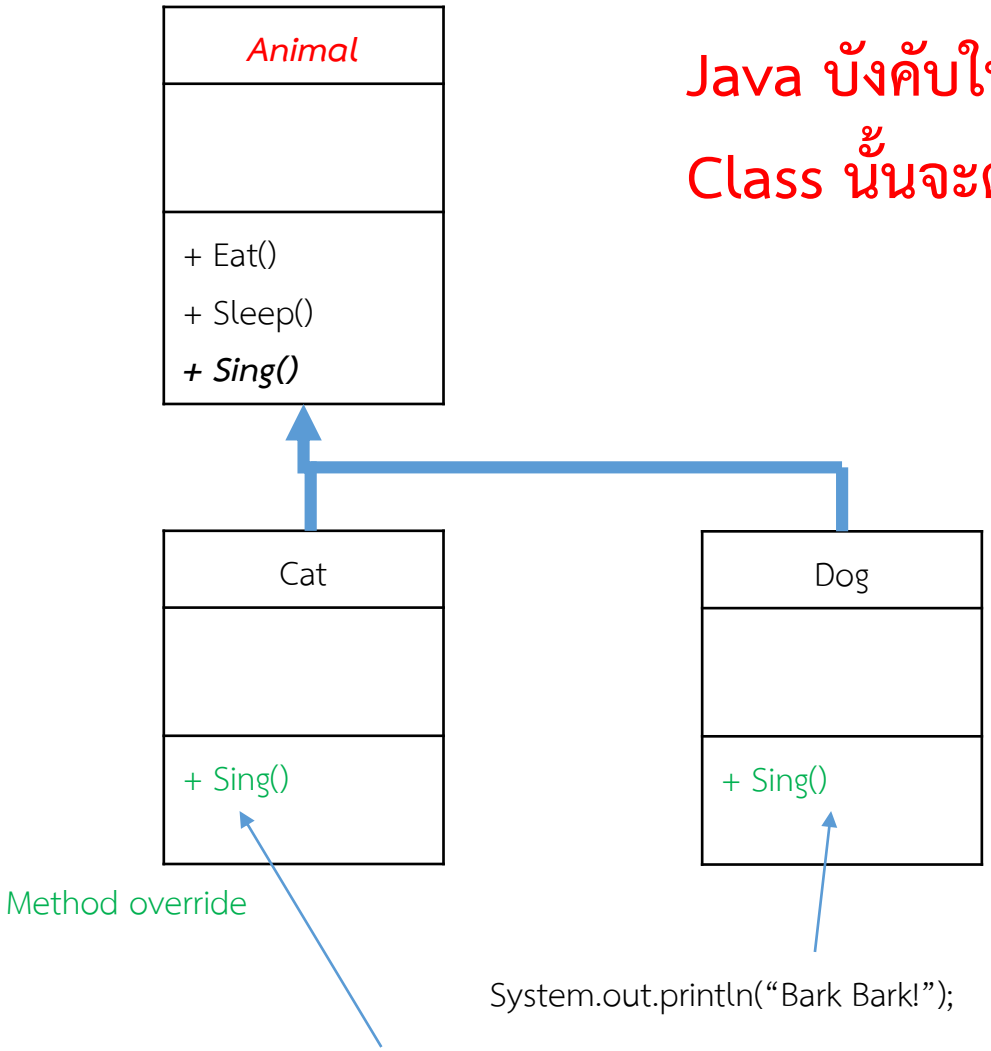
System.out.println(“Meaw Meaw!”);

Abstraction



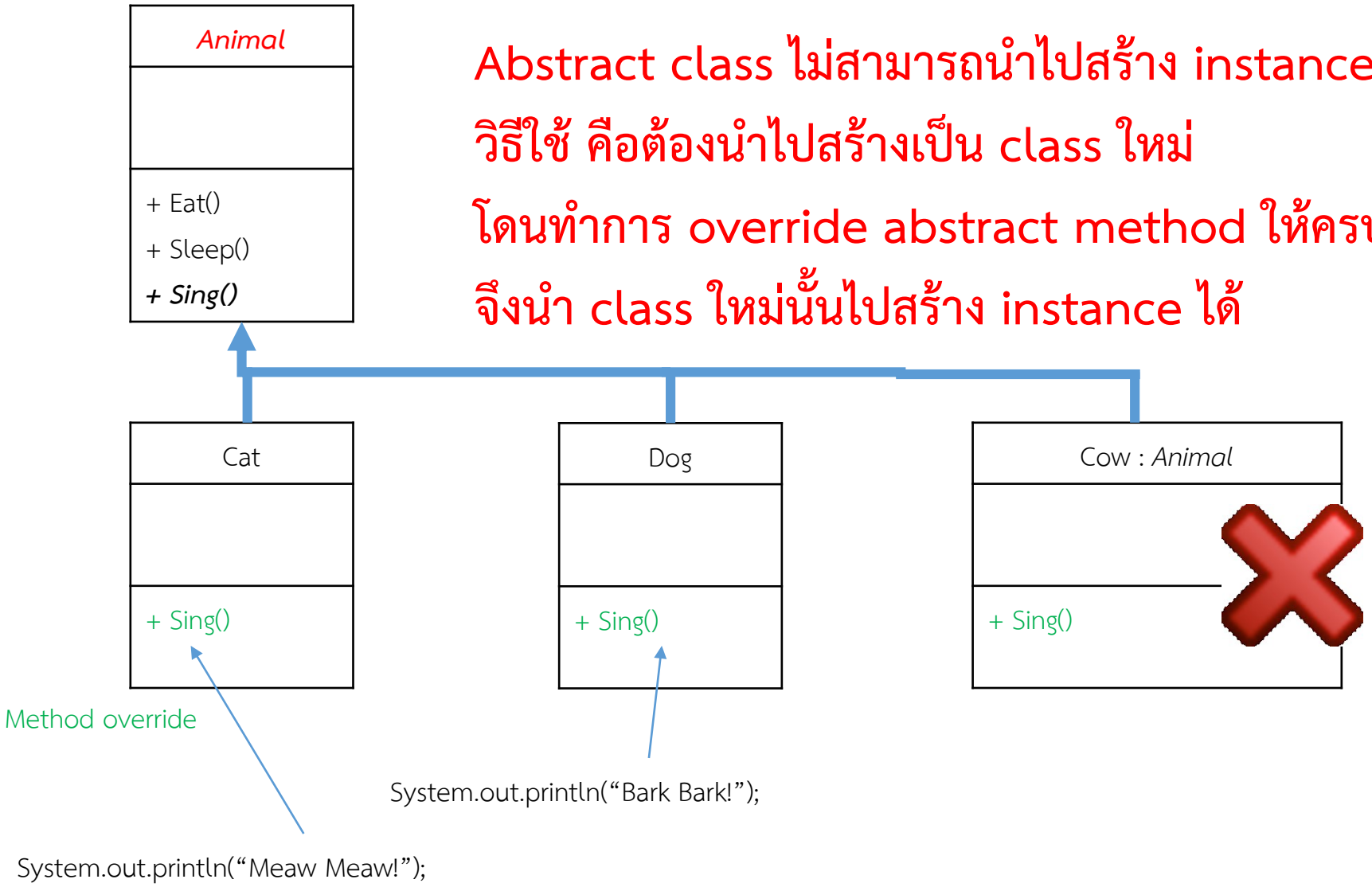
Java บังคับให้ class ใดมี abstract method
Class นั้นจะต้องเป็น abstract class ด้วย

Abstraction

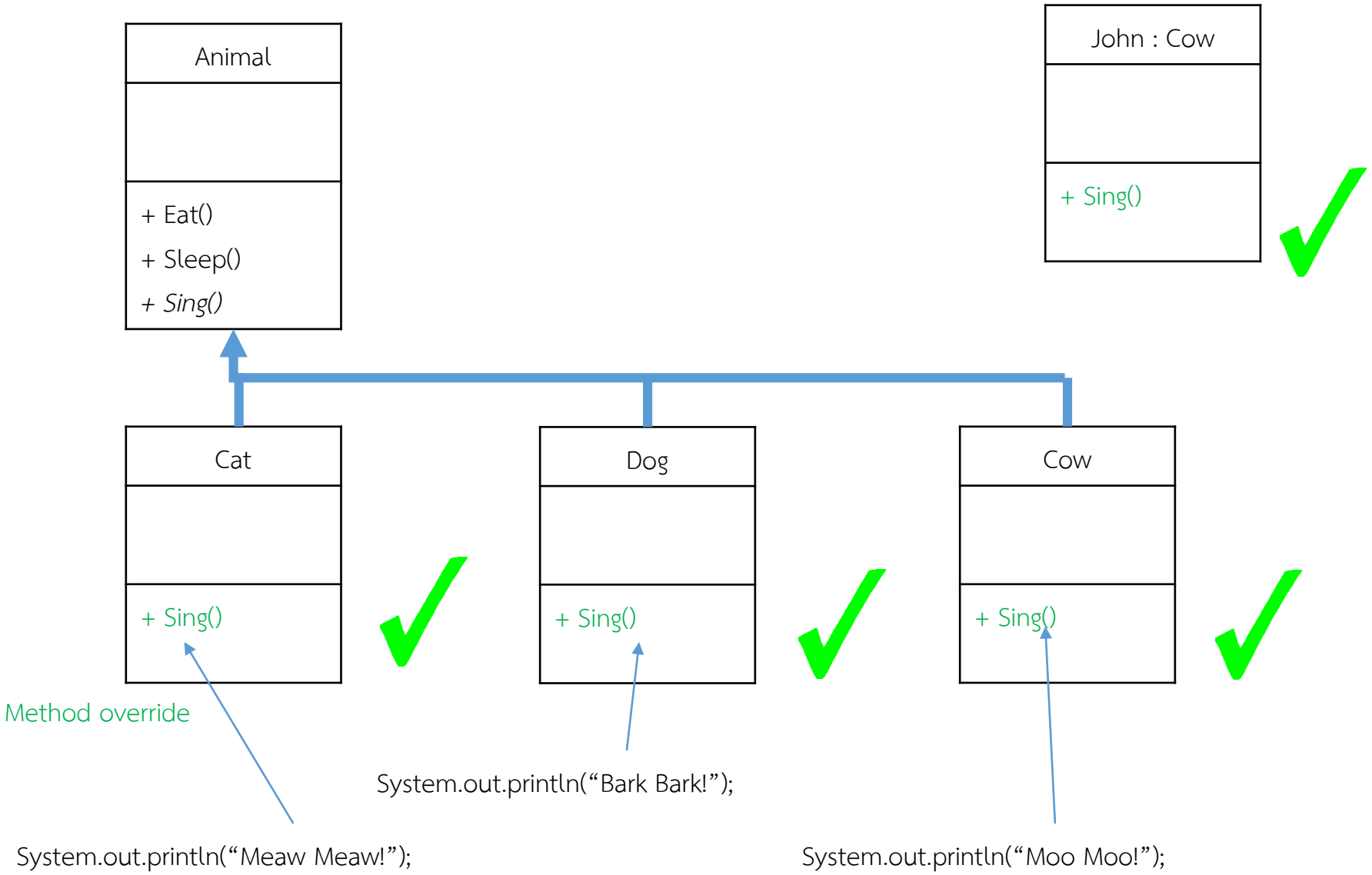


Java บังคับให้ class ใดมี abstract method Class นั้นจะต้องเป็น abstract class ด้วย

Abstraction



Abstraction



การทำ override กับ Abstract method เรียกอีกชื่อหนึ่งคือการ Implement

ทฤษฎีของการโปรแกรมเชิงวัตถุเรียกชื่อต่างกัน

สร้าง method ใหม่ที่เชื่อเหมือน method เดิมและพารามิเตอร์เหมือนเดิมเรียกว่า

Override

สร้าง method ใหม่ที่เชื่อเหมือน method เดิมแต่พารามิเตอร์แตกต่างจากเหมือนเดิม

เรียกว่า

Overload

สร้าง method ใหม่ที่เชื่อเหมือน abstract method เดิมเรียกว่า

Implement

ทั้งหมดนี้โปรแกรมภาษา JAVA ใช้คำเดียวคือ @Override

Interface คือ คลาสที่ method ทั้งหมดเป็น abstract method และไม่มี field หรือ properties ไม่มีสถานะ จะเรียกว่า Interface

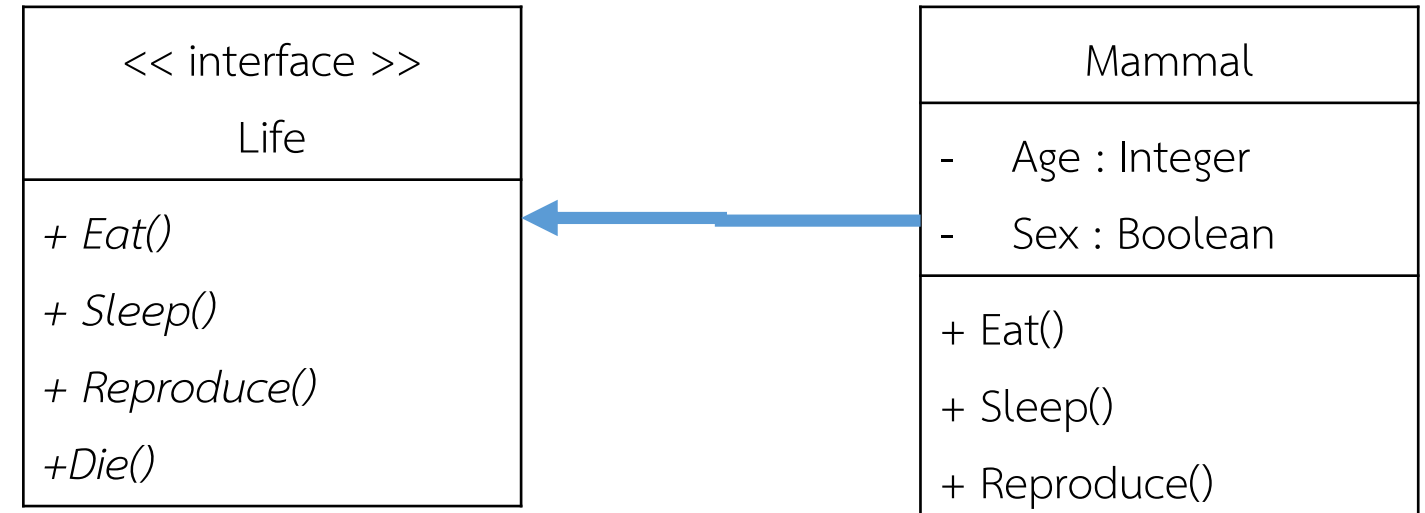
Interface นิยมใช้สำหรับสร้างต้นแบบของคลาส (prototype)

Class หรือ Interface สามารถสืบทอด จาก interface ได้มากกว่า 1 ตัวได้

Class จะไม่สามารถสืบทอดคลาสได้มากกว่า 1 ตัว

Interface นั้นมีเฉพาะในภาษา JAVA

Abstraction:interface



interface

class

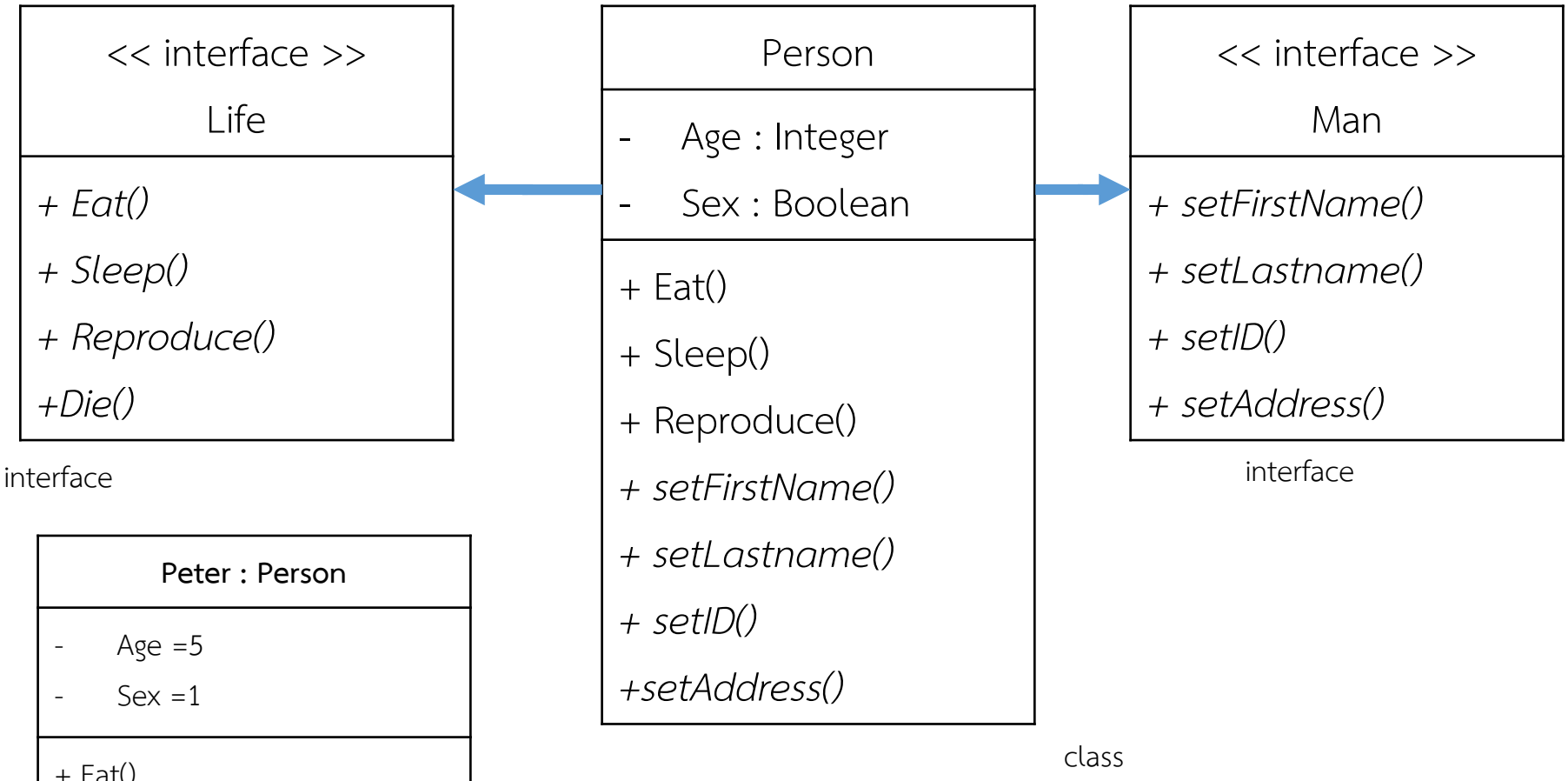
Cow : Mammal

- Age =5
- Sex =1

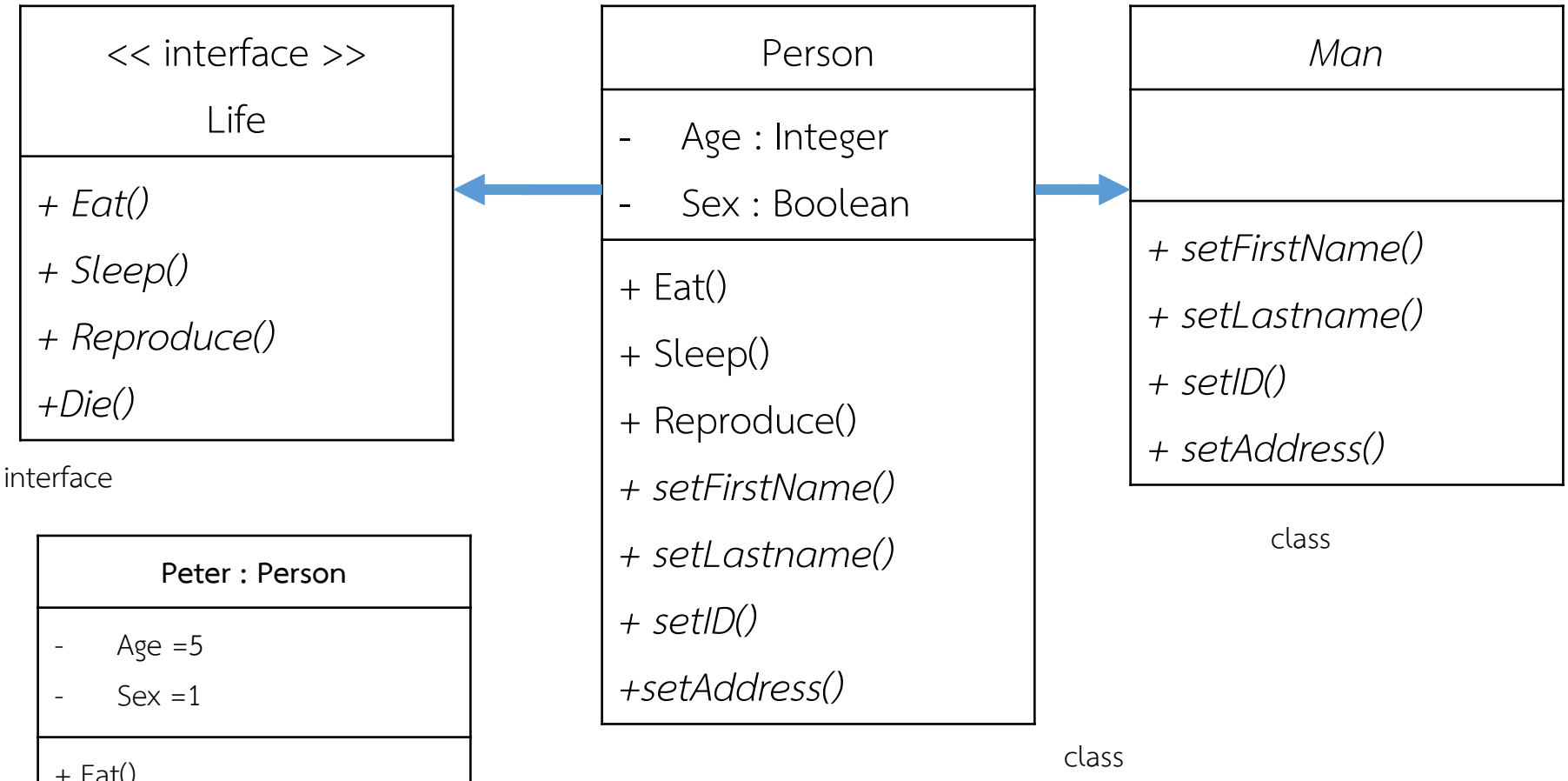
+ Eat()
+ Sleep()
+ Reproduce()
+Die()
+Sing()
+getMilk()
+setAge()

instance

Abstraction:interface

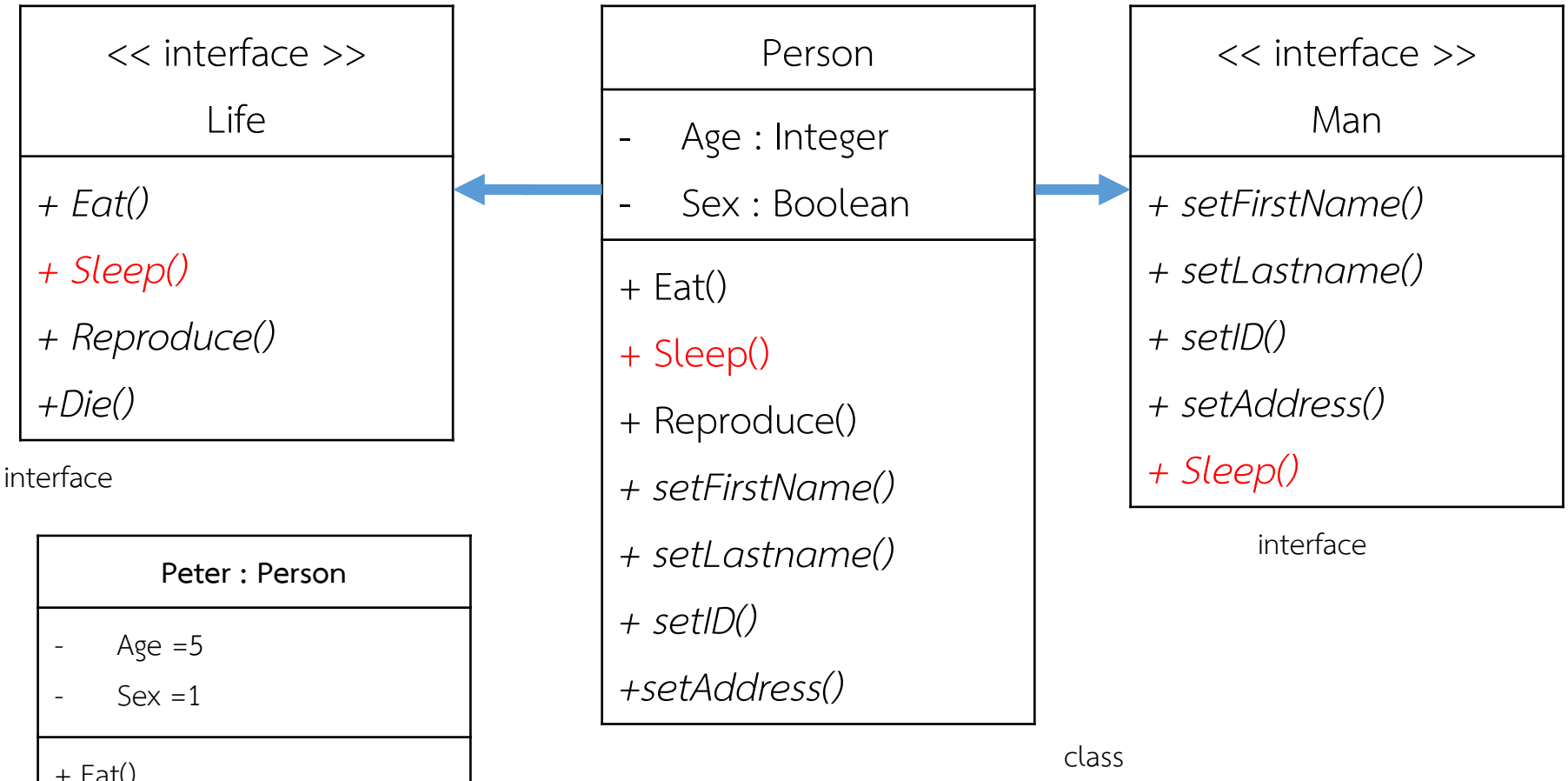


Abstraction:interface



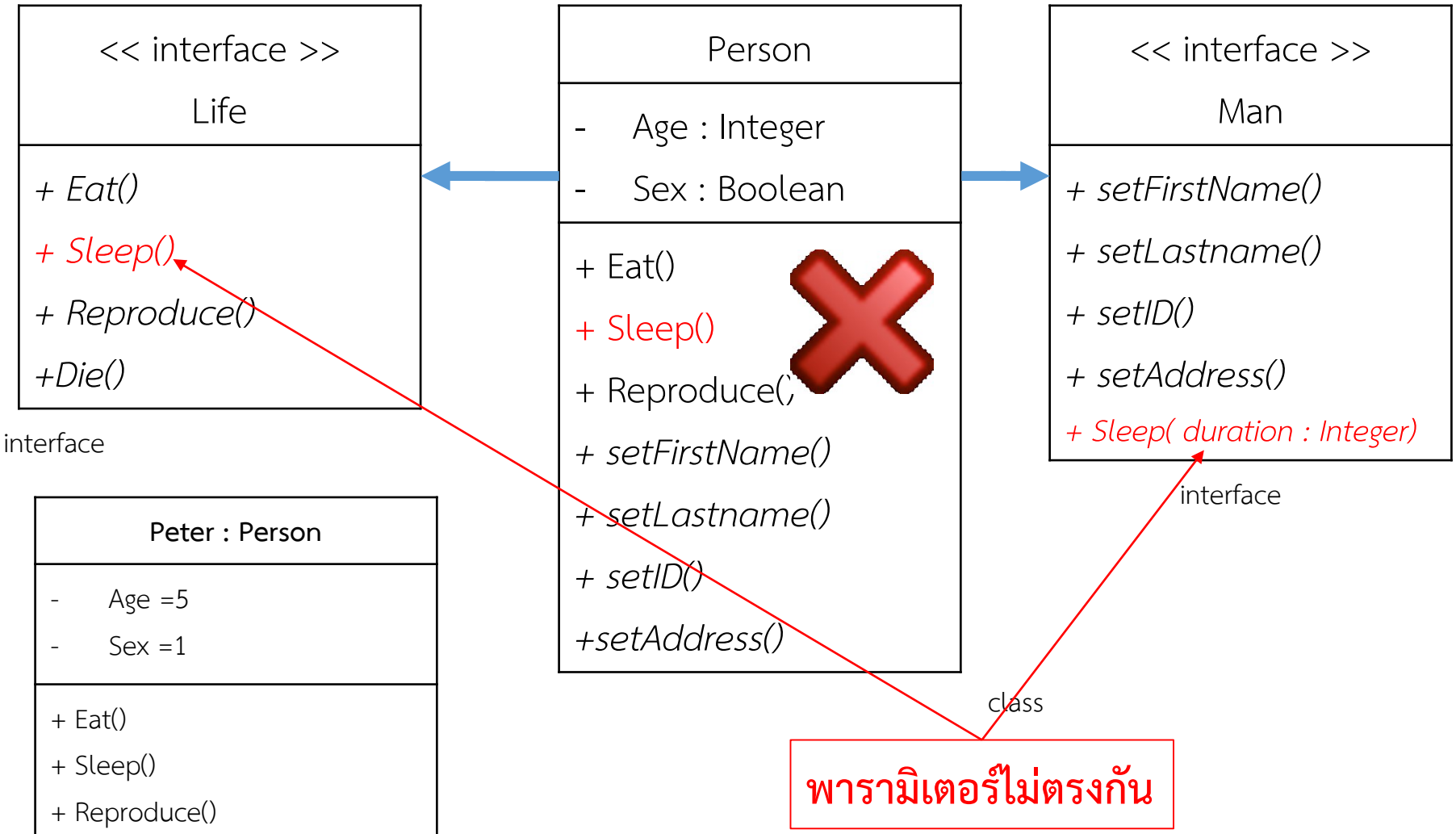
Person สืบทอด 1 interface และ 1 class สามารถทำได้

Abstraction:interface



การสืบทอดมากกว่า 1 interface สามารถทำได้
เราะทำ method override ได้
เนื่องจากทั้ง 2 method เหมือนกันและเป็น abstract จึงไม่มี code

Abstraction:interface



การสืบทอดมากกว่า 1 interface

ไม่สามารถทำ method overload ได้

Abstraction:interface

Life
+ Eat() *+ Sleep()* *+ Reproduce()* *+Die()*

class

Person
- Age : Integer - Sex : Boolean
+ Eat() + Sleep() + Reproduce() *+ setFirstName()* *+ setLastname()* *+ setID()* *+setAddress()*

class

Man
+ setFirstName() *+ setLastname()* *+ setID()* *+ setAddress()*

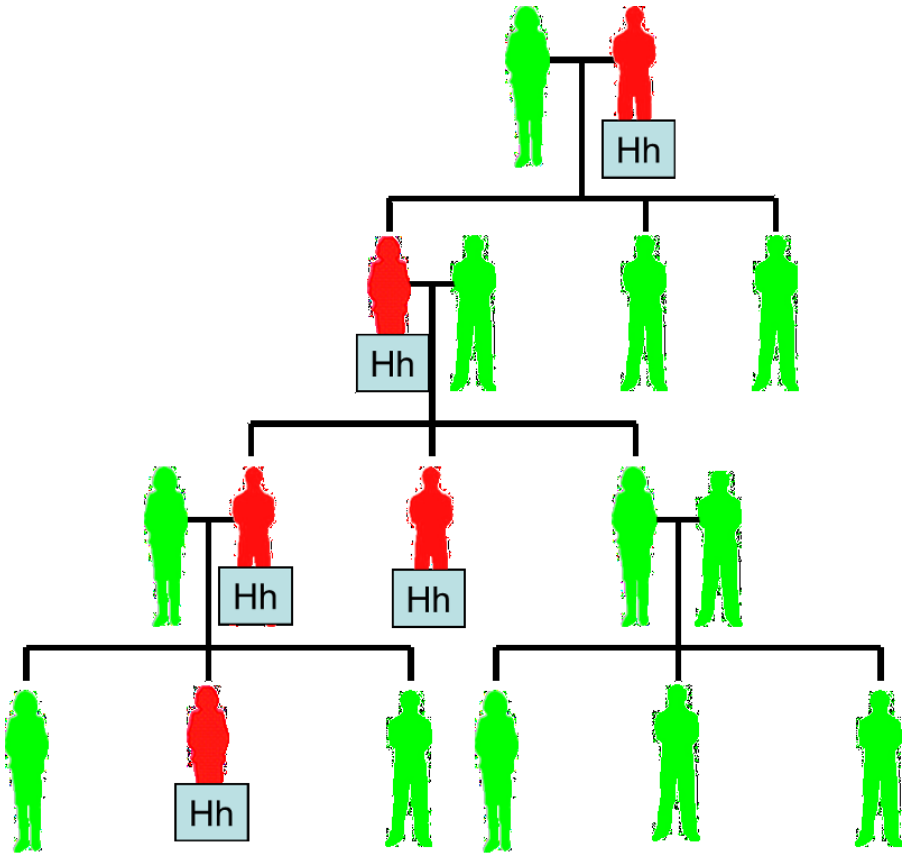
class

Peter : Person
- Age =5 - Sex =1
+ Eat() + Sleep() + Reproduce() + setFirstName() + setLastname() + setID() + setAddress()

instance

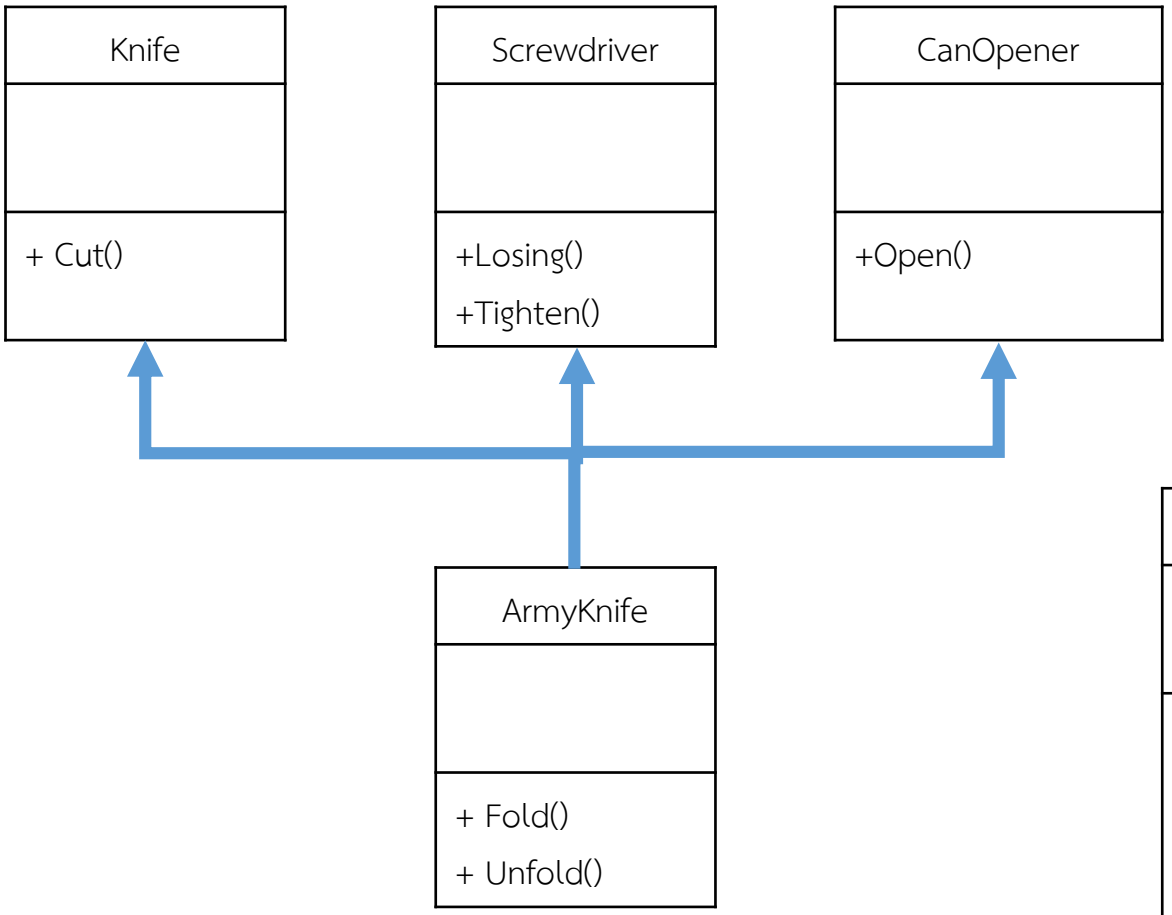
Person สืบทอด 2 class ไม่สามารถทำได้

การสืบทอดจากหลายคลาส



Multiple Inheritance

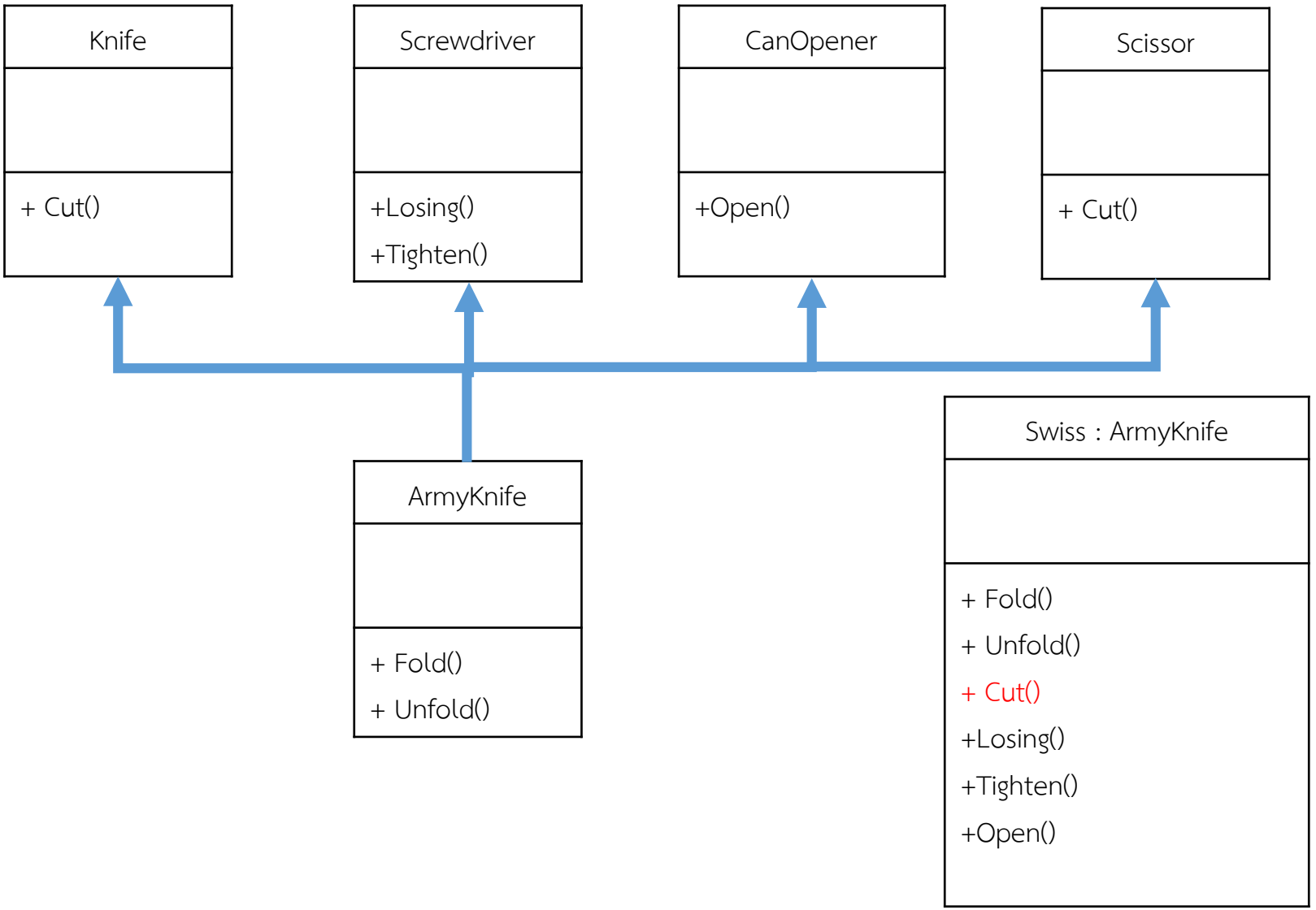
การสืบทอดจากหลายคลาส



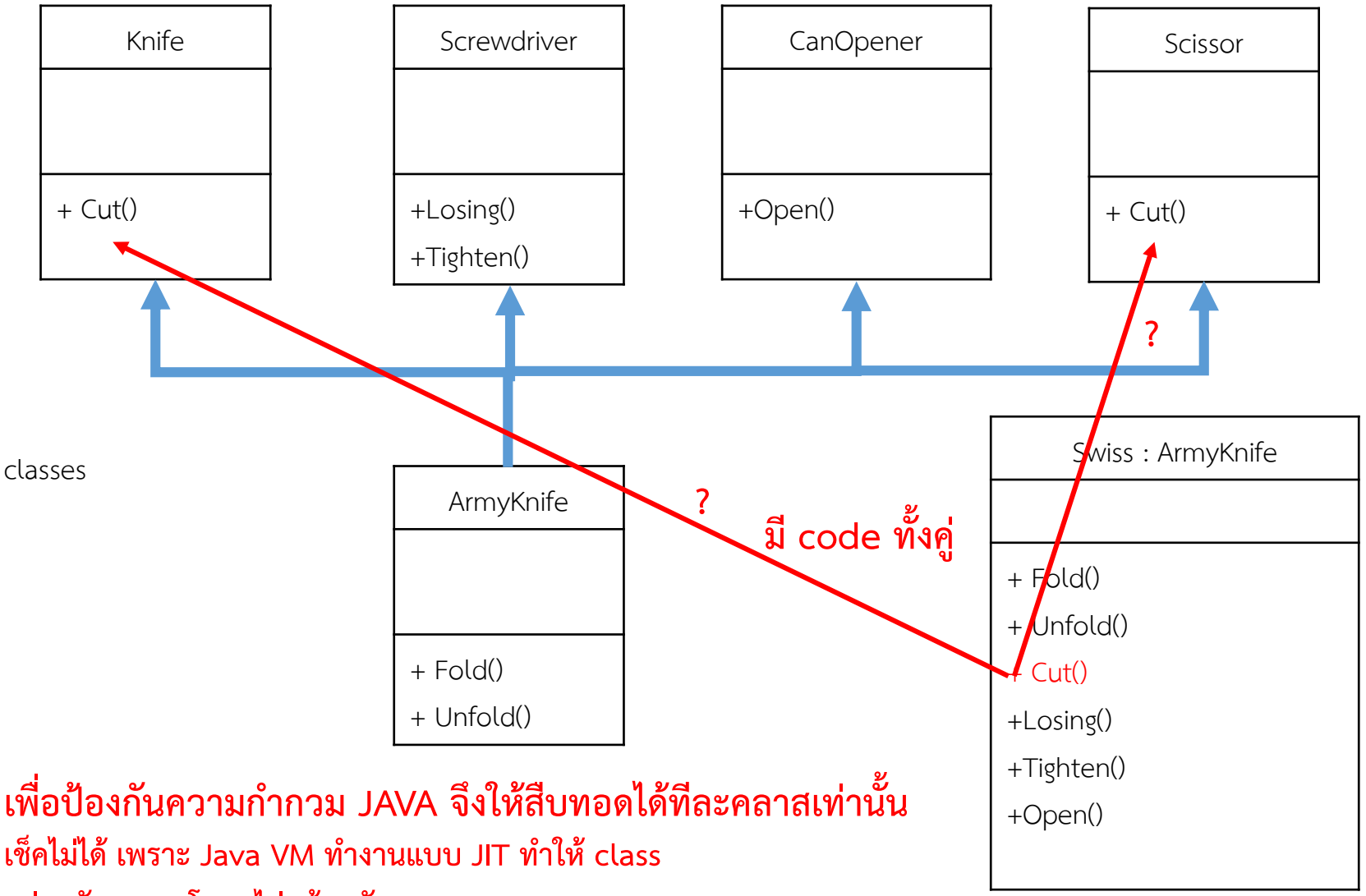
Swiss : ArmyKnife

+ Fold()
+ Unfold()
+ Cut()
+Losing()
+Tighten()
+Open()

การสืบทอดจากหลายคลาส และ ภัยที่ซ่อนเร้น



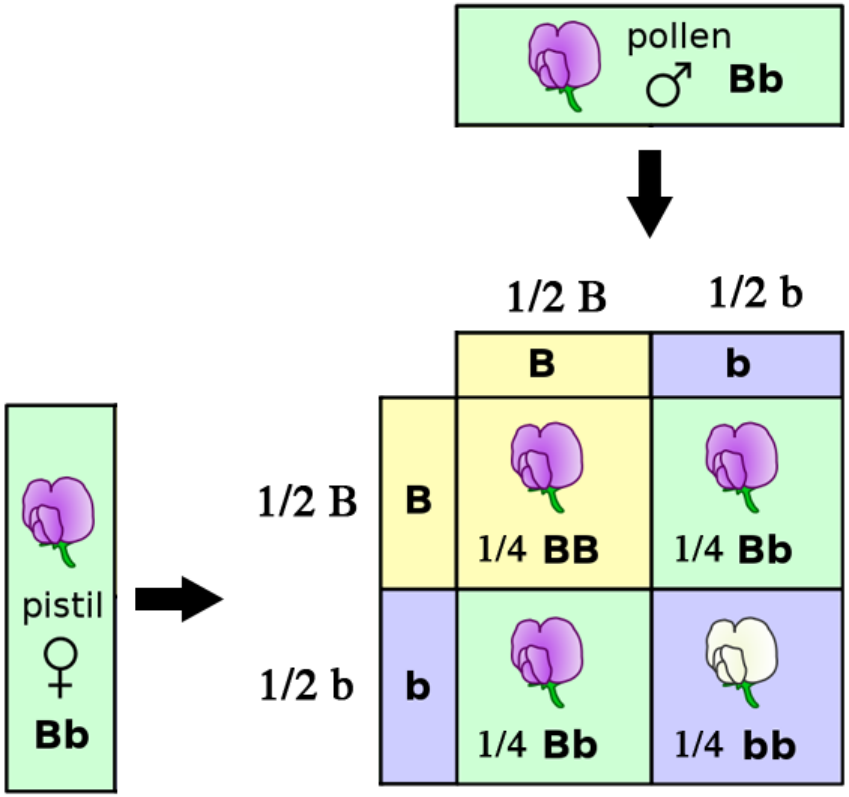
Multiple Inheritance

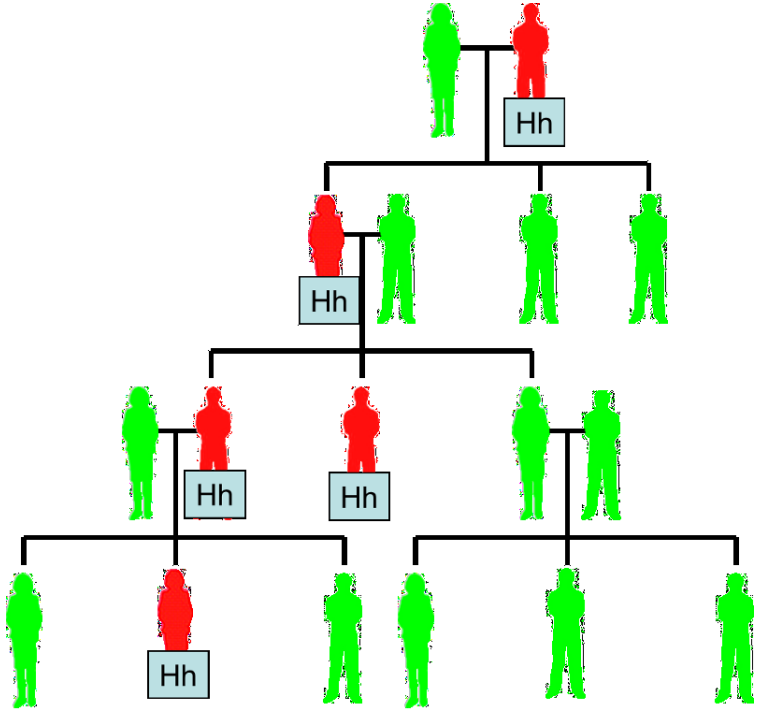


เพื่อป้องกันความกำกวม JAVA จึงให้สืบทอดได้ทีละคลาสเท่านั้น

เช็คไม่ได้ เพราะ Java VM ทำงานแบบ JIT ทำให้ class แต่ละตัวอาจถูกโหลดไม่พร้อมกัน

Multiple Inheritance



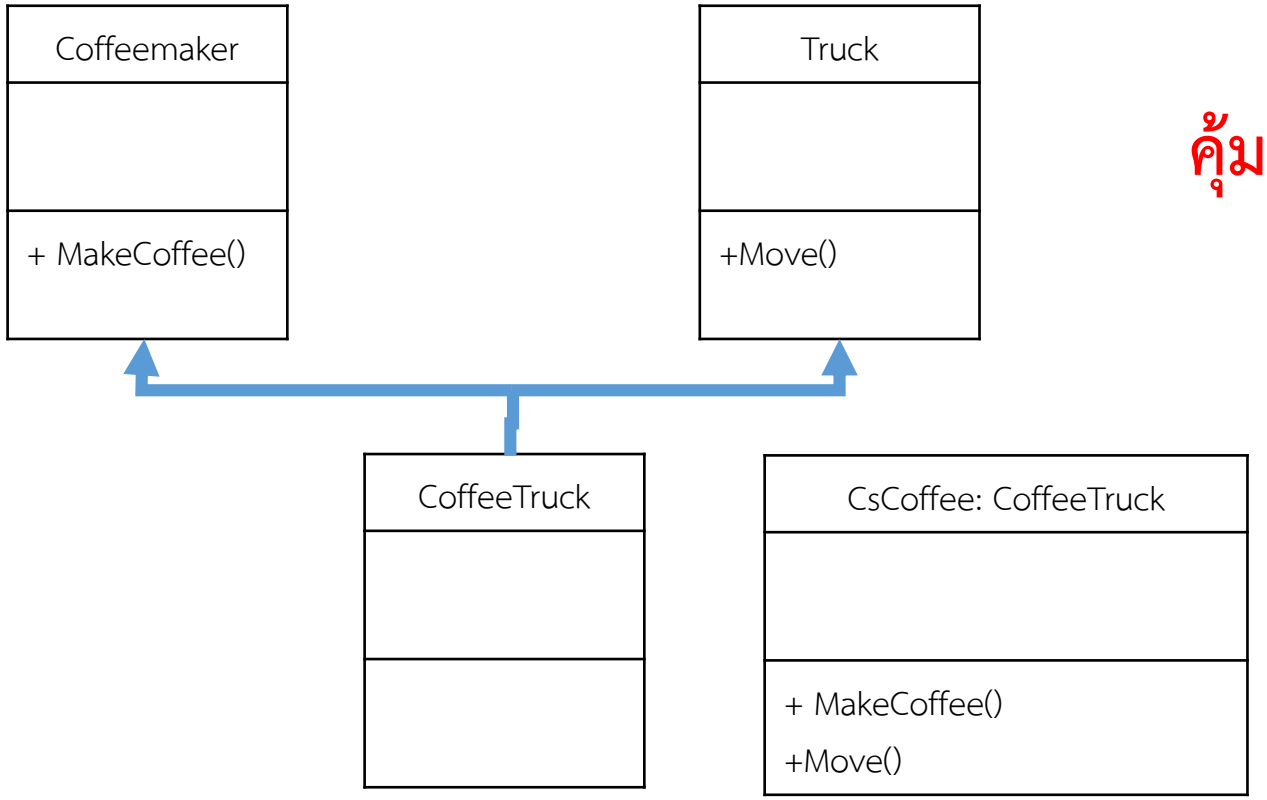


คอมพิวเตอร์คือเครื่องจักรที่มีการทำงานที่แน่นอน (Deterministic Machine)
รันทุกครั้งต้องคาดเดาผลลัพธ์ได้ สถานการณ์เหมือนกันผลลัพธ์ต้องเหมือนกันทุกครั้ง
เหตุการณ์นี้จึงผิดหลักทฤษฎีการคำนวณ (Theory of Computation)

Multiple Inheritance

เพื่อป้องกันความกำกวม JAVA จึงให้สืบทอดได้ทีละคลาสเท่านั้น

เรายอมแลกความสะดวก กับ ความปลอดภัย

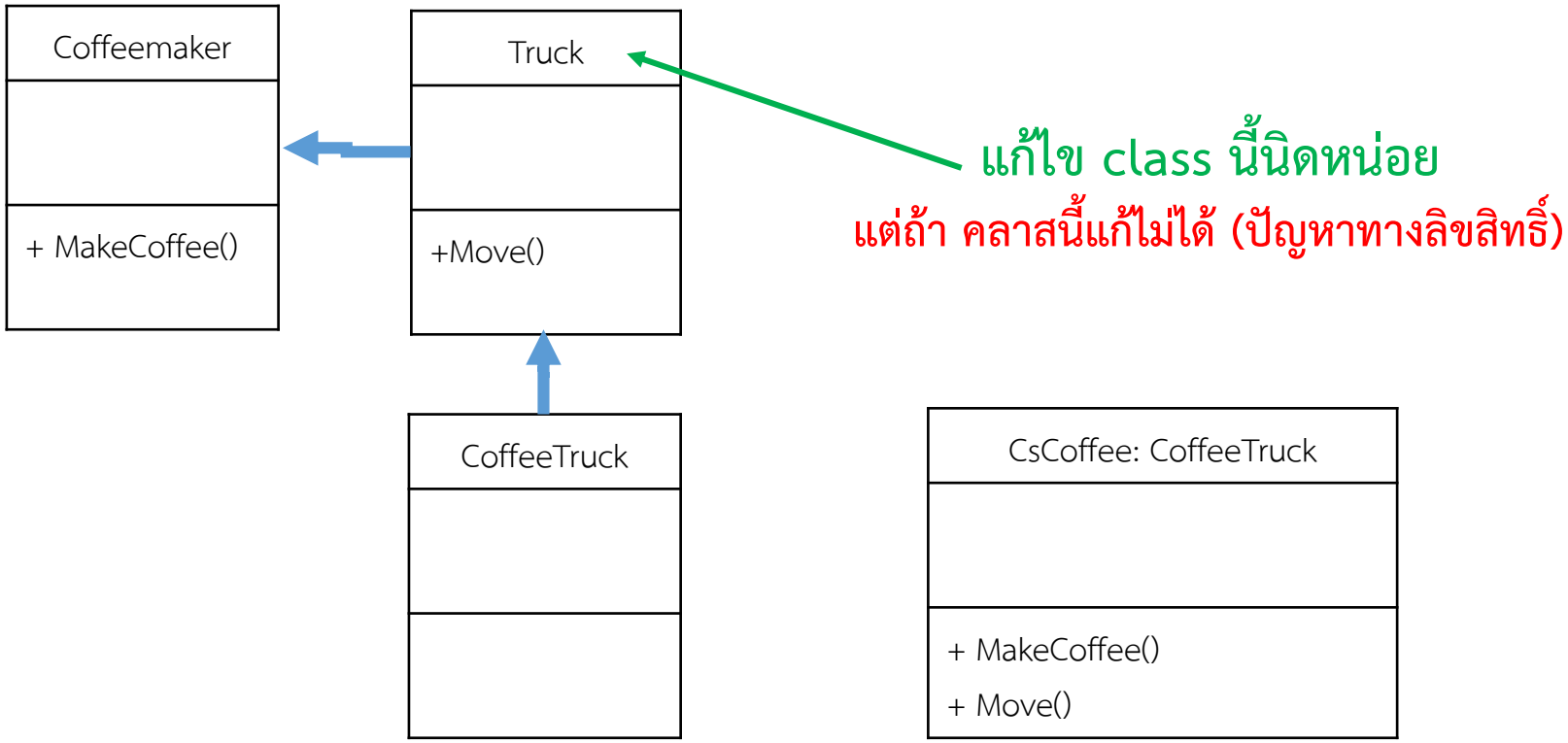


คุ้มหรือไม่ ?

C++ ทำได้ แต่ Java ทำไม่ได้

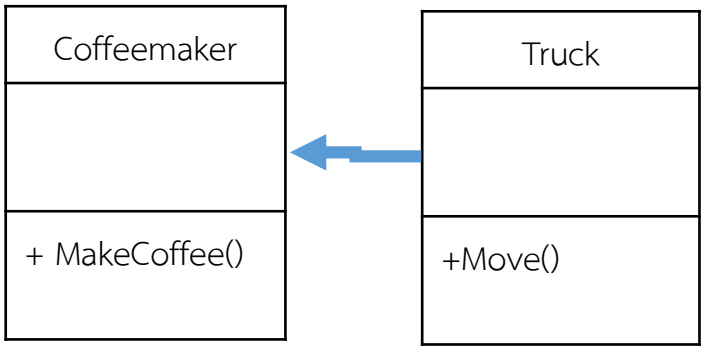
Multiple Inheritance

JAVA สามารถสืบทอดหลายคลาสได้ ด้วยวิธีสืบทอดทีละลำดับ



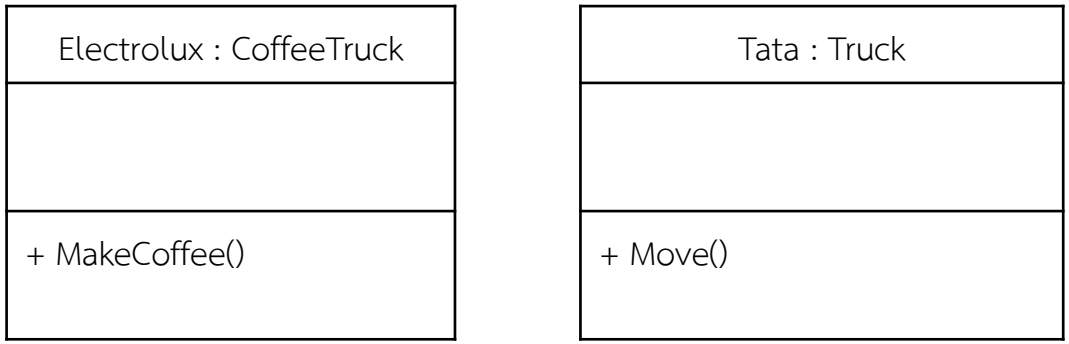
มีค่าเท่า สืบทอดจากหลายคลาส แต่ปลอดภัยกว่า

Multiple Inheritance



ก็ไม่ต้องสืบทอด เราสามารถเอาคลาสทั้ง 2 มาสร้างเป็น Instance อย่างละตัวแล้วค่อยใช้งานก็ได้

เรียกว่า Object composition




```
Coffeemaker Electrolux=new Coffeemaker();
Truck Tata = new Truck();
Electrolux.MAkeCoffee();
Tata.Move();
```

สรุปสิ่งที่เราเรียนไปแล้ว

Modularity (คลาสและ Instance หรือ Object)

Encapsulation (การซ่อน Field, Method และการป้องกันการเข้าถึง)

Inheritance (การสืบทอด)

Abstraction (การกำหนดต้นแบบ , interface , implement)

Polymorphism (Method ที่เหมือนกัน , Override, Overload)

การบ้าน 4

ให้ทุกคนกลับไปอ่านทบทวนหัวข้อที่เรียนมา

สอบกลางภาควันจันทร์ที่ 31 กรกฎาคม 2560

15.30-18.30 ห้อง 80-505อาคารเรียนรวม 80 ปี

ข้อสอบมีทั้งกา และ เติมคำ

ห้ามนำเครื่องคิดเลขเข้าห้องสอบ

จดเข้าห้องสอบได้ 1 แผ่น A4 เขียนได้ทั้ง 2 หน้า

ขอให้ทุกคนโชคดี