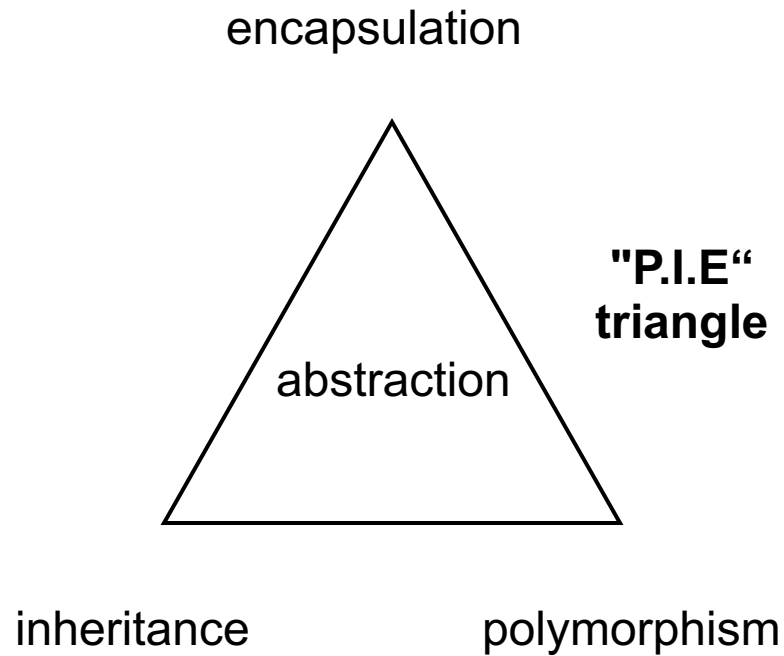


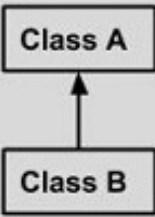
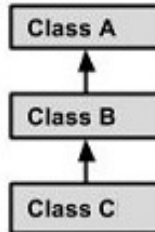
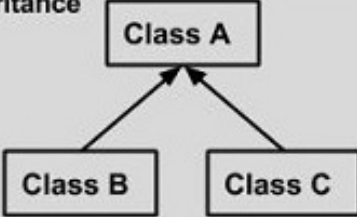
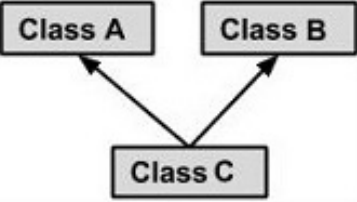
Object-Oriented Programming

Interfaces in Java

Important OO Concepts



Why care about Interface?

Single Inheritance	 <pre>graph BT; B[Class B] --> A[Class A]</pre>	<pre>public class A { } public class B extends A { }</pre>
Multi Level Inheritance	 <pre>graph BT; C[Class C] --> B[Class B]; B --> A[Class A]</pre>	<pre>public class A {} public class B extends A {} public class C extends B { }</pre>
Hierarchical Inheritance	 <pre>graph BT; B[Class B] --> A[Class A]; C[Class C] --> A</pre>	<pre>public class A {} public class B extends A {} public class C extends A { }</pre>
Multiple Inheritance	 <pre>graph BT; A[Class A] --> C[Class C]; B[Class B] --> C</pre>	<pre>public class A {} public class B {} public class C extends A,B { } // Java does not support multiple Inheritance</pre>

Supported via the use of **Interface**

What is Interface?

- In Java, interface is a special type of class which:
 - Define a set of **method prototypes**
 - Does **not** provide the **implementation** for the prototypes
 - Can also define final constants

```
public interface Animal {  
    public abstract void eat();  
    public abstract void travel();  
}
```

Creating Interface

- To **define** an interface:

```
public interface Animal {  
    public abstract void eat();  
    public abstract void travel();  
}
```

Use keyword **interface** instead of **class**

the methods are ALL abstract

- To **implement** an interface:

```
public class Mammal implements Animal {  
    public void eat(){  
        System.out.println("Mammal eats meat");  
    }
```

keyword **implements**

```
    public void travel(){  
        System.out.println("Mammal travels around");  
    }
```

implements ALL Animal methods

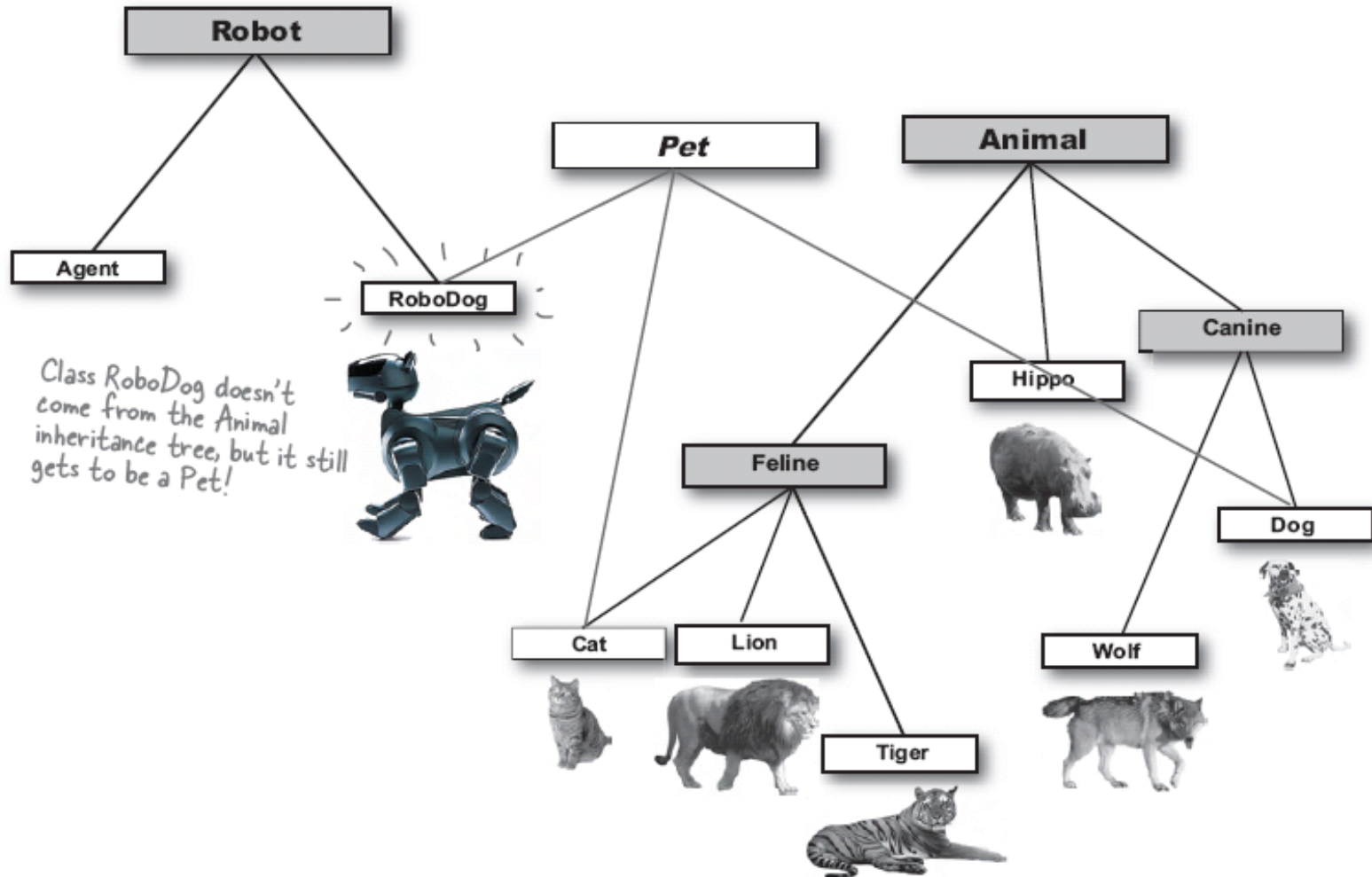
```
    public int noOfLegs(){  
        return 0;  
    }
```

normal overriding methods

```
}
```

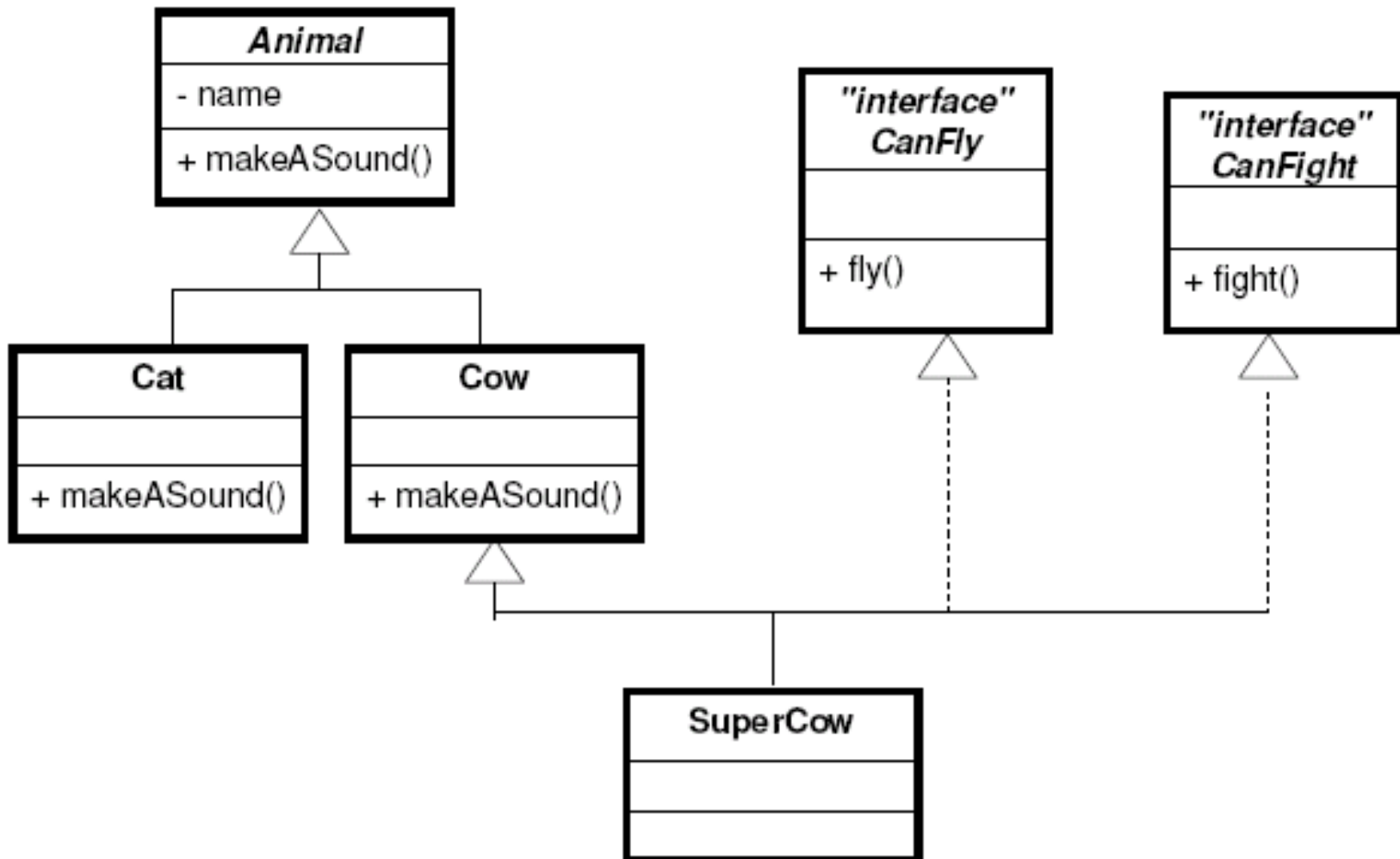
Multiple Inheritance with Interface

- Classes from different inheritance trees can implement the same interface



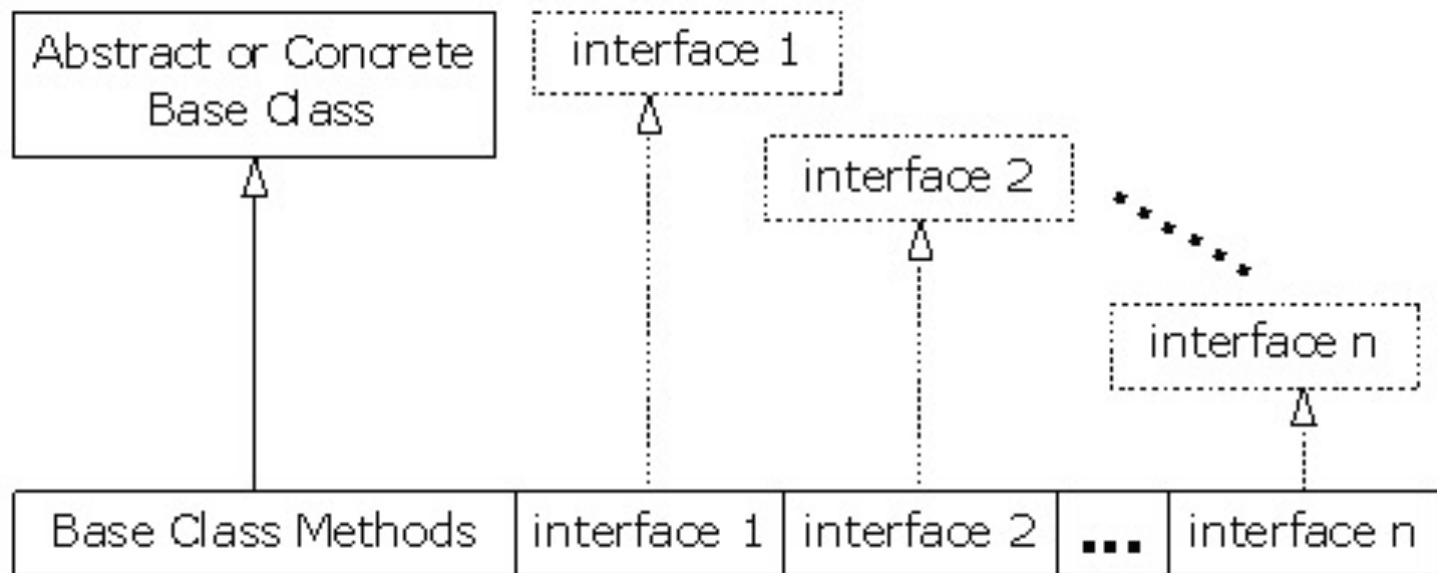
Multiple Inheritance with Interface

- A class can implement multiple interfaces

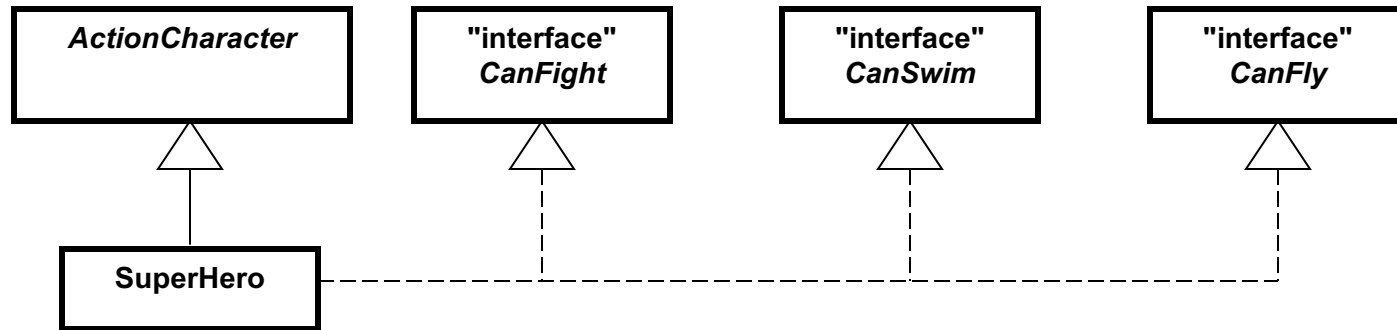


Extends vs. Implements Keyword

- A class
 - Can “**extend**” only one class, i.e. ONE superclass
 - Can “**implement**” MULTIPLE interfaces



Implement Multiple Interfaces



- Class “SuperHero”
 - **Extends** class “ActionCharacter”
 - **Implements** three interfaces “CanFight”, “CanSwim”, “CanFly”

Implement Multiple Interfaces

```
interface CanFight {  
    void fight();  
}  
interface CanSwim {  
    void swim();  
}  
interface CanFly {  
    void fly();  
}  
class ActionCharacter {  
    public void fight() {  
        System.out.print("Fight well");  
    }  
}  
  
class SuperHero extends ActionCharacter implements CanFight,  
CanSwim, CanFly {  
    public void swim() {  
        System.out.print("Swim well");  
    }  
    public void fly() {  
        System.out.print("Fly well");  
    }  
}
```

Extend an Interface with Inheritance

```
interface Monster {  
    void menace();  
}  
interface Lethal {  
    void kill();  
}  
interface Vampire extends Monster, Lethal {  
    void drinkBlood();  
}
```

```
class VeryBadVampire implements Vampire {  
    public void menace() {  
        System.out.print("Vampire menaces people");  
    }  
    public void kill() {  
        System.out.print("Vampire kills people");  
    }  
    public void drinkBlood() {  
        System.out.print("Vampire drinks blood");  
    }  
}
```

