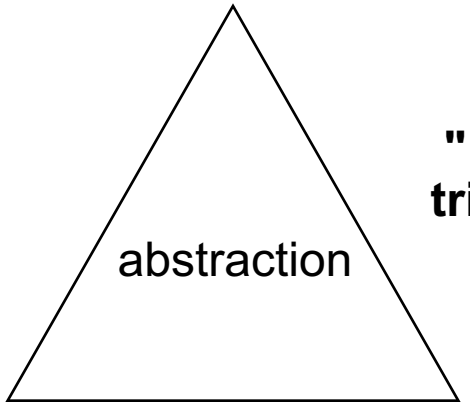
Object-Oriented Programming

# Interfaces in Java

# Important OO Concepts

encapsulation

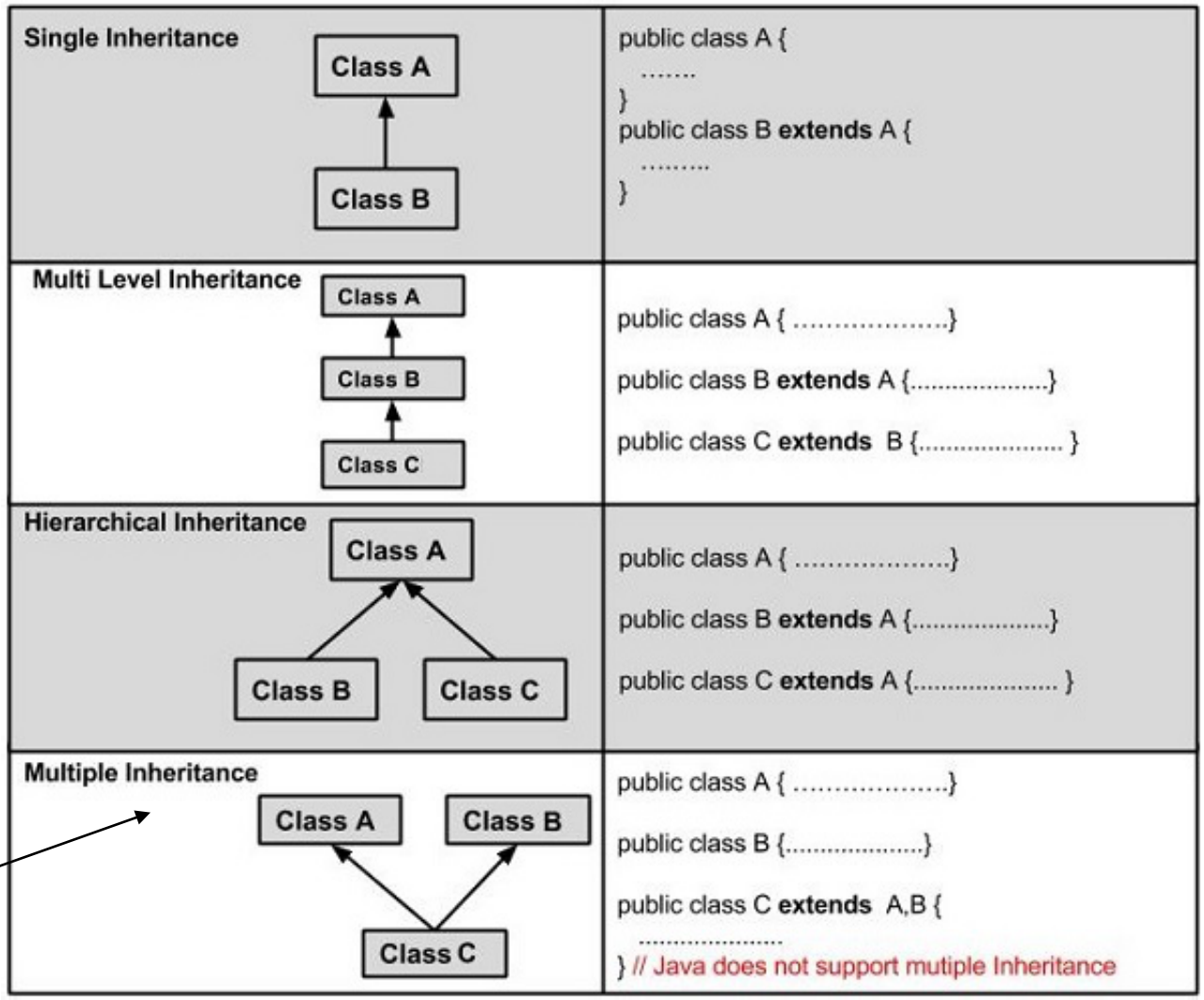



**"P.I.E“ triangle**

inheritance

polymorphism

# Why care about Interface?




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:

keyword **implements**

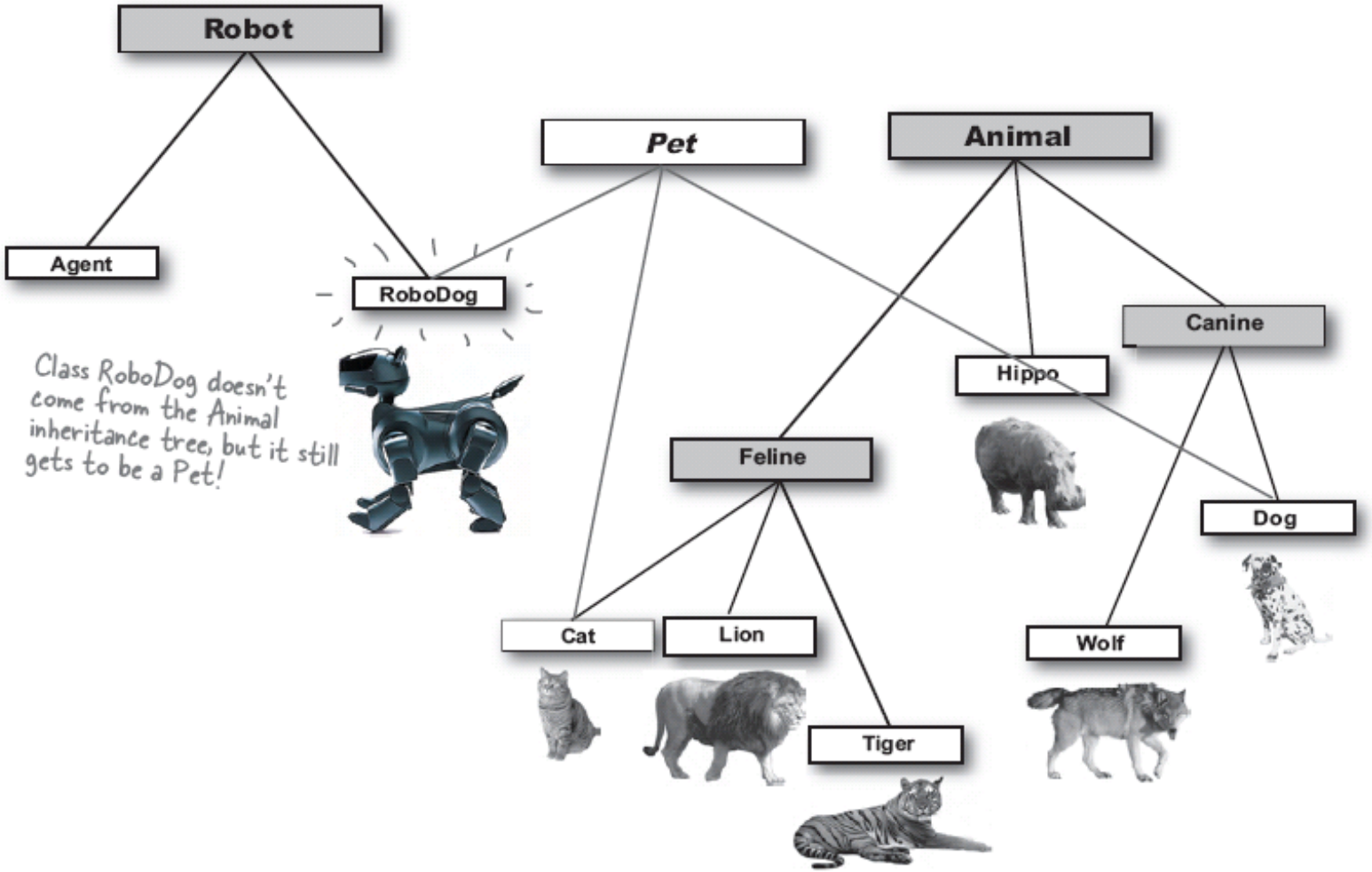
```
public class Mammal implements Animal {
   public void eat(){
       System.out.println("Mammal eats meat");
   }

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

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

implements ALL Animal methods

normal overriding methods

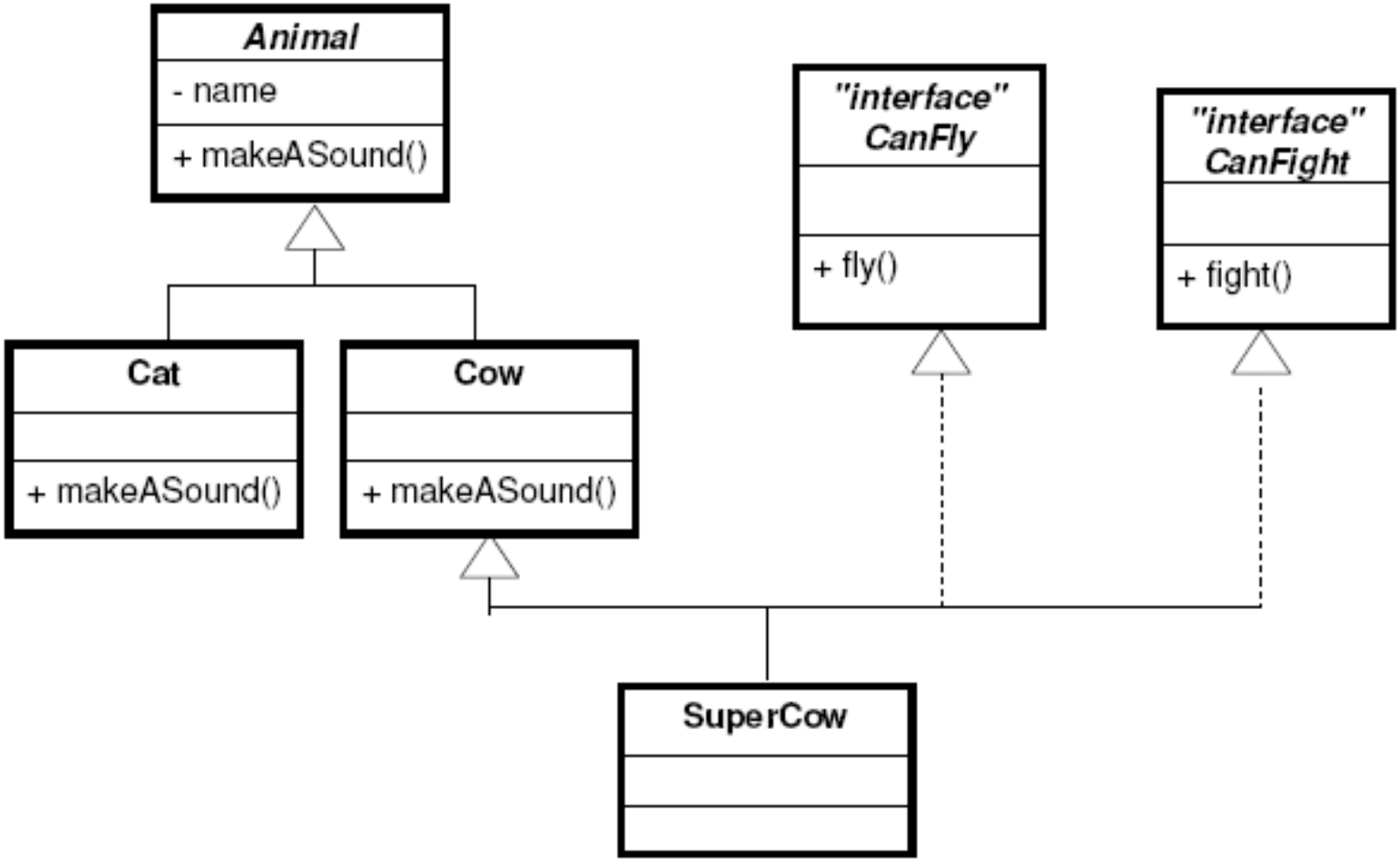
# Multiple Inheritance with Interface

- Classes from difference inheritance trees can implement the same interface



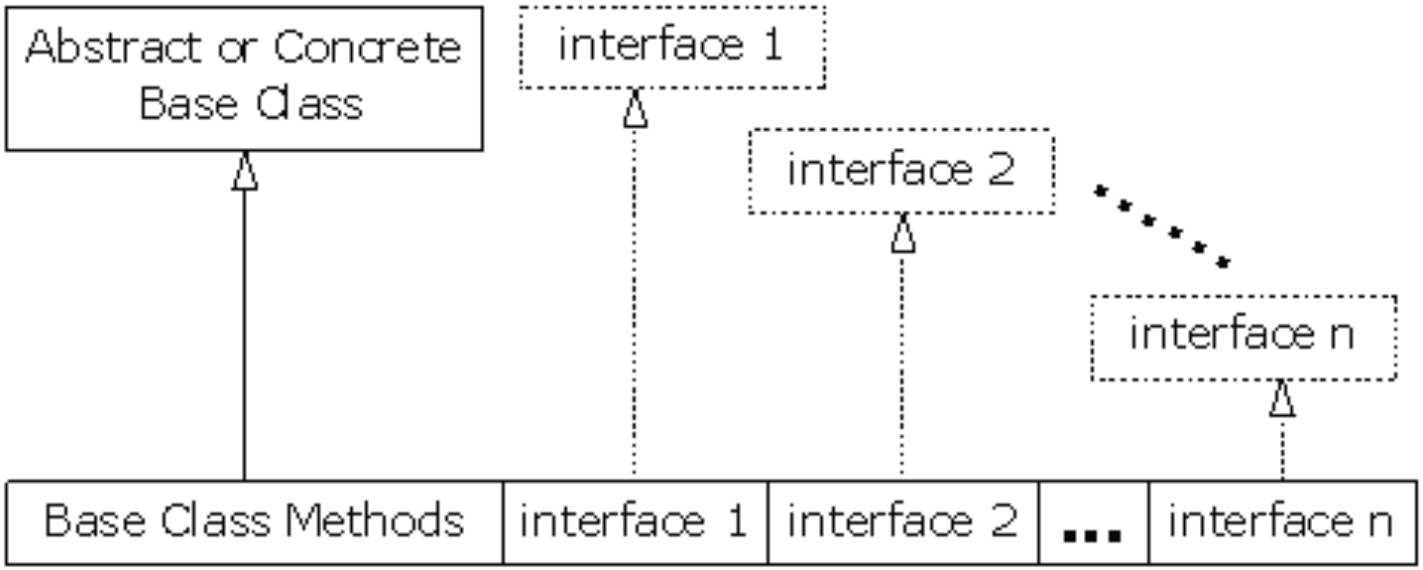
# Multiple Inheritance with Interface

- A class can implement multiple interfaces

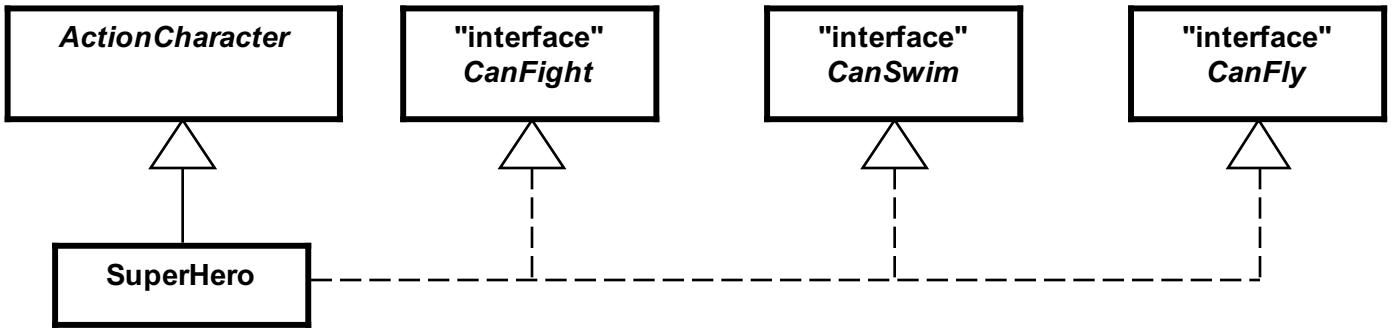


# Extends vs. Implements Keyword

- 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");
   }
}
```