# Task 7

### **Tính Kế Thừa và Đa Hình trong OOP**

### 1.Kế thừa (Inheritance):

Kế thừa trong lập trình hướng đối tượng cho phép một lớp (lớp con) kế thừa các thuộc tính và phương thức từ một lớp khác (lớp cha). Điều này giúp tái sử dụng mã và giảm thiểu sự trùng lặp.

**Từ khóa** extends: Trong Java, khi một lớp con kế thừa từ một lớp cha, ta sử dụng từ khóa extends.

class Animal {

String name;

public void eat() {

System.out.println("Animal is eating");

}

}

class Dog extends Animal {

public void bark() {

System.out.println("Dog is barking");

}

}

public class Main {

public static void main(String[] args) {

Dog dog = new Dog();

dog.name = "Buddy";

dog.eat(); // Kế thừa từ lớp Animal

dog.bark(); // Phương thức của lớp Dog

}

}

### **2.**Đa hình (Polymorphism):

Đa hình cho phép một đối tượng có thể hành xử theo nhiều cách khác nhau, tùy thuộc vào việc nó đang tham chiếu đến lớp nào.

**Ghi đè phương thức (Overriding)**: Lớp con có thể thay đổi cách hành xử của một phương thức được kế thừa từ lớp cha.

class Animal {

public void sound() {

System.out.println("Animal is making a sound");

}

}

class Dog extends Animal {

@Override

public void sound() {

System.out.println("Dog is barking");

}

}

class Cat extends Animal {

@Override

public void sound() {

System.out.println("Cat is meowing");

}

}

public class Main {

public static void main(String[] args) {

Animal animal1 = new Dog();

Animal animal2 = new Cat();

animal1.sound(); // In ra: Dog is barking

animal2.sound(); // In ra: Cat is meowing

}

}

### 2.Lớp trừu tượng và giao diện (Abstract class và Interface):

**Lớp trừu tượng (Abstract class)**: Là lớp không thể tạo đối tượng, nó chứa các phương thức có thể đã được triển khai hoặc chưa triển khai (phương thức trừu tượng). Các lớp con phải thực hiện các phương thức trừu tượng này.

**abstract class Animal {**

**public abstract void sound();**

**public void sleep() {**

**System.out.println("Animal is sleeping");**

**}**

**}**

**class Dog extends Animal {**

**@Override**

**public void sound() {**

**System.out.println("Dog is barking");**

**}**

**}**

**public class Main {**

**public static void main(String[] args) {**

**Animal animal = new Dog();**

**animal.sound();**

**animal.sleep();**

**}**

**}**