**Lớp và Đối Tượng trong Java**

Trong lập trình hướng đối tượng, **lớp** (class) và **đối tượng** (object) là các khái niệm cơ bản. Lớp là một bản thiết kế (blueprint) cho đối tượng, và đối tượng là thể hiện cụ thể của lớp.

**1. Khái Niệm về Lớp và Đối Tượng**

**Lớp (Class)**

Lớp là một cấu trúc dữ liệu tùy chỉnh, đại diện cho một tập hợp các thuộc tính (biến) và hành vi (phương thức). Lớp là bản thiết kế từ đó các đối tượng có thể được tạo ra.

**Cú pháp:**

class ClassName {

// Khai báo thuộc tính (biến instance)

int attribute;

// Khai báo phương thức

void methodName() {

// Nội dung phương thức

}

}

**Đối Tượng (Object)**

Đối tượng là thể hiện cụ thể của một lớp. Khi tạo một đối tượng từ lớp, Java sẽ cấp phát bộ nhớ để lưu trữ các thuộc tính của đối tượng đó.

**Tạo đối tượng:**

ClassName object = new ClassName();

**2. Cách Tạo và Sử Dụng Đối Tượng**

**Tạo Đối Tượng**

Để tạo một đối tượng, ta sử dụng từ khóa new kết hợp với tên lớp.

**Ví dụ:**

class Dog {

String name;

int age;

void bark() {

System.out.println("Woof! Woof!");

}

}

public class Main {

public static void main(String[] args) {

Dog myDog = new Dog(); // Tạo đối tượng từ lớp Dog

myDog.name = "Buddy"; // Gán giá trị cho thuộc tính

myDog.age = 3;

System.out.println(myDog.name + " is " + myDog.age + " years old.");

myDog.bark(); // Gọi phương thức

}

}

**Kết quả:**

Buddy is 3 years old.

Woof! Woof!

**3. Phương Thức Khởi Tạo (Constructor) và Nạp Chồng (Overloading)**

**Phương Thức Khởi Tạo (Constructor)**

Constructor là một phương thức đặc biệt được gọi khi một đối tượng được tạo ra. Constructor có cùng tên với lớp và không có kiểu trả về (ngay cả void).

**Cú pháp:**

class ClassName {

// Constructor mặc định

public ClassName() {

// Khởi tạo giá trị cho các thuộc tính

}

}

**Ví dụ:**

class Dog {

String name;

int age;

// Constructor

public Dog(String name, int age) {

this.name = name;

this.age = age;

}

void bark() {

System.out.println(name + " says Woof!");

}

}

public class Main {

public static void main(String[] args) {

Dog myDog = new Dog("Buddy", 3); // Gọi constructor

myDog.bark(); // Gọi phương thức

}

}

**Kết quả:**

Buddy says Woof!

**Nạp Chồng Constructor (Constructor Overloading)**

Trong Java, một lớp có thể có nhiều constructor với tham số khác nhau (nạp chồng constructor).

**Ví dụ:**

class Dog {

String name;

int age;

// Constructor không tham số

public Dog() {

this.name = "Unnamed";

this.age = 0;

}

// Constructor có tham số

public Dog(String name, int age) {

this.name = name;

this.age = age;

}

void bark() {

System.out.println(name + " says Woof!");

}

}

public class Main {

public static void main(String[] args) {

Dog unnamedDog = new Dog(); // Gọi constructor không tham số

Dog namedDog = new Dog("Buddy", 3); // Gọi constructor có tham số

unnamedDog.bark(); // Kết quả: "Unnamed says Woof!"

namedDog.bark(); // Kết quả: "Buddy says Woof!"

}

}

**4. Thuộc Tính và Phương Thức**

**Khai Báo Thuộc Tính (Biến Instance)**

Thuộc tính của một lớp (còn gọi là biến instance) là các biến được khai báo bên trong lớp nhưng bên ngoài bất kỳ phương thức nào. Mỗi đối tượng của lớp sẽ có một bản sao riêng của các thuộc tính này.

**Ví dụ:**

class Car {

String model;

int year;

}

**Khai Báo Phương Thức**

Phương thức là các hàm được định nghĩa bên trong lớp để mô tả hành vi của các đối tượng. Phương thức có thể nhận tham số và trả về giá trị.

**Cú pháp:**

class ClassName {

// Phương thức không trả về

void methodName() {

// Nội dung phương thức

}

// Phương thức có trả về

int sum(int a, int b) {

return a + b;

}

}

**Gọi Phương Thức Từ Đối Tượng**

Khi tạo đối tượng từ một lớp, ta có thể gọi các phương thức bằng cách sử dụng toán tử chấm (.).

**Ví dụ:**

class Calculator {

int add(int a, int b) {

return a + b;

}

}

public class Main {

public static void main(String[] args) {

Calculator calc = new Calculator();

int result = calc.add(5, 3); // Gọi phương thức add

System.out.println("Result: " + result); // Kết quả: 8

}

}

**Tổng Kết**

* **Lớp (Class)**: Là bản thiết kế cho các đối tượng, chứa thuộc tính và phương thức.
* **Đối tượng (Object)**: Là thể hiện cụ thể của lớp.
* **Constructor**: Là phương thức đặc biệt được gọi khi đối tượng được tạo, và có thể được nạp chồng để hỗ trợ nhiều cách khởi tạo khác nhau.
* **Thuộc tính (biến instance)**: Là các biến lưu trữ trạng thái của đối tượng.
* **Phương thức**: Là các hàm mô tả hành vi của đối tượng, được gọi từ đối tượng sau khi nó được tạo.

Trên đây là khái niệm và cách sử dụng lớp, đối tượng, constructor, thuộc tính và phương thức trong Java.