**1. Lập trình hướng đối tượng (OOP)**

Là một mô hình lập trình tổ chức phần mềm thành các đối tượng, mỗi đối tượng bao gồm thuộc tính và phương thức. Lập trình OOP trong Java giúp mã nguồn dễ quản lý, tái sử dụng, bảo trì và mở rộng.

**2. Sự khác nhau giữa lập trình hướng thủ tục và lập trình hướng đối tượng :**

- Lập trình hướng thủ tục : Chương trình chia thành các hàm (phương thức), tập trung vào cách thực hiện nhiệm vụ.

- Lập trình hướng đối tượng : Chương trình tổ chức thành các đối tượng, tập trung vào mô hình hóa thực thể, trạng thái và hành vi của chúng.

**3. Sự khác nhau giữa lớp và đối tượng :**

- Lớp (class) : Là bản thiết kế, mô hình hoặc khuôn mẫu xác định các thuộc tính và hành vi chung.

- Đối tượng (object) : Là một thực thể cụ thể được tạo ra từ lớp, đại diện cho một phiên bản của lớp.

**4. Ý nghĩa của phương thức, thuộc tính và phương thức khởi tạo :**

- Thuộc tính (attributes/fields) : Là các biến lưu trữ trạng thái của đối tượng.

- Phương thức (methods) : Là các hàm định nghĩa hành vi mà đối tượng có thể thực hiện.

- Phương thức khởi tạo (constructor) : Là phương thức đặc biệt được gọi khi đối tượng được tạo, dùng để khởi tạo thuộc tính ban đầu.

**5. Cú pháp khai báo lớp (trong Java):**

public class TenLop {

// Thuộc tính

private String thuocTinh1;

private int thuocTinh2;

// Phương thức khởi tạo

public TenLop(String thuocTinh1, int thuocTinh2) {

this.thuocTinh1 = thuocTinh1;

this.thuocTinh2 = thuocTinh2;

}

// Phương thức

public void hienThi() {

System.out.println("Thuộc tính 1: " + this.thuocTinh1);

System.out.println("Thuộc tính 2: " + this.thuocTinh2);

}

}

**6. Cú pháp khởi tạo đối tượng :**

TenLop tenDoiTuong = new TenLop("Giá trị 1", 123);

**7. Cách truy xuất thuộc tính và phương thức của lớp :**

- Truy xuất thuộc tính : Trong Java, thường dùng getter để truy cập các thuộc tính riêng tư (private):

String giaTri = tenDoiTuong.getThuocTinh1();

- Gọi phương thức :

tenDoiTuong.hienThi();

**8. Class diagram :**

Là một sơ đồ UML dùng để biểu diễn các lớp, thuộc tính, phương thức và mối quan hệ giữa chúng. Class diagram giúp lập trình viên hiểu rõ cấu trúc hệ thống, mối quan hệ giữa các lớp, kế thừa và phụ thuộc trong thiết kế phần mềm.

Công dụng chính:

- Thiết kế hệ thống : Giúp hình dung tổng quan cấu trúc của ứng dụng.

- Giao tiếp : Giúp các thành viên trong nhóm phát triển phần mềm hiểu rõ cách tổ chức và tương tác giữa các thành phần.

- Bảo trì và mở rộng : Giúp dễ dàng theo dõi và bảo trì mã nguồn, dễ dàng thêm hoặc chỉnh sửa các lớp và phương thức.