

# Thực hành: Hệ Thống Quản Lý Nhân Sự (HRM)

## I. Cấu Trúc Dự Án và Định Nghĩa Cơ Bản (Datatype, Package, Enum)

### 1. Cấu trúc Package:

- `com.hrm.model`: Chứa các lớp đối tượng, Enum, Interface.
- `com.hrm.service`: Chứa logic nghiệp vụ.
- `com.hrm.app`: Chứa lớp `main()` và logic điều khiển giao diện.

### 2. Enum: Tạo Enum `PhongBan` (trong `com.hrm.model`) với các giá trị: `IT`, `MARKETING`, `HCNG` (Hành chính nhân sự).

---

## II. Triển Khai OOP (Encapsulation, Abstraction, Inheritance, Polymorphism)

### 1. Interface `NhanSu` (Abstraction - Part 1)

- Tạo Interface `NhanSu` (trong `com.hrm.model`) để định nghĩa hai phương thức **trừu tượng**:
  - `double tinhLuongThucLanh();`
  - `String layTenChucVu();`

### 2. Lớp Abstract `NhanVien` (Encapsulation, Constructor, Object, This/Super)

- Tạo lớp Abstract `NhanVien` (trong `com.hrm.model`), **implement** Interface `NhanSu`.
- **Thuộc tính**: Sử dụng thuộc tính: `String maSo`, `String hoTen`, `LocalDate ngayVaoLam`, `PhongBan phongBan`, `double heSoLuong`.
- **Print Object**: **Override** phương thức `toString()` của **class Object** để in thông tin cơ bản của nhân viên có định dạng.

### 3. Các Lớp Con (Inheritance, Polymorphism, Override)

#### a) Lớp `LapTrinhVien` (Kế thừa từ `NhanVien`)

- Thuộc tính bổ sung: `int soGioOvertime`.
- Constructor: Sử dụng **super keyword**.
- Override  `tinhLuongThucLanh()`:  $Lương = (Hệ\ số\ lương \times 15,000,000) + (sốGioOvertime \times 50,000)$ .
- Override  `layTenChucVu()`: Trả về "Lập trình viên".

#### b) Lớp `QuanLyDuAn` (Kế thừa từ `NhanVien`)

- Thuộc tính bổ sung: `double luongTrachNhiem, int soDuAnDaHoanThanh`.
- Constructor: Sử dụng **super keyword**.
- Override  `tinhLuongThucLanh()`:  $Lương = (Hệ\ số\ lương \times 15,000,000) + luongTrachNhiem + (sốDuAnDaHoanThanh \times 1,000,000)$ .
- Override  `layTenChucVu()`: Trả về "Quản lý Dự án".

## III. Kiến thức cơ bản

### 1. Lớp `HRMService` (com.hrm.service)

- Lưu trữ: Dùng `NhanVien[]` (Array) để lưu trữ hồ sơ.
- Các Method:
  - `void themNhanVien(NhanVien nv)`: Thêm nhân viên.
  - `NhanVien timTheoMa(String ma)`: Tìm kiếm nhân viên. Sử dụng **for** và **break** khi tìm thấy.
  - `void hienThiTatCa()`: Hiển thị danh sách chi tiết (dùng **foreach**).
  - `double tinhLuongTrungBinh()`: Tính lương trung bình.

### 2. Lớp `Application` (com.hrm.app)

- Demo các hàm đã viết bên trên bằng cách viết logic sử dụng trong main

## IV. Vận Dụng Đa Kiến Thức

### 1. Operator & Flow Control:

- **Tăng lương thâm niên**: Viết một method kiểm tra nếu thâm niên trên 5 năm (dùng **Relational Operator**), **tăng hệ số lương lên 1.1 lần**. Dùng **if-else**.
- Sử dụng **Ternary Operator** để in chức vụ nhanh.

### 2. String Method:

- Khi tìm kiếm, sử dụng **String** methods như `trim()`, `toUpperCase()`, và `equalsIgnoreCase()` để xử lý input.

```
HRM_Project/
├── src/
│   ├── com/
│   │   ├── hrm/
│   │   │   ├── app/
│   │   │   │   └── Application.java (Chứa main, Scanner, Menu)
│   │   │   ├── model/
│   │   │   │   ├── Enum/
│   │   │   │   │   └── PhongBan.java
│   │   │   │   ├── Interface/
│   │   │   │   │   └── NhanSu.java
│   │   │   │   ├── Abstract/
│   │   │   │   │   └── NhanVien.java
│   │   │   │   ├── LapTrinhVien.java
│   │   │   │   └── QuanLyDuAn.java
│   │   │   └── service/
│   │   │       └── HRMService.java (Chứa List, Logic)
```