

Bài 1: Hệ thống đăng ký khóa học (Tổng hợp OOP)

Yêu cầu:

- Lớp `Person` là cha của `Student` và `Instructor`.
- Lớp `Course` có các thuộc tính: `courseId`, `courseName`, `instructor`, `students` (danh sách học viên).
- `Student` có thể đăng ký vào nhiều `Course`.
- Viết chương trình mô phỏng việc thêm/sửa/xóa khóa học và in danh sách học viên.

👉 Mục tiêu: Thực hành mối quan hệ 1-n, OOP đầy đủ.

Bài 2: Trò chơi chiến binh (Đa hình, Interface)

Yêu cầu:

- Tạo interface `Fighter` với phương thức `attack()`.
- Các lớp `Knight`, `Archer`, `Mage` implement `Fighter`, mỗi lớp có kiểu tấn công riêng.
- Tạo lớp `BattleField` với danh sách các chiến binh, cho mỗi chiến binh tấn công theo lượt.
- Thêm điểm năng lượng cho từng chiến binh và xử lý giảm theo mỗi đòn đánh.

👉 Mục tiêu: Sử dụng interface, đa hình động.

Bài 3: Trình quản lý file (Trừu tượng + Kế thừa + Interface nâng cao)

Yêu cầu:

- Tạo lớp trừu tượng `FileHandler` với các phương thức: `read()`, `write()`, `delete()`.
- Tạo các lớp con: `TextFileHandler`, `ImageFileHandler`, `VideoFileHandler`.

- Tạo interface **Compressible** cho các file có thể nén, và **Encryptable** cho các file cần mã hóa.
- Viết chương trình cho phép người dùng thao tác với file qua giao diện dòng lệnh.

👉 Mục tiêu: Kết hợp *abstract class* + *interface* + *OOP* nâng cao.

Homework

Tổng quan đề bài:

Thiết kế một hệ thống trò chơi **Plant vs Zombie dạng mô phỏng console** trong Java, sử dụng các đặc tính OOP (kế thừa, đa hình, đóng gói, trừu tượng, interface).

🧩 Yêu cầu 1: Tạo lớp trừu tượng **GameObject**

Mục tiêu: Thiết kế lớp cha **GameObject** chứa các thuộc tính và phương thức chung cho tất cả đối tượng trong game (cây và zombie).

✅ Đề bài:

- Tạo lớp **trừu tượng** **GameObject**.
 - Thuộc tính chung: **health**, **x**, **y**.
 - Phương thức trừu tượng: **update()** (được override bởi tất cả lớp con).
 - Phương thức: **takeDamage(int amount)** và **isAlive()**.
-

🧩 Yêu cầu 2: Tạo Interface để mô tả hành vi đặc biệt

Mục tiêu: Áp dụng interface để mô tả các hành vi như bắn, di chuyển, v.v.

✅ Đề bài:

- Tạo interface `Shooter` với phương thức `shoot(Zombie target)`.
 - Tạo interface `Mover` với phương thức `move()`.
-

Yêu cầu 3: Tạo lớp cây bắn đậu PeaShooter

Mục tiêu: Kế thừa từ `GameObject`, implement interface `Shooter`.

Đề bài:

- Tạo lớp `PeaShooter`:
 - Kế thừa từ `GameObject`.
 - Triển khai interface `Shooter`.
 - Có phương thức `shoot(Zombie target)` gây sát thương 20 HP.
 - Ghi đè phương thức `update()` để in trạng thái.
-

Yêu cầu 4: Tạo lớp Zombie có khả năng di chuyển

Mục tiêu: Kế thừa từ `GameObject`, implement `Mover`.

Đề bài:

- Tạo lớp `Zombie`:
 - Kế thừa từ `GameObject`.
 - Triển khai interface `Mover`.
 - Di chuyển từ phải qua trái bằng cách giảm `x`.
 - Ghi đè phương thức `update()` để di chuyển và in vị trí.
-

🧩 Yêu cầu 5: Viết game đơn giản mô phỏng trận chiến

Mục tiêu: Sử dụng OOP để mô phỏng trận chiến giữa cây và zombie.

✅ Đề bài:

- Tạo lớp `Game` với phương thức `main`.
 - Tạo một `PeaShooter` tại vị trí (5,1).
 - Tạo một `Zombie` tại vị trí (10,1).
 - Mỗi vòng lặp:
 - Zombie di chuyển về phía cây.
 - Nếu zombie đến gần trong phạm vi 5 ô, cây sẽ bắn.
 - In ra thông tin trận chiến.
 - Kết thúc khi zombie chết hoặc đi quá cây.
-

🧩 (Tuỳ chọn nâng cao): Yêu cầu 6: Mở rộng thêm các loại cây và zombie

Đề bài nâng cao nếu bạn muốn thử sức thêm:

- Thêm lớp `Sunflower` tạo ra năng lượng.
- Thêm lớp `WallNut` có nhiều máu, không bắn.
- Thêm `FastZombie` di chuyển 2 bước/lượt, máu yếu hơn.