**Bài tập 1: Hệ thống quản lý cửa hàng điện tử**

* **Mục tiêu**: Tìm hiểu về kế thừa và đa hình trong hệ thống quản lý sản phẩm.
* **Yêu cầu**:
  + Tạo lớp Product có các thuộc tính cơ bản như id, name, price, và quantity.
  + Tạo các lớp con như Laptop, Smartphone, và Tablet, kế thừa từ Product. Mỗi lớp có thêm các thuộc tính riêng biệt, ví dụ: Laptop có processor và ram; Smartphone có camera\_quality.
  + Thêm phương thức calculate\_discounted\_price(discount\_rate) trong lớp Product để tính giá sau khi giảm giá. Phương thức này sẽ được ghi đè ở các lớp con để tính theo cách riêng của từng loại sản phẩm.
  + Tạo một danh sách các sản phẩm trong cửa hàng, sau đó hiển thị thông tin chi tiết và giá đã giảm cho từng loại sản phẩm.

**Bài tập 2: Quản lý đội bóng và chiến thuật**

* **Mục tiêu**: Sử dụng các lớp con và phương thức trừu tượng.
* **Yêu cầu**:
  + Tạo lớp Player có các thuộc tính name, position, skill\_level, và phương thức play() (trừu tượng, chưa định nghĩa).
  + Tạo các lớp con như Forward, Midfielder, và Defender kế thừa từ Player, mỗi lớp sẽ định nghĩa lại phương thức play() để mô tả chiến thuật chơi riêng của từng vị trí.
  + Tạo lớp Team để quản lý danh sách các cầu thủ, thêm phương thức display\_team() để hiển thị danh sách cầu thủ theo từng vị trí, và total\_skill\_level() để tính tổng điểm kỹ năng của đội.
  + Tạo một đội bóng với các cầu thủ ở từng vị trí khác nhau và hiển thị đội hình cùng tổng điểm kỹ năng của đội.

**Bài tập 3: Hệ thống đặt phòng khách sạn**

* **Mục tiêu**: Thực hành các khái niệm quản lý trạng thái và sử dụng các lớp lồng nhau.
* **Yêu cầu**:
  + Tạo lớp HotelRoom có các thuộc tính room\_number, capacity, price\_per\_night, và status (trạng thái: “trống” hoặc “đã đặt”).
  + Thêm phương thức book\_room() để đặt phòng (đổi trạng thái thành “đã đặt”) và checkout() để trả phòng.
  + Tạo lớp Hotel chứa danh sách các phòng. Thêm phương thức search\_room(capacity) để tìm phòng trống theo sức chứa yêu cầu và phương thức calculate\_revenue() để tính tổng thu nhập của khách sạn từ các phòng đã đặt.
  + Tạo một khách sạn với danh sách các phòng, thử đặt phòng và trả phòng, rồi tính toán doanh thu.

**Bài tập 4: Hệ thống ngân hàng với các loại tài khoản khác nhau**

* **Mục tiêu**: Áp dụng kế thừa và đa hình để quản lý tài khoản ngân hàng.
* **Yêu cầu**:
  + Tạo lớp BankAccount với các thuộc tính account\_number, balance, và phương thức withdraw(amount) và deposit(amount).
  + Tạo các lớp con CheckingAccount và SavingsAccount kế thừa từ BankAccount. Mỗi lớp có quy tắc riêng cho rút tiền. Ví dụ: CheckingAccount có thể cho phép số dư âm (với mức phí) trong khi SavingsAccount chỉ cho phép rút nếu đủ số dư.
  + Tạo phương thức display\_balance() để hiển thị số dư. Ghi đè phương thức này trong từng lớp con để hiển thị thêm các chi tiết như số tiền có thể rút trong tài khoản tiết kiệm.
  + Tạo một vài tài khoản, thử rút tiền và nạp tiền vào các tài khoản này, sau đó hiển thị số dư.

**Bài tập 5: Hệ thống đặt vé máy bay với các loại ghế**

* **Mục tiêu**: Áp dụng kế thừa, đa hình và các quy tắc đặc biệt cho từng loại ghế.
* **Yêu cầu**:
  + Tạo lớp Seat với các thuộc tính seat\_number, price, và status (trạng thái: “trống” hoặc “đã đặt”).
  + Tạo các lớp con EconomySeat, BusinessSeat, và FirstClassSeat kế thừa từ Seat. Mỗi lớp có phương thức calculate\_final\_price() để tính giá cuối cùng với các phụ phí riêng, ví dụ: BusinessSeat có phụ phí dịch vụ và FirstClassSeat có phụ phí cao hơn.
  + Tạo lớp Flight chứa danh sách các ghế và phương thức book\_seat(seat\_number) để đặt ghế theo số ghế và calculate\_total\_revenue() để tính tổng doanh thu của chuyến bay.
  + Tạo một chuyến bay với danh sách các loại ghế khác nhau, thử đặt ghế và tính doanh thu cuối cùng từ các ghế đã đặt.