**Bài tập trên lớp**

**Ex1**

Xây Dựng Hệ Thống Quản Lý Học Tập

1. Tạo lớp Person

* FirstName: họ của người
* LastName: tên của người
* Age: tuổi của người

Tạo phương thức DisplayInfo() để hiển thị thông tin của người, bao gồm họ, tên và tuổi.

1. Lớp Student kế thừa từ lớp Person và có thêm các thuộc tính

* StudentID: mã số sinh viên
* Courses: danh sách các khóa học mà sinh viên đã đăng ký (dạng List<string>)

Tạo phương thức EnrollCourse(string course) để thêm một khóa học vào danh sách khóa học của sinh viên.

Tạo phương thức DisplayStudentInfo() để hiển thị thông tin của sinh viên, bao gồm thông tin cá nhân và các khóa học đã đăng ký.

1. Lớp Teacher kế thừa từ lớp Person và có thêm các thuộc tính

* TeacherID: mã số giáo viên
* TeachingCourses: danh sách các khóa học mà giáo viên đang dạy (dạng List<string>)

Tạo phương thức AddCourse(string course) để thêm một khóa học vào danh sách khóa học mà giáo viên đang dạy.

Tạo phương thức DisplayTeacherInfo() để hiển thị thông tin của giáo viên, bao gồm thông tin cá nhân và các khóa học đang dạy.

1. Lớp Course để quản lý thông tin về các khóa học và có các thuộc tính

* CourseName: tên khóa học
* CourseTeacher: giáo viên phụ trách khóa học (dạng Teacher)
* EnrolledStudents: danh sách các sinh viên đã đăng ký khóa học (dạng List<Student>)

Tạo phương thức EnrollStudent(Student student) để thêm sinh viên vào khóa học và cập nhật danh sách khóa học của sinh viên.

Tạo phương thức DisplayCourseInfo() để hiển thị thông tin của khóa học, bao gồm tên khóa học, thông tin giáo viên phụ trách và danh sách sinh viên đã đăng ký.

1. Chương Trình Chính:

Tạo một đối tượng giáo viên và một số đối tượng sinh viên.

Tạo một đối tượng khóa học và gán giáo viên phụ trách cho khóa học đó.

Đăng ký các sinh viên vào khóa học.

Hiển thị thông tin của khóa học bao gồm thông tin giáo viên và danh sách sinh viên đã đăng ký.

**Bài tập về nhà**

Mục tiêu: Xây dựng một hệ thống quản lý eCommerce sử dụng lập trình hướng đối tượng

1. Tạo lớp Product với các thuộc tính và phương thức cơ bản:

* Name: Tên sản phẩm (kiểu string).
* BasePrice: Giá cơ bản của sản phẩm (kiểu decimal).

Định nghĩa một phương thức virtual để tính giá trị sản phẩm CalculatePrice (kiểu decimal), mặc định trả về giá cơ bản của sản phẩm.

Tạo phương thức DisplayInfo để hiển thị thông tin sản phẩm bao gồm tên và giá.

1. Lớp DigitalProduct kế thừa từ lớp Product và có thêm các thuộc tính

* FileSize: Kích thước file của sản phẩm số (kiểu double, đơn vị MB).

Ghi đè phương thức CalculatePrice để tính giá trị sản phẩm số. Giá trị sản phẩm số sẽ được tính bằng giá cơ bản cộng với 0.1 đơn vị tiền cho mỗi MB của kích thước file.

1. Lớp PhysicalProduct kế thừa từ lớp Product và có thêm các thuộc tính

* Weight: Trọng lượng của sản phẩm vật lý (kiểu double, đơn vị kg).
* ShippingCost: Phí vận chuyển của sản phẩm vật lý (kiểu decimal).

Ghi đè phương thức CalculatePrice để tính giá trị sản phẩm vật lý. Giá trị sản phẩm vật lý sẽ được tính bằng giá cơ bản cộng với phí vận chuyển.

1. Tạo lớp Order với các thuộc tính và phương thức để quản lý danh sách sản phẩm và tính tổng giá trị đơn hàng:

* Products: Danh sách các sản phẩm trong đơn hàng (kiểu List<Product>).

Tạo phương thức AddProduct để thêm sản phẩm vào đơn hàng.

Tạo phương thức CalculateTotalPrice để tính tổng giá trị đơn hàng dựa trên giá của từng sản phẩm trong đơn hàng.

Tạo phương thức DisplayOrderDetails để hiển thị thông tin chi tiết đơn hàng, bao gồm thông tin của từng sản phẩm và tổng giá trị đơn hàng.

1. Tạo lớp trừu tượng Payment với thuộc tính và phương thức cơ bản:

* Amount: Số tiền thanh toán (kiểu decimal).

Định nghĩa phương thức trừu tượng ProcessPayment để xử lý thanh toán.

1. Lớp CreditCardPayment và PayPalPayment kế thừa từ lớp Payment.

+ CreditCardPayment:

Thêm thuộc tính CreditCardNumber: Số thẻ tín dụng (kiểu string).

Ghi đè phương thức ProcessPayment để xử lý thanh toán qua thẻ tín dụng.

+ PayPalPayment:

Thêm thuộc tính PayPalAccount: Tài khoản PayPal (kiểu string).

Ghi đè phương thức ProcessPayment để xử lý thanh toán qua PayPal.