

1. Trình bày ưu và khuyết điểm của lập trình cấu trúc và lập trình hướng đối tượng ?

	Lập trình cấu trúc	Lập trình hướng đối tượng
Ưu điểm	<ul style="list-style-type: none"> - Cấu trúc rõ ràng, đơn giản, dễ hiểu - Dễ bảo trì và quản lý 	<ul style="list-style-type: none"> - Che dấu dữ liệu (đóng gói) - Khả năng tái sử dụng mã nguồn cao (kế thừa) - Chương trình dễ dàng mở rộng thông qua cơ chế kế thừa (không tác động đến chương trình khi cài thêm plugins)
Khuyết điểm	<ul style="list-style-type: none"> - Khó quản lý sự thay đổi dữ liệu - Chương trình khó mở rộng - Khả năng tái sử dụng mã nguồn hạn chế 	<ul style="list-style-type: none"> - Phức tạp - Khó theo dõi luồng dữ liệu - Không thích hợp các hệ thống nhỏ

2. Phân biệt lớp trừu tượng (abstract class) và giao diện (interface) ?

Abstract class	Interface
<ul style="list-style-type: none"> - Abstract class có phương thức abstract (không có thân hàm) và phương thức non-abstract (có thân hàm). - Abstract class không hỗ trợ đa kế thừa. - Abstract class có các biến final, non-final, static and non-static. - Abstract class có thể cung cấp nội dung cài đặt cho phương thức của interface. - Từ khóa abstract được sử dụng để khai báo abstract class. 	<ul style="list-style-type: none"> - Interface chỉ có phương thức abstract. Từ java 8, nó có thêm các phương thức default và static. - Interface có hỗ trợ đa kế thừa - Interface chỉ có các biến static và final. - Interface không thể cung cấp nội dung cài đặt cho phương thức của abstract class. - Từ khóa interface được sử dụng để khai báo interface.

3. Các thành viên tĩnh sử dụng làm gì ? Nó hoạt động thế nào trong lớp và trong quan hệ kế thừa ?

- Thuộc tính tĩnh và phương thức tĩnh dùng chung cho tất cả các đối tượng lớp
- Chúng được sử dụng thông qua tên lớp mà không cần tạo đối tượng
- Trong phương thức tĩnh chỉ truy xuất được các thành viên tĩnh của lớp
- Phương thức tĩnh được phép kế thừa ở lớp con nhưng không được ghi đè

4. Mặc định một lớp kế thừa lớp nào ? Trình bày vài phương thức quan trọng lớp đó ?

- Mặc định kế thừa lớp Object
- Một vài phương thức quan trọng:
 - + equals(Object obj): kiểm tra 2 đối tượng có bằng nhau
 - + toString(): trả về chuỗi đại diện cho đối tượng
 - + getClass(): trả về lớp mà đối tượng tạo ra
 - + clone(): sao chép đối tượng ra đối tượng mới
 - + hashCode(): trả về mã băm của đối tượng

5. Đa hình là gì ? Nó được thể hiện thông qua cơ chế nào ?

- Đa hình là khả năng của 1 đối tượng có thể thực hiện 1 tác vụ theo nhiều cách khác nhau
- Đa hình được thể hiện thông qua cơ chế nạp chồng (overloading) và ghi đè (overriding)

6. Sự khác nhau giữa Association, Dependency, Composition và Aggregation trong OOP ?

Association	Dependency	Composition	Aggregation
- Lớp A có thuộc tính kiểu lớp B hoặc lớp B có thuộc tính kiểu lớp A	<ul style="list-style-type: none">- Lớp A và lớp B không có quan hệ Association- Đối tượng kiểu lớp B có thể là đối số hoặc kết quả trả về hoặc biến cục bộ trong các phương thức của lớp A	<ul style="list-style-type: none">- Lớp A và lớp B đã có quan hệ Association- Lớp A có thuộc tính kiểu lớp B, nếu đối tượng a của lớp A bị huỷ thì đối tượng b (thuộc tính của đối tượng a) của lớp B không thể tồn tại	<ul style="list-style-type: none">- Lớp A và lớp B đã có quan hệ Association- Lớp A có thuộc tính kiểu lớp B, nếu đối tượng a của lớp A bị huỷ thì đối tượng b (thuộc tính của đối tượng a) của lớp B vẫn có thể tồn tại

7. Các đặc điểm quan trọng của lập trình hướng đối tượng ?

- Tính đóng gói (encapsulation): cho phép che dấu thông tin và những tính chất xử lý bên trong của đối tượng
- Tính kế thừa (inheritance): cho phép xây dựng lớp con, kế thừa các phương thức, thuộc tính của lớp cha
- Tính đa hình (polymorphism): cho phép các đối tượng khác nhau thực thi chức năng giống nhau theo những cách khác nhau
- Tính trừu tượng (abstraction): loại bỏ những thứ phức tạp, không cần thiết của đối tượng và chỉ quan tâm vào những thứ cốt lõi, quan trọng

8. Phân biệt nạp chồng (overloading) và ghi đè (overriding) ?

Overloading	Overriding
<ul style="list-style-type: none"> - Thể hiện đa hình trong lúc compile time - Thêm hành vi cho phương thức - Khác nhau về số lượng và kiểu dữ liệu của tham số - Xảy ra trong cùng 1 class 	<ul style="list-style-type: none"> - Thể hiện đa hình trong lúc runtime - Thay đổi hành vi hiện tại của phương thức - Số lượng và kiểu dữ liệu của tham số phải giống nhau - Xảy ra ở 2 class có quan hệ kế thừa

9. Trình bày cơ chế hoạt động của phương thức khởi tạo trong quan hệ kế thừa ?

- Phương thức khởi tạo không được kế thừa ở lớp con
- Thứ tự các phương thức khởi tạo được gọi là từ các lớp cha trước rồi mới đến lớp con
- Các phương thức khởi tạo lớp con phải gọi phương thức khởi tạo lớp cha, nếu không Java sẽ ngầm định gọi phương thức khởi tạo không tham số của lớp cha

10. Phân biệt từ khóa this và super ?

This	Super
<ul style="list-style-type: none"> - Tham chiếu tới biến instance của lớp hiện tại - Gọi phương thức (method) của lớp hiện tại - Gọi phương thức khởi tạo (constructor) của lớp hiện tại - Trả về instance của lớp hiện tại - Được truyền như một tham số trong phương thức (method) - Được truyền như một tham số trong phương thức khởi tạo (constructor) 	<ul style="list-style-type: none"> - Gọi trực tiếp phương thức khởi tạo (constructor) của lớp cha gần nhất (đặt đầu tiên khi gọi) - Gọi trực tiếp thuộc tính (field) của lớp cha gần nhất - Gọi trực tiếp phương thức (method) của lớp cha gần nhất