|  |  |
| --- | --- |
| **1. Kế thừa là gì ?** | Lớp con có thể kế thừa lại những đặc điểm và hành vi của lớp cha mà không cần phải khai báo lại . Và lớp con có thể mở rộng thêm thuộc tính và hành vi riêng. |
| **2. Đặc điểm kế thừa trong java ?** | + Superclass lớp cha , subclass lớp con (is-a)  + Extends để triển khai kế thừa  + Contructor không thể kế thừa 🡺 phải dùng super để gọi  + Thuộc tính và hành vi khai báo bằng private không được kế thừa  + Không hỗ trợ đa kế thừa |
| **3. Vì sao java không hỗ trợ đa kế thừa ?** | -Để tránh xung đột khi 2 lớp cha có phương thức trùng tên dễ gây mơ hồ. -Nếu cần đa kế thừa, có thể sử dụng interface 🡺 Một class không thể kế thừa từ nhiều class khác nhưng có thể implements nhiều interface khác nhau một lúc  -Giúp đơn giản hóa bộ máy xử lý của JVM. |
| **4. Từ khóa super , final dùng để làm gì ?** | -Từ khóa Super: + Dùng để gọi constructor hoặc method hoặc field từ lớp cha.  - Từ Khóa Final:  + Được sử dụng để ngăn chặn việc kế thừa từ một lớp và việc ghi đè phương thức |
| **5. Class Object là gì ? một số method trong Object ?** | Class object:là lớp gốc của tất cả các lớp trong java,tất cả các lớp trong java điều được thừa kế tử lớp object Các method cơ bản của class object -toString() : trả về một chuỗi đại diện của object - Equals(object obj) : so sánh hai object có bằng nhau không - HashCode() : trả về mã hastcode của object - GetClass() : trả về đối tượng class biểu diễn lớp thực thi đối tượng |
| **6. Tính đa hình là gì ? được thể hiện ở đâu trong java ?** | - Là cơ chế cho phép một biến thuộc kiểu dữ liệu cha có thể trỏ đến một đối tượng thuộc lớp con - Tính đa hình được thể hiện rõ khi - Khi class con kế thừa class cha thì biến thuộc kiểu dữ liệu của cha sẽ được tham chiếu đến các class con,có thể dùng một class cha để tạo ra các class con khác nhau  - Thể hiện đa hình qua interface - Thể hiện qua overloading |
| **7. Overriding là gì ? Điều kiện để có Overriding ?** | -Ghi đè phương thức là hình thức lớp con viết lại các phương thức đã có của lớp cha -Điều kiện: + Cùng tên , tham số , kiểu dữ liệu trả về , accseemodifier của lớp cha |
| **8. Phân biệt overloading và overriding ?** | Overloading :  + Cùng tên phương thức , khác tham số (khác kiểu , số lượng , thứ tự)  + Trong cùng một class hoặc lớp con  + Tăng tính linh hoạt cho method  Overriding :  + Cùng tên, cùng tham số, định nghĩa lại ở lớp con  + Giữa lớp cha và lớp con  + Thay đổi , hoàn thiện hành vi của method |
| **9. Ép kiểu là gì ? vì sao phải ép kiểu ?** | - Ép kiểu là cơ chế để chuyển đổi tham chiếu đến đối tượng thuộc loại này thành tham chiếu đến đối tượng thuộc loại khác Có 2 loại ép kiểu dữ liệu : + Implicit casting : (ép kiểu ngầm định) : ép từ subtype lên super type + Explicit casting : (ép kiểu tường minh) : ép từ supertype thành subtype - Vì sao cần ép kiêu dữ liệu: + Java là ngôn ngữ kiểm soát dữ liệu chặt chẽ ,cần ép kiểu khi không tương thích + Cần chuyển đổi giữa các kiểu số int ,double,float,... +Cần ép kiểu đối tượng,dùng khi kế thừa + Nếu không ép đúng kiểu thì chương trình sẽ không biên dịch được gây ra lỗ |
| **10. Lưu ý khi ép kiểu ?** | Lưu ý khi ép kiểu dữ liệu : - Chỉ ép khi cần thiết - Khi ép từ kiểu dữ liệu lớn sang kiểu dữ liệu nhỏ có thể bị mất dữ liệu - Khi cần ép kiểu đối tượng cần dùng instanceof để kiểm tra kiểu dữ liệu trước - Không thể ép các kiểu dữ liệu không liên quan |