|  |  |
| --- | --- |
| **1. Tính đa hình là gì? Cho ví dụ** | Tính đa hình (Polymorphism) Cùng một đối tượng sẽ có các thể hiện khác nhau tùy thuộc vào ngữ cảnh . Cùng một phương thức sẽ có các cách triển khai khác nhau tùy thuộc vào các lớp. Ví dụ: class Animal { void sound() { System.out.println("Animal makes a sound"); } } class Dog extends Animal { void sound() { System.out.println("Dog barks"); } } Animal a = new Dog(); a.sound(); // Output: Dog barks |
| **2. Tính đa hình thể hiện ở đâu trong Java?** | - Thể hiện rõ qua:  + Ghi đè phương thức (method overriding)  + Interface và abstract class  + Runtime Polymorphism: đối tượng có thể thay đổi hình dạng trong lúc chạy. |
| **3. Abstract method là gì? Đặc điểm** | - Là phương thức trừu tượng, chỉ khai báo mà không có phần thân (body). - Chỉ tồn tại trong abstract class hoặc interface. - Phải được override trong lớp con. |
| **4. Abstract class là gì? Đặc điểm** | - Là lớp trừu tượng không có đối tượng chỉ khai báo để kế thừa Đặc điểm:  + không sử dụng final  + có thuộc tính và method bình thường  + một abstract class có thể có hoặc không có abstract method  + một class có một abstract method thì class đó là abstract method  + môt class cụ thể kế thừa abstract class phải triển khai abstract method  + một abstract class kế thừa abstract class không nhất thiết phải có abstract method  + không thể khởi tạo đối tượng |
| **5. Interface là gì? Đặc điểm** | Là một abnr thiết kế quy định hành vi chung cho các class để sử dụng nó  Đặc điểm :  + không có contructor , không thể tạo object  + các method mặc định là abstract + public  + JDK 8 có thể tạo ra các method có phần thân (defoult, Static)  + một class cụ thể implement interface 🡺 phải triển khai abstract method  + một abstract class implement interface 🡺 không cần thiết phải triển khai abstract class  + một class có thể implement nhiều interface  + interface có thể extends nhiều interface khác |
| **6. So sánh Abstract class và Interface** | |  |  |  | | --- | --- | --- | | **Tiêu chí** | **Abtract class** | **Interface** | | **Từ khóa** | Abstract | Inheriface | | **Kế thừa** | Một lớp duy nhất | Có thể nhiều interface | | **Phương thức** | Có cả abstract và non-abastract | Mặc định là abstract(từ java 8 có default/static) | | **Biến** | Có thể là mọi loại biến | Chỉ có pulic static final | | **Contructor** | Có | Không có | | **Dùng khi** | Có tlieen quan “is-a” | Dùng để định nghĩa hành vi “can-do” | |
| **7. Khi nào dùng Abstract class và khi nào dùng Interface?** | Dùng Abstract khi :  + Có một khuôn mẫu chung và muốn kế thừa  + có dữ liệu thành phần và logic cụ thể  Dùng Interface khi :  + Muốn định nghĩa một loại hành vi  + Muốn hỗ trợ kế thừa đa hình |
| **8. Tìm hiểu Comparable và Comparator** | - Comparable:  + So sánh chính đối tượng với chính nó.  + Override compareTo().  + Nằm trong java.lang.  - Comparator:  + So sánh 2 đối tượng bất kỳ.  + Override compare().  + Nằm trong java.util. |