**1. Tại sao Java chỉ có đơn kế thừa?**

Java chỉ hỗ trợ đơn kế thừa bởi vì nó được coi là một tính năng an toàn và dễ hiểu hơn. Với đơn kế thừa, một lớp con chỉ có thể kế thừa từ một lớp cha duy nhất, giúp giảm thiểu sự phức tạp của cấu trúc lớp và đảm bảo tính nhất quán của kiểu dữ liệu. Ngoài ra, đơn kế thừa cũng giúp tránh được các vấn đề mâu thuẫn của kế thừa đa năng, nơi một lớp con có thể kế thừa từ nhiều lớp cha và xảy ra xung đột giữa các phương thức hoặc thuộc tính có tên giống nhau.

**2. Tại sao cần dùng interfaces?**

Interface là một cách để khai báo các phương thức mà các lớp khác nhau phải triển khai, bất kể chúng có cùng tính chất hay không. Nó cho phép các lớp không liên quan về mặt kế thừa vẫn có thể triển khai các phương thức chung, giúp tăng tính linh hoạt và tái sử dụng mã. Ngoài ra, sử dụng interface cũng giúp giảm sự phụ thuộc giữa các lớp, nâng cao tính mở rộng và khả năng thay đổi của chương trình.

**3. Interface vs Lớp cha trừu tượng (Abstract Superclass)?**

Cả Interface và Lớp cha trừu tượng đều được sử dụng để định nghĩa các phương thức mà các lớp con phải triển khai. Tuy nhiên, chúng có một số điểm khác nhau:

* Một lớp chỉ có thể kế thừa từ một lớp cha trừu tượng duy nhất, trong khi một lớp có thể triển khai nhiều interface.
* Lớp cha trừu tượng có thể có các phương thức có thân hàm, còn interface chỉ có thể khai báo các phương thức không có thân hàm.
* Lớp cha trừu tượng có thể định nghĩa các thuộc tính và phương thức không phải là trừu tượng, còn interface chỉ có thể khai báo các phương thức và thuộc tính là trừu tượng.
* Các lớp con kế thừa từ lớp cha trừu tượng có thể sử dụng các phương thức và thuộc tính được định