1. **Interface là gì? Tại sao dùng Interface**

Interface trong Java là một cơ chế để đạt được sự trừu tượng. Có thể chỉ có các phương thức trừu tượng trong Interface, không có phần thân phương thức. Nó được sử dụng để đạt được tính trừu tượng và đa kế thừa trong Java.

Tại sao dùng interface

Sử dụng Interface để đạt được sự trừu tượng. (Tính trừu tượng là một tiến trình ẩn các cài đặt chi tiết và chỉ hiển thị tính năng tới người dùng.

Nói cách khác, nó chỉ hiển thị các thứ quan trọng tới người dùng và ẩn các chi tiết nội tại, ví dụ: để gửi tin nhắn, người dùng chỉ cần soạn text và gửi tin. Bạn không biết tiến trình xử lý nội tại về phân phối tin nhắn.

)

Interface có thể hỗ trợ chúng ta đạt được đa kế thừa (Vì Java mặc định không hỗ trợ đa kế thừa thông qua class).

Nó có thể được sử dụng để đạt được Loose coupling.

**Điểm giống – khác nhau giữa Interface và Abstract class:**

Một interface về cơ bản hoàn toàn giống với abstract class. Một interface cũng có các abstract method và các method được thiết kế để implement ở các class khác. Việc implement này hoàn toàn giống với việc extend và override method trong Abstract class.

Về điểm khác nhau, thì mình thấy có 2 điểm chính:

**Về cách viết code:** các method trong interface phải là empty method, tức là method không có nội dung, chỉ có tên hàm, tham số và kiểu dữ liệu trả về. Trong khi với abstract class thì bạn hoàn toàn có thể thêm abstract method hoặc method bình thường (là method có nội dung, logic bên trong body).

**Về mục đích sử dụng:**

Interface là một chức năng mà bạn có thể thêm và bất kì class nào. Từ chức năng ở đây không đồng nghĩa với method (hay còn gọi là hàm).

Abstract class là một class cha cho tất cả các class có cùng bản chất. Bản chất ở đây được hiểu là kiểu, loại, nhiệm vụ của class.

**Sự khác biệt giữa Interface và Lớp trong Java**

Không giống như một lớp, bạn không thể khởi tạo hoặc tạo một đối tượng của giao diện.

Tất cả các phương thức trong một giao diện nên được khai báo là trừu tượng.

Một interface trong Java không chứa bất kỳ hàm tạo nào, nhưng một lớp có thể.

Một giao diện không thể chứa các trường cá thể. Nó chỉ có thể chứa các trường được khai báo là tĩnh và cuối cùng.

Một giao diện không thể được mở rộng hoặc kế thừa bởi một lớp; nó được thực hiện bởi một lớp.

Một interface trong Java không thể triển khai bất kỳ lớp nào hoặc một giao diện khác.