

Dò bài cũ



1. Kế thừa và đặc điểm của kết thừa

2. Từ khóa final

3. Tính đa hình

Kế thừa

Đặc điểm



- Là cơ chế cho phép lớp con sử dụng lại các đặc điểm và hành vi đã được định nghĩa ở lớp cha.
- Mục đích
 - Tái sử dụng mã nguồn

Mối quan hệ và đặc điểm



- Mối quan hệ giữa lớp cha và lớp con là : **is – a**
 - **Lớp cha** (super class, parent class, base class)
 - **Lớp con** (subclass, child class, derived class – lớp dẫn xuất)
- Đặc điểm
 - Lớp con **không thể kế thừa constructor** của lớp cha
 - Lớp con **không** thể truy cập tài sản **private** của lớp cha
 - Java không hỗ trợ đa kế thừa.

Final



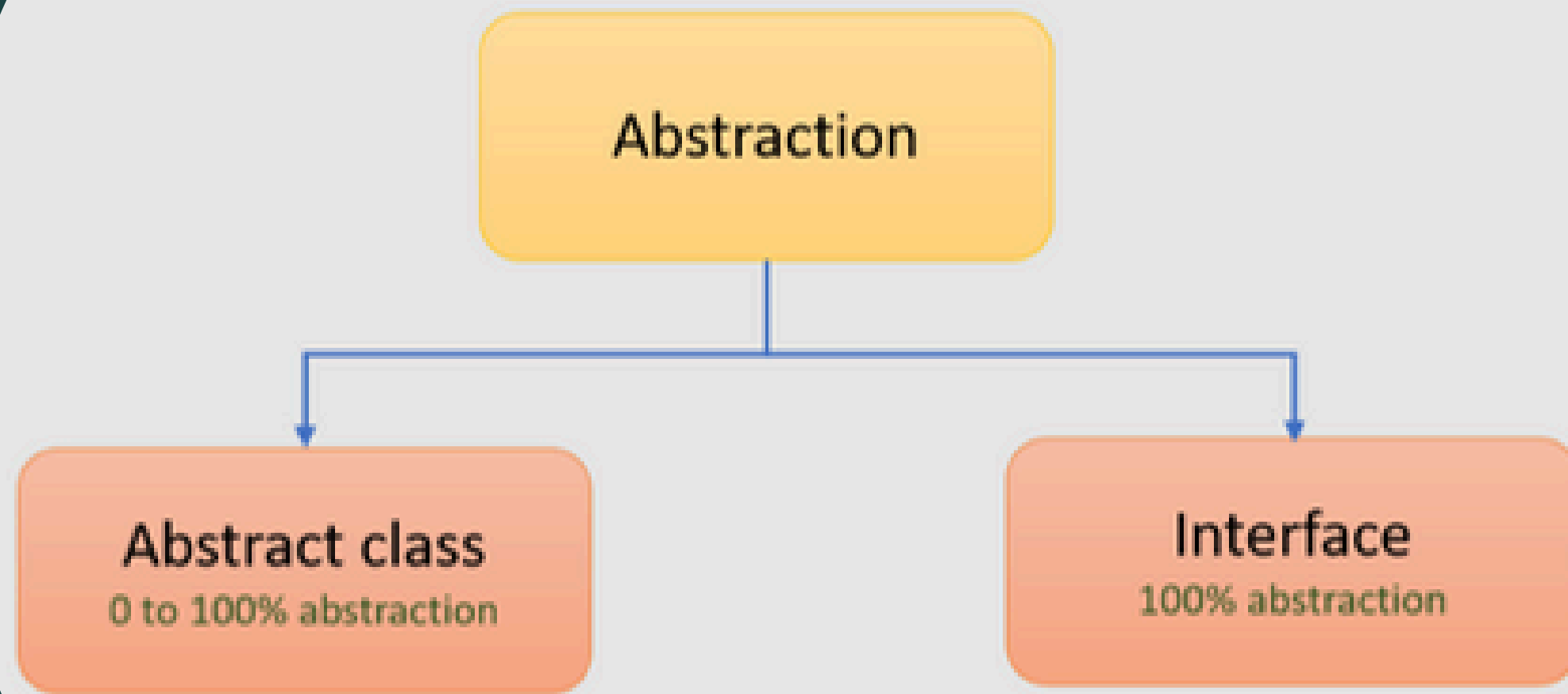
- Được sử dụng trong rất nhiều ngữ cảnh
 - Đi với biến
 - Hằng số
 - Đi với phương thức
 - Không cho class con ghi đè
 - Đi với class
 - Không cho kế thừa (Vô sinh)

Tính đa hình



- Là khả năng một đối tượng có thể hiện/hành vi theo nhiều cách khác nhau tùy thuộc vào ngữ cảnh.
- **Overriding (ghi đè)**
 - Định nghĩa lại phương thức của lớp cha sao cho phù hợp với lớp con.
 - Xảy ra ở quan hệ **is-a**
 - Đa hình tại **runtime**
- **Overloading (Nạp chồng)**
 - Một lớp có nhiều phương thức có cùng tên, nhưng khác nhau về tham số truyền vào.
 - Xảy ra trong **1 class**
 - Đa hình tại **compile**

Abstract class, Interface





NỘI DUNG CHÍNH

1	Khái niệm về tính trừu tượng
2	Abstract class
3	Interface
4	Phân biệt



Tính trừu tượng (Abstraction)



- Là khả năng **ẩn các chi tiết** xử lý, chỉ **hiển thị tính năng/kết quả** (tên phương thức) cho người dùng
- Chỉ quan tâm đến **kết quả** đạt được, không quan tâm cách thực hiện/implement.
- Trong **java** tính **trừu tượng** được thể hiện qua **abstract class** và **interface**.

Abstract class

Tính chất



- Là class có tính trừu tượng cao đến mức **không** thể **tạo** được **đối tượng**
- **Lợi ích**
 - **Không** muốn tạo ra **đối tượng**
 - Có một số phương thức **không thể triển khai**
 - Sử dụng một đối tượng của một lớp mà **không cần quan tâm** đối tượng đó thực sự thuộc lớp nào.
- **Lưu ý**
 - Phương thức **rỗng**: **Không làm gì cả**
 - Phương thức **trừu tượng**: **Cần** phải được **triển khai** bởi các lớp con

Các tính chất của abstract



- Không thể tạo được đối tượng.
- Một lớp chứa **method abstract** thì bắt buộc lớp đó phải là **lớp abstract**. (ngược lại thì không)
- **Lớp abstract** có thể chứa thuộc tính và **method bình thường**.
- **Abstract** không thể đi chung với **final**.
- **Lớp abstract** có thể **extend** từ **lớp abstract** khác. (không cần override method abstract của lớp cha)

interface

Tính chất



- **Interface** là một bản **thiết kế** của một **lớp**
- Quy định các **hành vi chung** cho lớp triển khai nó. (dùng từ khóa implements)

Các tính chất của interface

Java 8



- Không thể tạo được đối tượng.
- Các trường trong interface đều là public, static và final.
- Các method toàn bộ là public abstract.
- Không tạo được constructor.
- Có thể extend từ một nhiều hoặc nhiều interface khác.

inteface của Java 8



- Từ **java 8** trở đi, **interface** bổ sung thêm **2** loại phương thức có body
 - **Static method**
 - **Không cho** phép ghi đè
 - Gọi thông qua tên interface
 - **Default method**
 - **Cho** phép ghi đè
 - **Thêm** phương thức mới vào **interface** mà không làm ảnh hưởng đến các **lớp đã triển**



DATABASE

**THANK
YOU**