

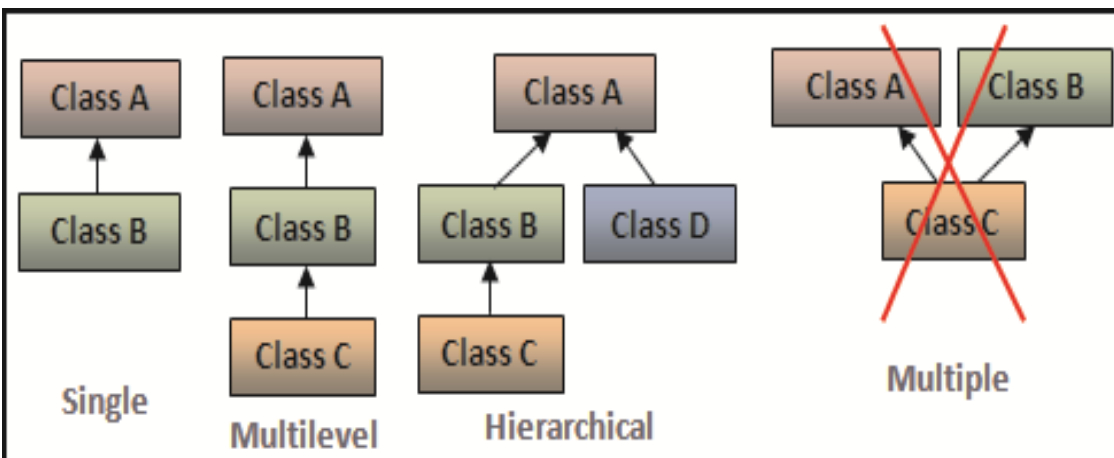
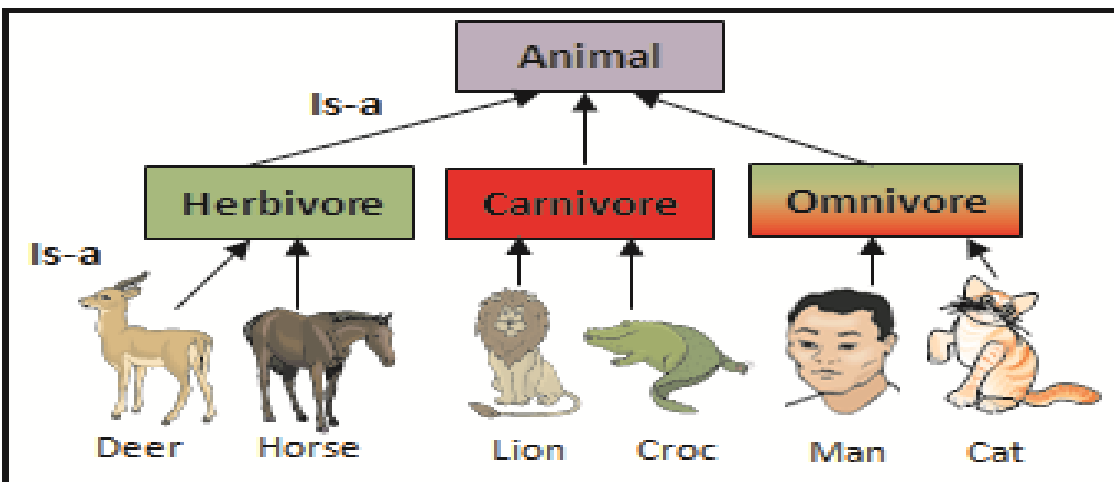


Bài 5

Trừu tượng và Kế thừa Interface

- Bổ từ truy cập static và final
- Kế thừa và ghi đè (Overriding)
- Từ khóa abstract và Lớp trừu tượng
- Interface

- Bổ từ truy cập static
 - Dùng để quản lý bộ nhớ tốt hơn
 - Có thể truy cập trực tiếp thông qua lớp mà không cần khởi tạo đối tượng
 - Thuộc về lớp chứ không thuộc về instance của lớp
 - Được sử dụng với biến, phương thức, khối lệnh, lớp lồng (Nested Class)
- Bổ từ truy cập final
 - Dùng để hạn chế thao tác của người dùng
 - Được sử dụng với biến, phương thức, lớp
 - Final variable: là biến hằng số, không thể thay đổi và chỉnh sửa
 - Final Method: Không thể bị ghi đè và ẩn đi trong lớp con
 - Final Class: Không thể bị kế thừa, các biến và phương thức trong Final Class đều được hiểu ngầm được khai báo là final

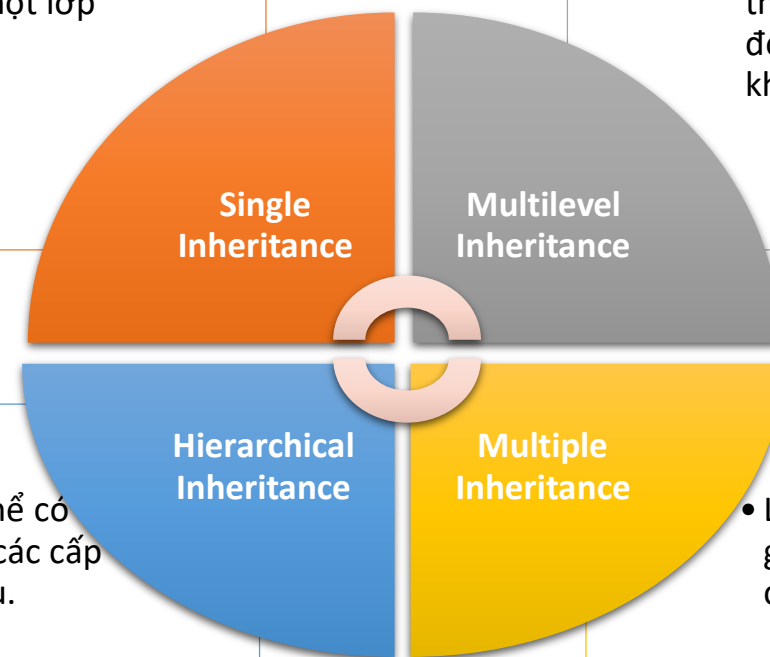


- Lớp chỉ kế thừa từ một và chỉ một lớp cha.

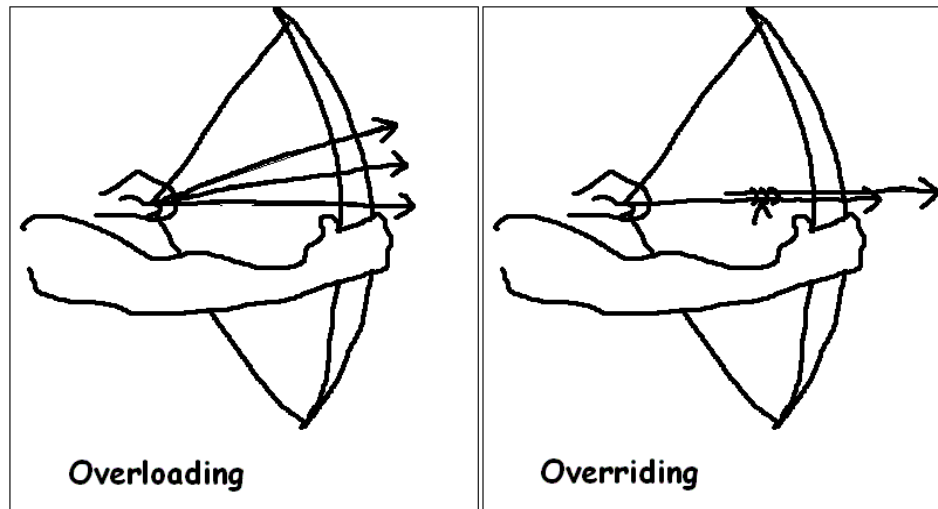
- Lớp con có thể kế thừa từ lớp mà lớp đó lại là con của lớp khác.

- Lớp cha có thể có nhiều con ở các cấp độ khác nhau.

- Lớp con có nguồn gốc từ hơn một lớp cha.



- Ghi đè – Overriding
 - Cho phép tạo tên phương thức giống với phương thức ở lớp cha
 - Thay đổi hành vi của lớp cha khi cần thiết
 - Quy tắc ghi đè
 - Phương thức ghi đè cần trùng tên, kiểu, số lượng tham số, kiểu dữ liệu trả về với phương thức ở lớp cha nó
 - Phương thức ghi đè phải có bộ từ truy cập có phạm vi lớn hơn hoặc bằng so với phương thức của lớp cha nó
- So sánh overloading và overriding



- Abstract

- Ứng dụng cho phương thức và lớp

- Abstract method:

- Được khai báo với từ khóa abstract và không có phần thực thi (không có thân)

- Cú pháp:

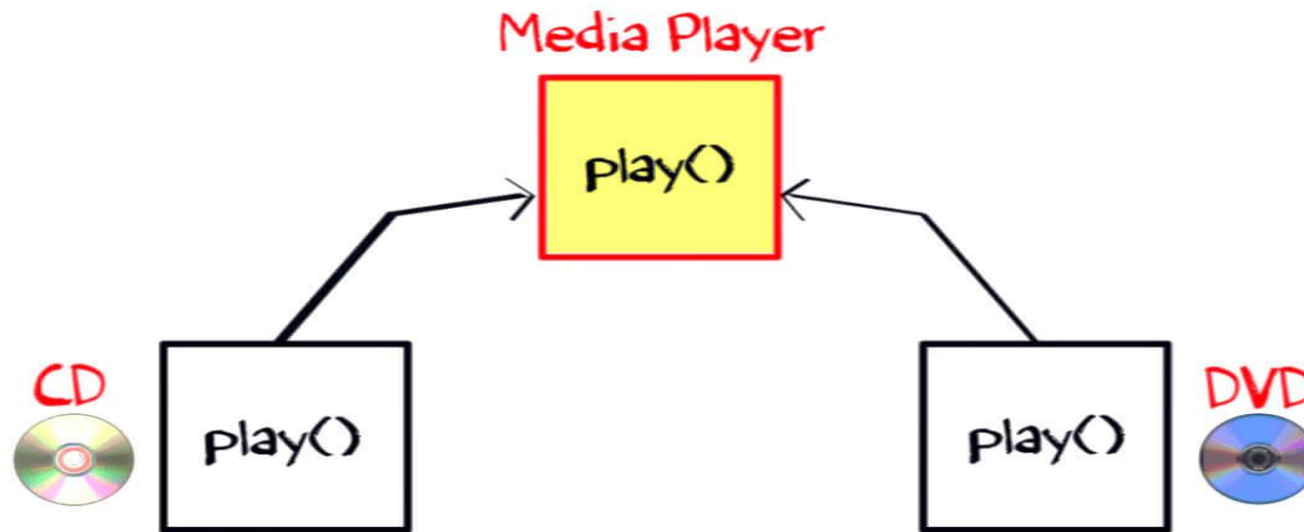
```
abstract ReturnDataType MethodName(Parameters);
```

- Abstract class:

- Có thể chứa abstract method và non-abstract method

- Cú pháp

```
AccessModifier abstract class ClassName
```



- Tập hợp các ràng buộc, quy định tiêu chuẩn (bản kiến trúc) cho các lớp kế thừa nó
- Các lớp kế thừa interface thì phải triển khai tất cả các phương thức được khai báo trong interface
- Giải quyết vấn đề đa kế thừa trong java – một lớp có thể kế thừa nhiều interface, interface có thể kế thừa từ nhiều interface khác

- Interface và Class giống nhau:

Interface có thể có nhiều phương thức.

Một interface lưu trữ trong file có phần mở rộng là .java và có tên phải trùng với tên file tương tự như Java class.

Mã bytecode của interface được lưu trữ trong .class file.

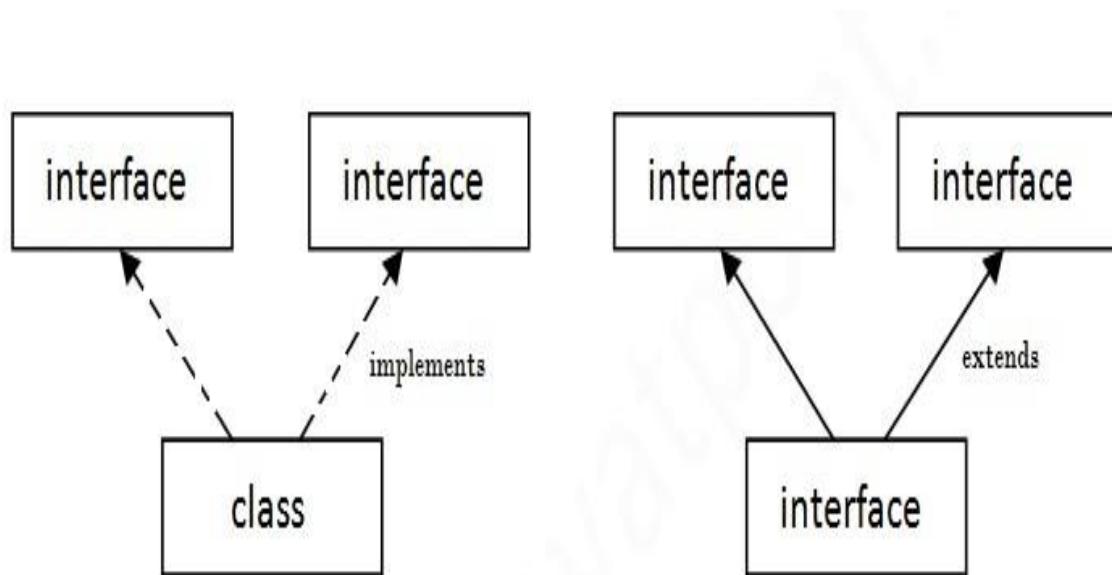
Interfaces được lưu trữ trong packages và bytecode file lưu trữ trong thư mục có cấu trúc khớp với tên package.

- Interface và Class khác nhau:

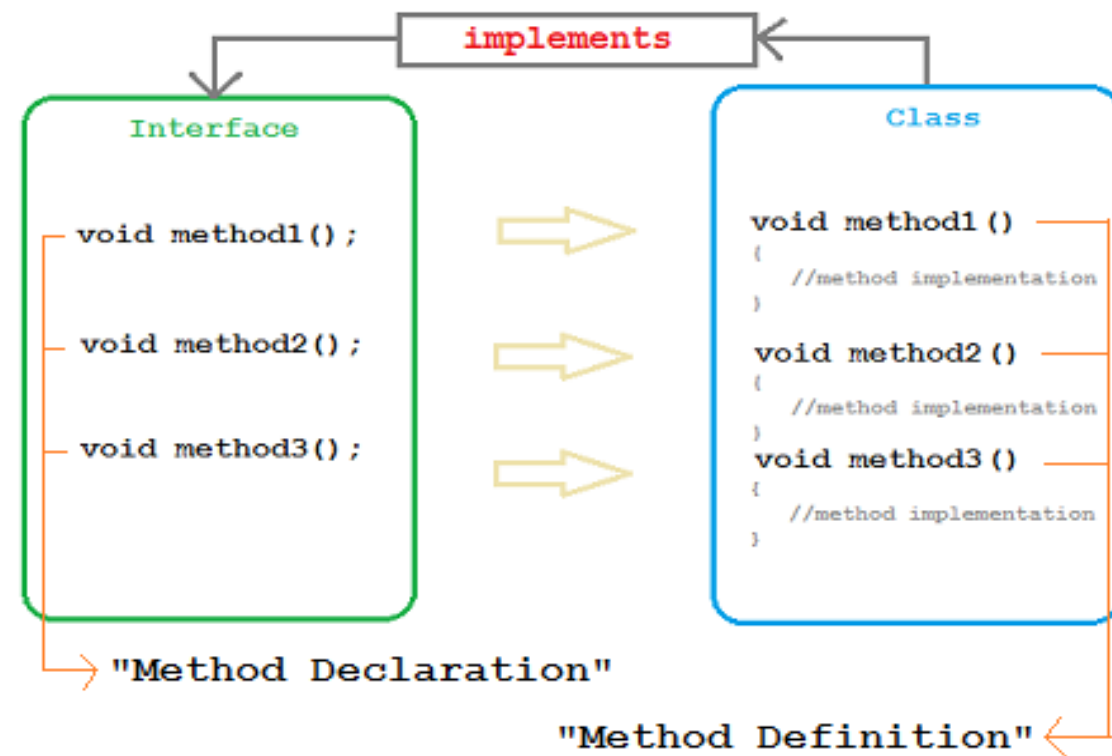


- Cú pháp:

AccessModifier interface InterfaceName extends <other-interfaces>



Multiple Inheritance in Java



Abstract Class	Interface
Có đồng thời phương thức abstract và non-abstract	Chỉ có phương thức abstract
Có thể có biến không khai báo final	Biến mặc định hiểu ngầm là khai báo final
Có các thành phần với các bộ từ truy cập là public, protected, private...	Thành phần mặc định khai báo là public
Kế thừa sử dụng từ khóa extends	Kế thừa sử dụng từ khóa implements
Có thể kế thừa từ một lớp khác và nhiều interface	Kế thừa từ nhiều interface khác

HỎI ĐÁP





TRẢI NGHIỆM THỰC HÀNH



TRÂN TRỌNG CẢM ƠN!



238 Hoàng Quốc Việt, Bắc Từ Liêm, Hà Nội



0968.27.6996



tuyensinh@bachkhoa-apttech.edu.vn



www.bachkhoa-apttech.edu.vn