



Java Inheritance

By Wislam

Tóm tắt

Kế Thừa

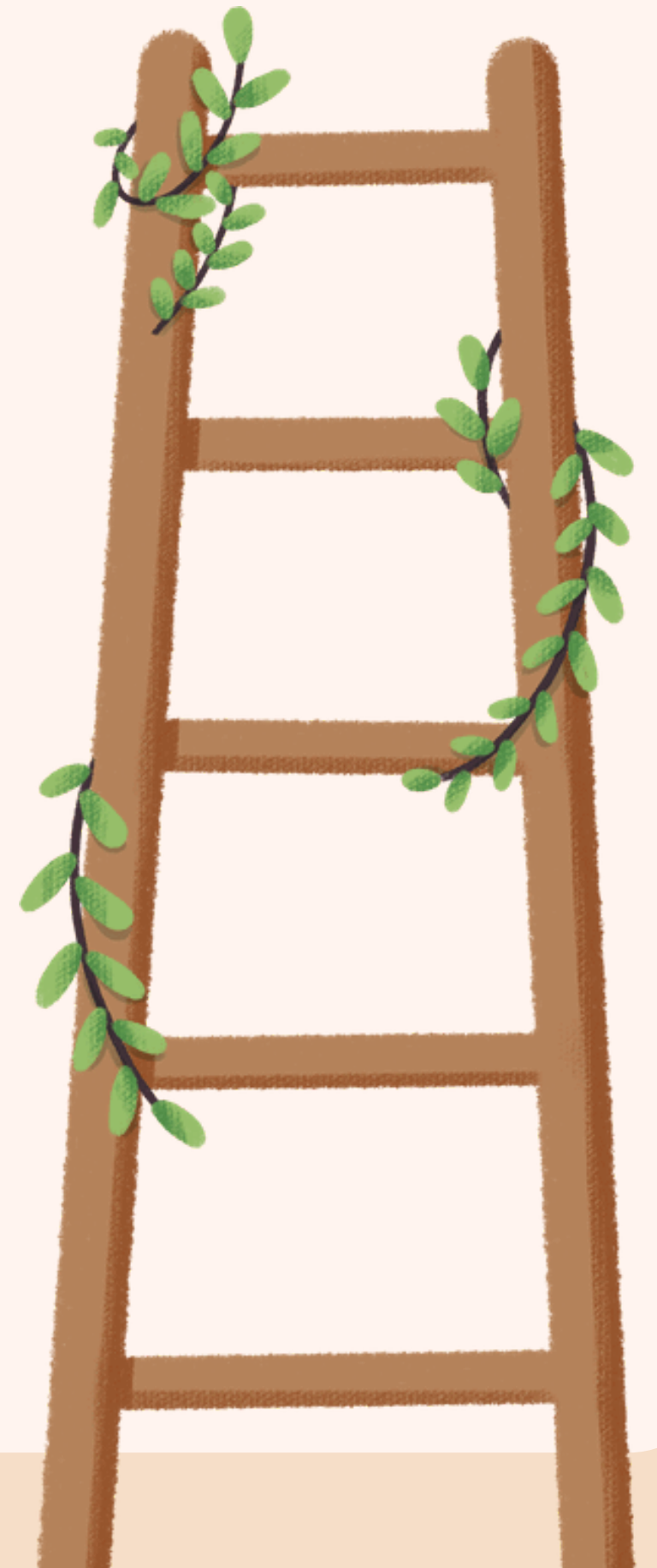
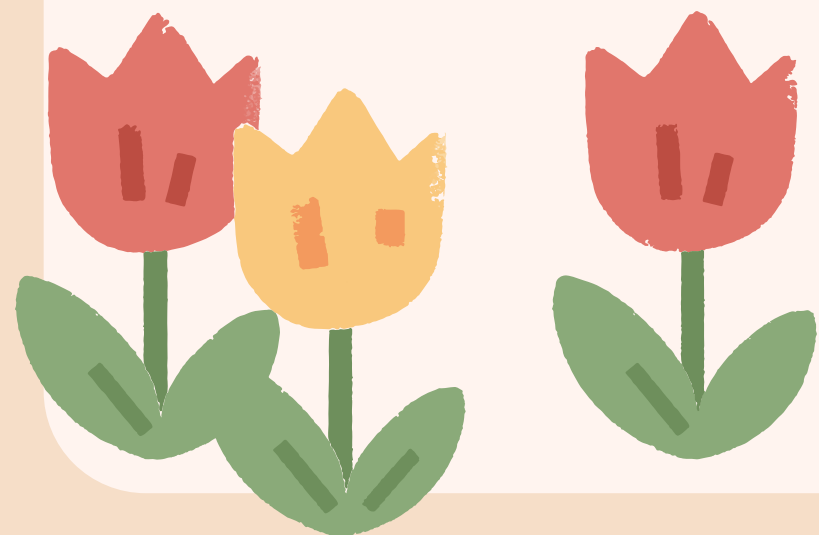
Truy Cập

Override

Polymorphism

Tổng kết

Ví dụ



Kế Thừa trong Java

Student_It

Gamer

coder

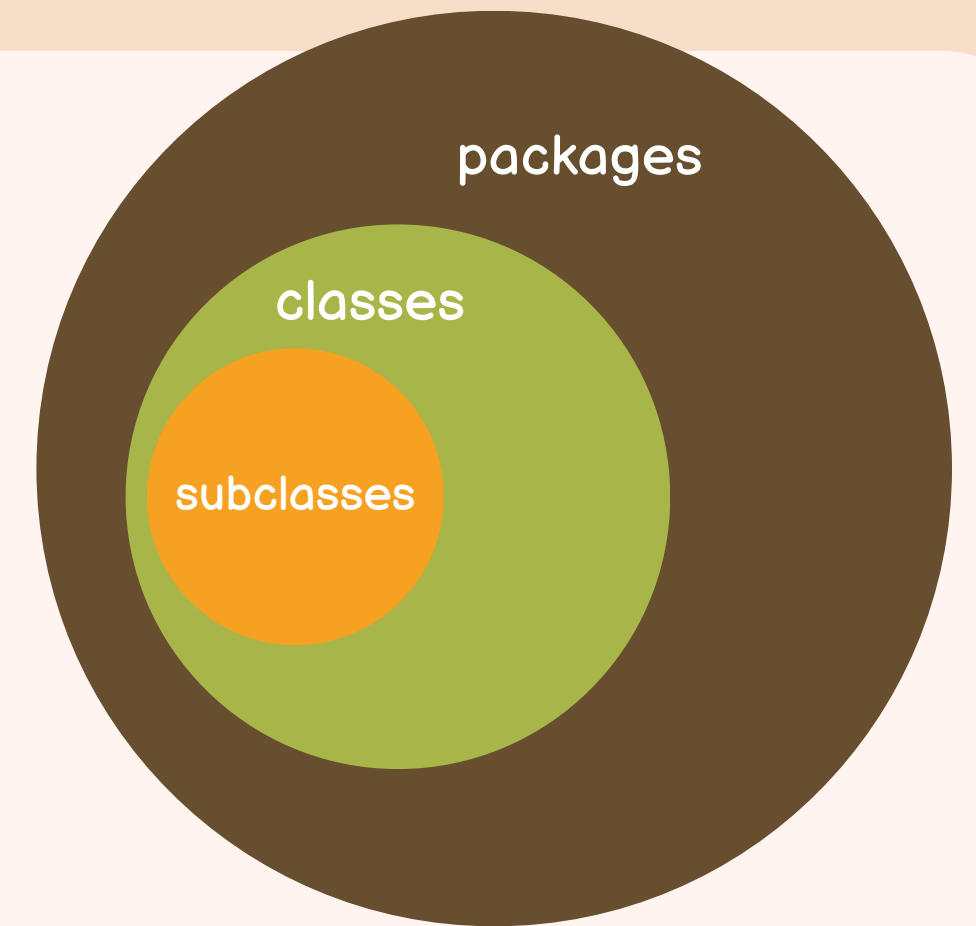
hacker

Định nghĩa: Kế thừa là một cơ chế trong đó một lớp con (subclass) kế thừa các thuộc tính và phương thức KHÔNG private của một lớp cha (superclass).

Mục đích: Tái sử dụng mã nguồn, tổ chức mã một cách hiệu quả hơn, và dễ dàng bảo trì.



Truy cập



01 PRIVATE

- Chỉ có thể truy cập trong cùng một **class**.
- Không thể truy cập từ **subclasses** hay **từ bất cứ nơi nào khác**.

02 PROTECTED

- Có thể truy cập trong cùng một **package** và các **subclass**, ngay cả khi các **subclass** nằm ở các **package** khác nhau.

03 PUBLIC

- Có thể truy cập từ bất kỳ đâu.

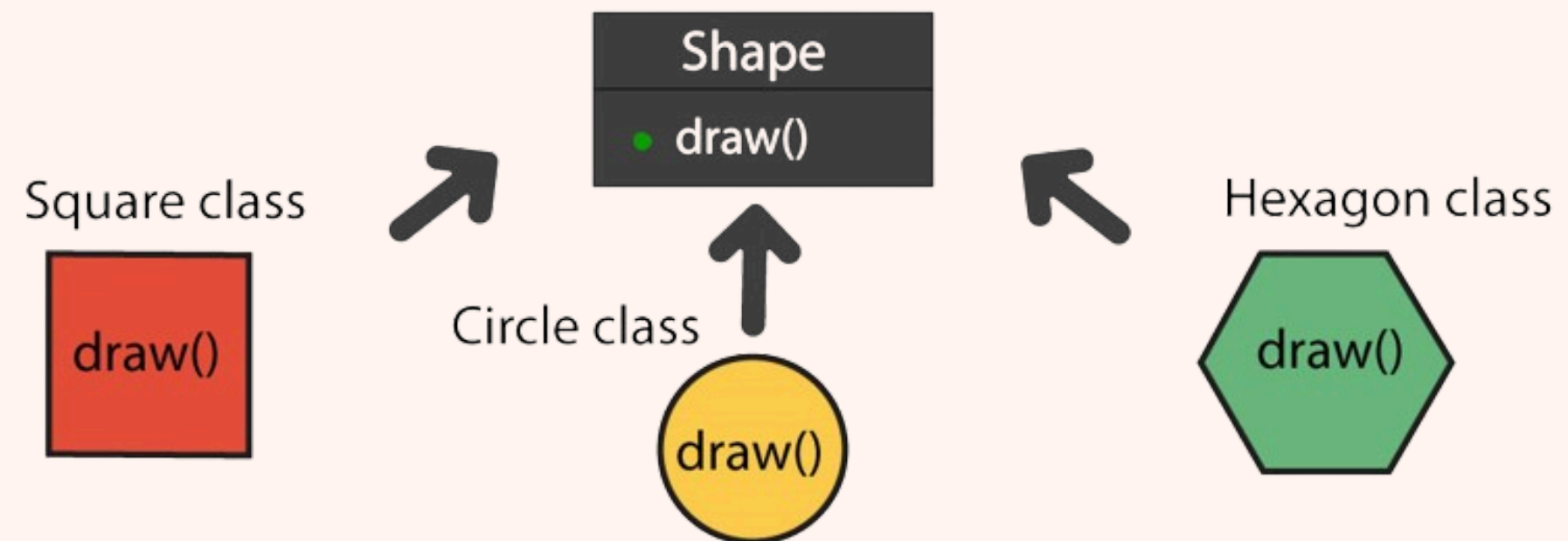
Ghi Đè Phương Thức (Method Override)

Định nghĩa: Khi một *subclass* có *method* *trùng tên*, kiểu trả về và tham số với *method* trong *superclass*.

Mục đích: Cho phép *subclass* *cung cấp cách triển khai cụ thể* của một *method* đã được định nghĩa trong *superclass*.

Sử dụng @Override: Annotations để *đảm bảo rằng method* đang ghi đè *method* của *superclass*.

Method Overriding





Tính Đa Hình (Polymorphism)



01 ĐỊNH NGHĨA

Khả năng của một **đối tượng** có thể **hành xử khác nhau** dựa trên **ngữ cảnh gọi**.

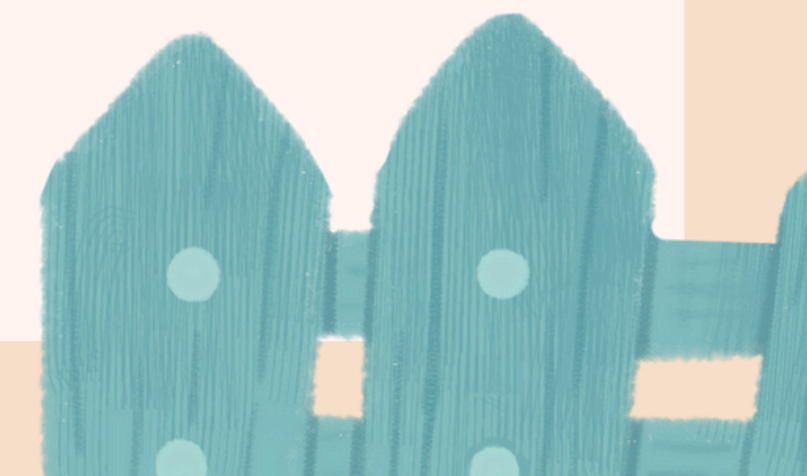
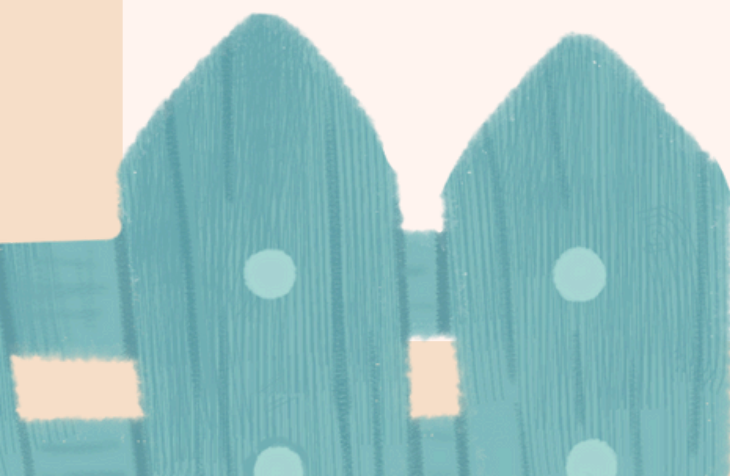
02 COMPILE-TIME POLYMORPHISM

Sử dụng nạp chồng (**method overloading**).

```
int myMethod(int x)
float myMethod(float x)
double myMethod(double x, double y)
```

03 RUNTIME POLYMORPHISM

Sử dụng ghi đè (**method overriding**) để **quyết định phương thức nào sẽ được gọi tại thời điểm chạy**.



Ví dụ

```
class Animal {  
    protected void makeSound() {  
        System.out.println("Animal sound");  
    }  
}
```

```
class Dog extends Animal {  
    @Override  
    public void makeSound() {  
        System.out.println("Woof!");  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Animal myDog = new Dog();  
        myDog.makeSound(); // Output: Woof!  
    }  
}
```

Tổng Kết



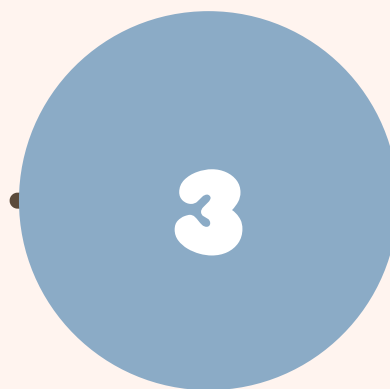
KẾ THỪA

là một trong những tính năng quan trọng nhất của lập trình hướng đối tượng, **giúp tái sử dụng** mã nguồn và xây dựng cấu trúc lớp dễ hiểu hơn.



TRUY CẬP

cung cấp **khả năng kiểm soát** việc truy cập dữ liệu và phương thức từ các lớp khác nhau, đảm bảo tính bảo mật và toàn vẹn của dữ liệu.



OVERRIDE

cho phép các lớp con cung cấp **triển khai cụ thể** cho các phương thức đã có trong lớp cha, giúp hiện thực hóa tính linh hoạt và mở rộng chức năng mà không thay đổi mã nguồn lớp cha.



POLYMORPHISM

là yếu tố then chốt trong lập trình hướng đối tượng, cho phép các **đối tượng được xử lý khác nhau tùy thuộc vào lớp mà chúng thuộc về**. Điều này làm cho các chương trình dễ mở rộng và bảo trì hơn.