

Ghi đè phương thức trong Java

Nếu lớp con có cùng phương thức như đã được khai báo trong lớp cha, thì đó gọi là **Ghi đè phương thức (Method Overriding)** trong Java. Nói cách khác, nếu lớp con cung cấp trình triển khai cụ thể của phương thức mà đã được cung cấp bởi một trong các lớp cha của nó, thì đó là ghi đè phương thức.

Sự sử dụng của ghi đè phương thức trong Java

Ghi đè phương thức được sử dụng để cung cấp trình triển khai cụ thể của một phương thức mà đã được cung cấp bởi lớp cha của nó.

Ghi đè phương thức được sử dụng để thu được tính đa hình tại runtime.

Quy tắc cho ghi đè phương thức trong Java

- Phương thức phải có cùng tên như trong lớp cha.
- Phương thức phải có cùng tham số như trong lớp cha.
- Phải là quan hệ IS-A (kế thừa).

Vấn đề xảy ra nếu không có ghi đè phương thức

Bạn theo dõi ví dụ sau để hiểu vấn đề chúng ta có thể phải đối mặt nếu chúng ta không sử dụng ghi đè phương thức.

```
class Vehicle{
    void run(){System.out.println("Vehicle đang chạy");}
}
class Bike extends Vehicle{

    public static void main(String args[]){
        Bike obj = new Bike();
        obj.run();
    }
}
```

Vấn đề là bạn phải cung cấp một trình triển khai cụ thể của phương thức run() trong lớp con.

Ví dụ về ghi đè phương thức trong Java

Trong ví dụ này, chúng ta định nghĩa phương thức run trong lớp con như đã được định nghĩa trong lớp cha nhưng nó có trình triển khai cụ thể. Tên và tham số của phương thức là giống nhau và quan hệ giữa hai lớp là IS-A, vì thế đó là ghi đè phương thức.

```
class Vehicle{
    void run(){System.out.println("Vehicle đang chạy");}
}

class Bike2 extends Vehicle{
    void run(){System.out.println("Bike đang chạy an toàn");}

    public static void main(String args[]){
        Bike2 obj = new Bike2();
        obj.run();
    }
}
```

Ví dụ thực về ghi đè phương thức trong Java

Giả sử Bank là một đối tượng cung cấp lãi suất. Nhưng lãi suất lại khác nhau giữa từng ngân hàng. Ví dụ, các ngân hàng VCB, AGR và CTG có thể cung cấp các lãi suất lần lượt là 8%, 7% và 9%.

```
class Bank{
    int getRateOfInterest(){return 0;}
}

class VCB extends Bank{
    int getRateOfInterest(){return 8;}
}

class AGR extends Bank{
    int getRateOfInterest(){return 7;}
}

class CTG extends Bank{
    int getRateOfInterest(){return 9;}
}
```

```
class Test2{
public static void main(String args[]){
VCB s=new VCB();
AGR i=new AGR();
CTG a=new CTG();
System.out.println("VCB Rate of Interest: "+s.getRateOfInterest());
System.out.println("AGR Rate of Interest: "+i.getRateOfInterest());
System.out.println("CTG Rate of Interest: "+a.getRateOfInterest());
}
}
```

Câu hỏi: Chúng ta có thể ghi đè phương thức static?

Không, phương thức static không thể được ghi đè. Điều này có thể được chứng minh bởi tính đa hình tại runtime, chúng ta sẽ tìm hiểu sau.

Câu hỏi: Tại sao chúng ta không thể ghi đè phương thức static?

Bởi vì phương thức static được gắn kết với lớp trong khi đó phương thức instance được gắn kết với đối tượng. Static thuộc sở hữu Class Area và instance thuộc sở hữu Heap Area.

Câu hỏi: Chúng ta có thể ghi đè phương thức main?

Không, bởi vì main là phương thức static.

Điểm khác nhau giữa ghi đè phương thức và nạp chồng phương thức trong Java?

Có nhiều điểm khác nhau giữa nạp chồng phương thức và ghi đè phương thức trong Java. Bảng dưới đây liệt kê các điểm khác nhau này:

STT	Nạp chồng phương thức	Ghi đè phương thức

1	Nạp chồng phương thức được sử dụng để tăng tính có thể đọc của chương trình	Ghi đề phương thức được sử dụng để cung cấp trình triển khai cụ thể của phương thức mà đã được cung cấp bởi lớp cha của nó
2	Nạp chồng phương thức được thực hiện bên trong lớp	Ghi đề phương thức xuất hiện trong hai lớp mà có mối quan hệ IS-A (kế thừa)
3	Trong Nạp chồng phương thức, tham số phải khác nhau	Trong Ghi đề phương thức, tham số phải là giống nhau
4	Nạp chồng phương thức là ví dụ của đa hình tại compile time	Ghi đề phương thức là ví dụ của

		đa hình tại runtime
5	Trong Java, Nạp chồng phương thức không thể được thực hiện bởi thay đổi kiểu trả về của phương thức. Kiểu trả về có thể là giống hoặc khác trong Nạp chồng phương thức. Nhưng bạn phải thay đổi tham số	Kiểu trả về phải là giống hoặc covariant trong Ghi đề phương thức