

Nạp chồng phương thức trong Java

Nếu một lớp có nhiều phương thức cùng tên nhưng có số tham số khác nhau, thì đó là nạp chồng phương thức (Method Overloading). Nếu bạn phải thực hiện chỉ một hoạt động, có cùng tên phương thức, thì kỹ thuật này làm tăng tính có thể đọc cho chương trình. Giả sử bạn viết phương thức như `a(int, int)` cho hai tham số, và `b(int, int, int)` cho ba tham số, thì khi đó điều này có thể gây khó khăn cho bạn cũng như cho các lập trình viên khác để hiểu hành vi của phương thức, bởi vì tên của nó là khác nhau. Vì thế, chúng ta thực hiện nạp chồng phương thức để giúp việc phân tích chương trình nhanh hơn.

Có hai cách để nạp chồng phương thức trong Java, đó là:

- Bằng việc thay đổi số tham số
- Bằng việc thay đổi kiểu dữ liệu

Ghi chú: Trong Java, nạp chồng phương thức là không thể bằng việc thay đổi kiểu trả về của phương thức.

Ví dụ về nạp chồng phương thức bằng cách thay đổi số tham số

Trong ví dụ này, chúng ta sử dụng nạp chồng phương thức: phương thức `sum` đầu tiên thực hiện phép cộng hai số và phương thức `sum` thứ hai thực hiện phép cộng ba số.

```
class Calculation{  
    void sum(int a,int b){System.out.println(a+b);}   
    void sum(int a,int b,int c){System.out.println(a+b+c);}   
  
    public static void main(String args[]){  
        Calculation obj=new Calculation();  
        obj.sum(10,10,10);  
        obj.sum(20,20);  
    }  
}
```

Kết quả là:

Output:30

40

Ví dụ nạp chồng phương thức bằng cách thay đổi kiểu dữ liệu của tham số

Trong ví dụ này, chúng ta sử dụng nạp chồng phương thức mà khác nhau về kiểu dữ liệu. Phương thức sum đầu tiên nhận hai số nguyên và phương thức sum thứ hai nhận hai số thực double.

```
class Calculation2{  
    void sum(int a,int b){System.out.println(a+b);}   
    void sum(double a,double b){System.out.println(a+b);}   
  
    public static void main(String args[]){  
        Calculation2 obj=new Calculation2();  
        obj.sum(10.5,10.5);  
        obj.sum(20,20);  
  
    }  
}
```

Kết quả là:

Output:21.0

40

Câu hỏi: Tại sao nạp chồng phương thức là không thể bằng cách thay đổi kiểu trả về của phương thức?

Bởi vì việc này có thể gây ra tính lưỡng nghĩa, mơ hồ (ambiguity). Bạn theo dõi ví dụ sau để thấy rõ điều này:

Kết quả là:

```
class Calculation3{  
    int sum(int a,int b){System.out.println(a+b);}   
    double sum(int a,int b){System.out.println(a+b);}   
}
```

```
public static void main(String args[]){
    Calculation3 obj=new Calculation3();
    int result=obj.sum(20,20); //Gay ra Compile Time Error

}
}
```

Trong dòng code này: `int result=obj.sum(20,20);` thì Java có thể quyết định phương thức `sum()` nào nên được gọi?

Câu hỏi: Chúng ta có thể nạp chồng phương thức `main()`?

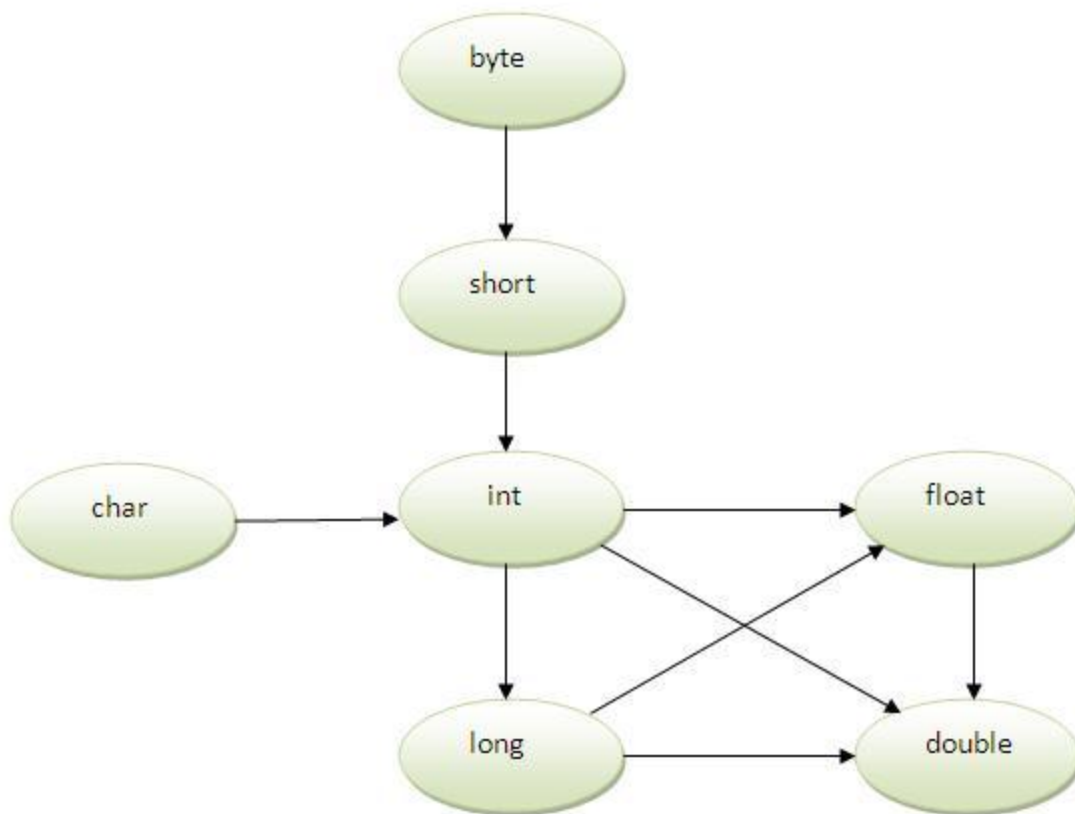
Có. Bạn có thể có bất kỳ số lượng phương thức `main` nào trong một lớp bởi nạp chồng phương thức. Bạn theo dõi ví dụ đơn giản sau:

```
class Overloading1{
    public static void main(int a){
        System.out.println(a);
    }

    public static void main(String args[]){
        System.out.println("Phuong thuc main() duoc trieu hoi");
        main(10);
    }
}
```

Nạp chồng phương thức và TypePromotion trong Java

Một kiểu được promote thành một kiểu khác hoàn toàn nếu không tìm thấy kiểu dữ liệu nào được kết nối. Bạn theo dõi sơ đồ sau để hiểu khái niệm này.



Như sơ đồ trên, byte có thể được promote thành short, int, long, float, hoặc double. Kiểu dữ liệu short có thể được promote thành int, long, float, hoặc double. Kiểu dữ liệu char có thể được promote thành kiểu int, long, float hoặc double và ...

Ví dụ về nạp chồng phương thức với TypePromotion:

```
class OverloadingCalculation1{
    void sum(int a,long b){System.out.println(a+b);}
    void sum(int a,int b,int c){System.out.println(a+b+c);}

    public static void main(String args[]){
        OverloadingCalculation1 obj=new OverloadingCalculation1();
        obj.sum(20,20); //Bay gio int literal thu hai se duoc promote thanh long
        obj.sum(20,20,20);
    }
}
```

```
}  
}
```

Ví dụ về nạp chồng phương thức với TypePromotion nếu kết nối là không được tìm thấy

Nếu có kết nối về các tham số kiểu trong phương thức, thì TypePromotion không được thực hiện.

```
class OverloadingCalculation2{  
    void sum(int a,int b){System.out.println("phuong thuc int arg duoc trieu hoi");}  
    void sum(long a,long b){System.out.println("phuong thuc long arg duoc trieu hoi");}  
  
    public static void main(String args[]){  
        OverloadingCalculation2 obj=new OverloadingCalculation2();  
        obj.sum(20,20); //Bay gio phuong thuc int arg sum() duoc trieu hoi  
    }  
}
```

Ví dụ về nạp chồng phương thức với TypePromotion trong trường hợp lưỡng nghĩa

Nếu không có kết nối các tham số kiểu trong phương thức, và mỗi phương thức promote số tham số giống nhau, thì đó là trường hợp lưỡng nghĩa.

```
class OverloadingCalculation3{  
    void sum(int a,long b){System.out.println("Phuong thuc a duoc trieu hoi");}  
    void sum(long a,int b){System.out.println("Phuong thuc b duoc trieu hoi");}  
  
    public static void main(String args[]){  
        OverloadingCalculation3 obj=new OverloadingCalculation3();  
        obj.sum(20,20); //Khong co tinh luong nghia o day  
    }  
}
```

Kết quả là:

Output:Compile Time Error