

Tính kế thừa trong java

Tính đóng gói trong java

Overloading phương thức trong java

Kế thừa trong java là sự liên quan giữa hai class với nhau, trong đó có class cha (superclass) và class con (subclass). Khi kế thừa class con được hưởng tất cả các phương thức và thuộc tính của class cha. Tuy nhiên, nó chỉ được truy cập các thành viên public và protected của class cha. Nó không được phép truy cập đến thành viên private của class cha.

Tư tưởng của kế thừa trong java là có thể tạo ra một class mới được xây dựng trên các lớp đang tồn tại. Khi kế thừa từ một lớp đang tồn tại bạn có sử dụng lại các phương thức và thuộc tính của lớp cha, đồng thời có thể khai báo thêm các phương thức và thuộc tính khác.

Cú pháp của kế thừa trong java

Sử dụng từ khóa *extends* để kế thừa.

```
class Subclass-name extends Superclass-name {
    //methods and fields
}
```

Ví dụ về kế thừa trong java

```
class Employee {
    float salary = 1000;
}

class Programmer extends Employee {
    int bonus = 150;
}

public class InheritanceSample1 {
    public static void main(String args[]) {
        Programmer p = new Programmer();
        System.out.println("Programmer salary is: " + p.salary);
        System.out.println("Bonus of Programmer is: " + p.bonus);
    }
}
```

Recent Updates

- Hàm DateAdd() trong VBA
- Hàm CDate() trong VBA
- Hàm Date() trong VBA
- String trong VBA
- Hàm Trim() trong VBA
- Hàm Right() trong VBA
- Hàm Left() trong VBA
- Hàm UCase() trong VBA
- Hàm LCase() trong VBA
- Hàm InStrRev() trong VBA
- Hàm InStr() trong VBA
- Hàm StrReverse() trong VBA

VietTuts on facebook



Học Java Cơ Bản Đến Nào Thì Học
2.136 lượt thích
it works..... Why?

Kết quả:

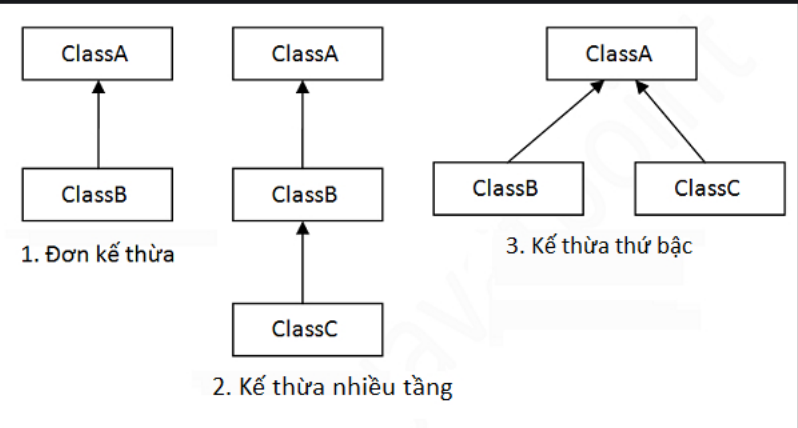
Programmer salary is: 1000.0
Bonus of Programmer is: 150

Trong ví dụ trên class Programmer là con của class Employee, nên nó được phép truy cập đến trường *salary* của class cha.

Các kiểu kế thừa trong java

Có 3 kiểu kế thừa trong java đó là đơn kế thừa, kế thừa nhiều cấp, kế thừa thứ bậc.

Khi một class được kế thừa từ nhiều class được gọi là đa kế thừa. Trong java, đa kế thừa chỉ được support thông qua interface, như đã được nói đến trong bài [interface trong java](#)

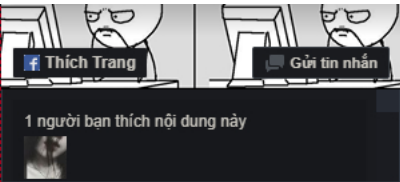


⚠️ **Chú ý:** Đa kế thừa trong java không được support thông qua class.

Ví dụ về đơn kế thừa

File: TestInheritance1.java

```
class Animal {  
    void eat() {  
        System.out.println("eating...");  
    }  
}  
  
class Dog extends Animal {  
    void bark() {  
        System.out.println("barking...");  
    }  
}  
  
public class TestInheritance1 {  
    public static void main(String args[]) {  
        Dog d = new Dog();  
        d.bark();  
        d.eat();  
    }  
}
```



```
    }  
}
```

Output:

```
barking...  
eating...
```

Ví dụ về kế thừa nhiều cấp

File: TestInheritance2.java

```
class Animal {  
    void eat() {  
        System.out.println("eating...");  
    }  
}  
  
class Dog extends Animal {  
    void bark() {  
        System.out.println("barking...");  
    }  
}  
  
class BabyDog extends Dog {  
    void weep() {  
        System.out.println("weeping...");  
    }  
}  
  
public class TestInheritance2 {  
    public static void main(String args[]) {  
        BabyDog d = new BabyDog();  
        d.weep();  
        d.bark();  
        d.eat();  
    }  
}
```

Kết quả:

```
weeping...  
barking...  
eating...
```

Ví dụ về kế thừa thứ bậc

File: TestInheritance3.java

```
class Animal {  
    void eat() {  
        System.out.println("eating...");  
    }  
}  
  
class Dog extends Animal {  
    void bark() {  
        System.out.println("barking...");  
    }  
}
```

```
class Dog extends Animal {
    void bark() {
        System.out.println("barking...");
    }
}

class Cat extends Animal {
    void meow() {
        System.out.println("meowing...");
    }
}

public class TestInheritance3 {
    public static void main(String args[]) {
        Cat c = new Cat();
        c.meow();
        c.eat();
        // c.bark(); // compile error
    }
}
```

Kết quả:

```
meowing...
eating...
```

Câu hỏi: Tại sao đa kế thừa không được support trong java?

Để giảm thiểu sự phức tạp và đơn giản hóa ngôn ngữ, đa kế thừa không được support trong java.

Hãy suy xét kịch bản sau: Có 3 lớp A, B, C. Trong đó lớp C kế thừa từ các lớp A và B. Nếu các lớp A và B có phương thức giống nhau và bạn gọi nó từ đối tượng của lớp con, như vậy khó có thể xác định được việc gọi phương thức của lớp A hay B.

Vì vậy lỗi khi biên dịch sẽ tốt hơn lỗi khi runtime, java sẽ print ra lỗi "compile time error" nếu bạn cố tình kế thừa 2 class.

```
class A {
    void msg() {
        System.out.println("Hello");
    }
}

class B {
    void msg() {
        System.out.println("Welcome");
    }
}

public class C extends A,B {
    public static void main(String args[]) {
        C obj = new C();
        obj.msg();
    }
}
```

Kết quả:

- Từ khóa this trong java
- Từ khóa super trong java
- Từ khóa final trong java
- Từ khóa static trong java
- Access Modifier trong Java
- Tính đồng gói trong java
- Tính kế thừa trong java
- Tính đa hình trong java



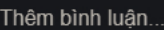

Bộ nhớ stack và heap trong java - học Java cơ bản đến nâng cao

viethuts.vn




Swing trong Java - Giới thiệu Java Swing

viettuts.vn

Sắp xếp theo **Cũ nhất** 

Jennifer Do
Cảm ơn các bài viết của anh chị nhiều lắm a ❤️


Trần Tấn Huy
 Dễ hiểu thật. Thank admin nhiều vì viết bài chất lượng như vậy.



Khải Nguyễn Quang

Có bài tập về kế thừa ko ad?

Thích · Phản hồi · 30 tuần



Plugin bình luận trên Facebook



Câu hỏi phỏng vấn

201 câu hỏi phỏng vấn java

25 câu hỏi phỏng vấn servlet

75 câu hỏi phỏng vấn jsp

52 câu hỏi phỏng vấn Hibernate

70 câu hỏi phỏng vấn Struts2

70 câu hỏi phỏng vấn Spring

57 câu hỏi phỏng vấn SQL

About VietTuts.Vn

Hệ thống bài học trên VietTuts.Vn bao gồm các bài lý thuyết và thực hành về các công nghệ java và công nghệ web. Các bài lý thuyết trên hệ thống VietTuts.Vn được tham khảo và tổng hợp từ các trang <http://javatpoint.com>, <http://www.tutorialspoint.com>, <http://docs.oracle.com/en> ...

