
**RoboForex**
WANT TO TRADE CRYPTOCURRENCIES?
CFD trading involves high risks, which can result in losing your entire deposit.



- 24/7 TRADING
- MT5, MT4, MOBILE
- LEVERAGE 1:10

TRADE NOW

Tính đa hình trong java

Overloading và overriding trong java

Lớp Object trong java

Đa hình trong java (Polymorphism) là một khái niệm mà chúng ta có thể thực hiện một hành động bằng nhiều cách khác nhau. Polymorphism được cấu tạo từ 2 từ Hy Lạp: poly và morphs. Trong đó “poly” có nghĩa là nhiều và “morphs” có nghĩa là hình thể. Vậy polymorphism có nghĩa là nhiều hình thể.

Có hai kiểu của đa hình trong java, đó là đa hình lúc biên dịch (compile) và đa hình lúc thực thi (runtime). Chúng ta có thể thực hiện đa hình trong java bằng cách nạp chồng phương thức và ghi đè phương thức.

Nếu bạn nạp chồng phương thức static trong java, đó là một ví dụ về đa hình lúc biên dịch. Trong bài này, chúng ta tập trung vào đa hình lúc runtime trong java.

Đa hình lúc runtime trong java

Đa hình lúc runtime là quá trình gọi phương thức đã được ghi đè trong thời gian thực thi chương trình. Trong quá trình này, một phương thức được ghi đè được gọi thông qua biến tham chiếu của một lớp cha.

Trước khi tìm hiểu về đa hình tại runtime, chúng ta cùng tìm hiểu về Upcasting.

Upcasting là gì?

Khi biến tham chiếu của lớp cha tham chiếu tới đối tượng của lớp con, thì đó là Upcasting. Ví dụ:

```
class A{}  
class B extends A{}
```

```
A a=new B();//upcasting
```

Ví dụ về đa hình lúc runtime trong java

Ví dụ 1:

Chúng ta tạo hai lớp Bike và Splendar. Lớp Splendar kế thừa lớp Bike và ghi đè phương thức run() của nó. Chúng ta gọi phương thức run bởi biến tham chiếu của lớp cha. Khi nó tham chiếu tới đối tượng của lớp

Recent Updates

- Hàm TimeValue() trong VBA
- Hàm TimeSerial() trong VBA
- Hàm Timer() trong VBA
- Hàm Time() trong VBA
- Hàm Second() trong VBA
- Hàm Minute() trong VBA
- Hàm Hour() trong VBA
- Hàm Now() trong VBA
- Hàm WeekDayName() trong VBA
- Hàm WeekDay() trong VBA
- Hàm MonthName() trong VBA
- Hàm Year() trong VBA

VietTuts on facebook



Học Java Cơ Bản Đến
2.142 lượt thích
it works..... Why?

Chúng ta gọi phương thức run bởi biến tham chiếu của lớp cha. Khi nó tham chiếu tới đối tượng của lớp con và phương thức lớp con ghi đè phương thức của lớp cha, phương thức lớp con được gọi lúc runtime.

Khi việc gọi phương thức được quyết định bởi JVM chứ không phải Compiler, vì thế đó là đa hình lúc runtime.

```
class Bike {
    void run() {
        System.out.println("running");
    }
}

public class Splender extends Bike {
    void run() {
        System.out.println("running safely with 60km");
    }

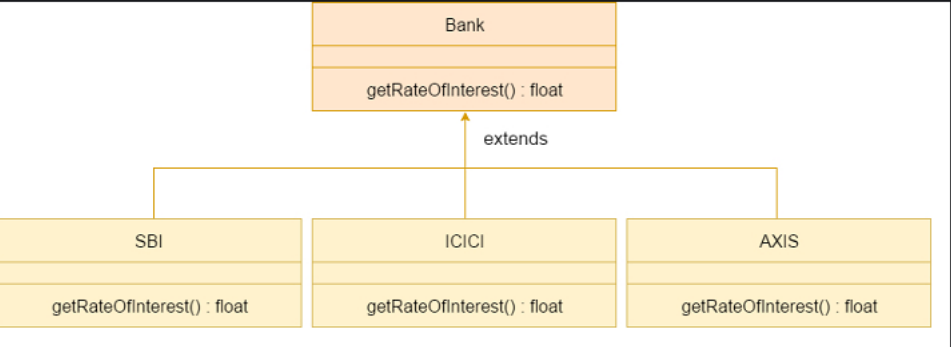
    public static void main(String args[]) {
        Bike b = new Splender();// upcasting
        b.run();
    }
}
```

Kết quả:

running safely with 60km


Ví dụ 2: Bank:


Giả sử Bank là một lớp cung cấp chức năng xem thông tin tỷ lệ lãi suất. Nhưng mỗi ngân hàng có một lãi suất khác nhau, ví dụ các ngân hàng SBI, ICICI và AXIS có tỷ lệ lãi suất lần lượt là 8%, 7% và 9%.



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

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

 Thích Trang

 Gửi tin nhắn

1 người bạn thích nội dung này



```

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

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

public class Test2 {
    public static void main(String args[]) {
        Bank b;
        b = new SBI();
        System.out.println("SBI Rate of Interest: " + b.getRateOfInterest());
        b = new ICICI();
        System.out.println("ICICI Rate of Interest: " + b.getRateOfInterest());
        b = new AXIS();
        System.out.println("AXIS Rate of Interest: " + b.getRateOfInterest());
    }
}

```

Kết quả:

```

SBI Rate of Interest: 8
ICICI Rate of Interest: 7
AXIS Rate of Interest: 9

```

Ví dụ 3: Shape:

```

class Shape {
    void draw() {
        System.out.println("drawing...");
    }
}

class Rectangle extends Shape {
    void draw() {
        System.out.println("drawing rectangle...");
    }
}

class Circle extends Shape {
    void draw() {
        System.out.println("drawing circle...");
    }
}

class Triangle extends Shape {
    void draw() {
        System.out.println("drawing triangle...");
    }
}

class TestPolymorphism2 {
    public static void main(String args[]) {
        Shape s;
        s = new Rectangle();
        s.draw();
        s = new Circle();
        s.draw();
    }
}

```


```
        s = new Triangle();  
        s.draw();  
    }  
}
```

Kết quả:

```
drawing rectangle...  
drawing circle...  
drawing triangle...
```

Đa hình lúc runtime trong Java với thành viên dữ liệu

Phương thức bị ghi đè không là thành viên dữ liệu, vì thế đa hình tại runtime không thể có được bởi thành viên dữ liệu. Trong ví dụ sau đây, cả hai lớp có một thành viên dữ liệu là speedlimit, chúng ta truy cập thành viên dữ liệu bởi biến tham chiếu của lớp cha mà tham chiếu tới đối tượng lớp con. Khi chúng ta truy cập thành viên dữ liệu mà không bị ghi đè, thì nó sẽ luôn luôn truy cập thành viên dữ liệu của lớp cha.

 Quy tắc: Đa hình lúc runtime không thể xảy ra với thành viên dữ liệu.

```
class Bike{  
    int speedlimit=90;  
}  
class Honda3 extends Bike{  
    int speedlimit=150;  
  
    public static void main(String args[]){  
        Bike obj=new Honda3();  
        System.out.println(obj.speedlimit);//90  
    }  
}
```

Kết quả:

```
90
```

Đa hình lúc runtime trong Java với kế thừa nhiều tầng

Ví dụ 1:

```
class Animal {  
    void eat() {  
        System.out.println("eating");  
    }  
}  
  
class Dog extends Animal {  
    void eat() {  
        System.out.println("eating fruits");  
    }  
}  
  
class BabyDog extends Dog {  
    void eat() {  
        System.out.println("drinking milk");  
    }  
}
```

```
        System.out.println( "drinking milk" );
    }

    public static void main(String args[]) {
        Animal a1, a2, a3;
        a1 = new Animal();
        a2 = new Dog();
        a3 = new BabyDog();
        a1.eat();
        a2.eat();
        a3.eat();
    }
}
```

Kết quả:

```
eating
eating fruits
drinking Milk
```

Ví dụ 2:

```
class Animal {
    void eat() {
        System.out.println("animal is eating...");
    }
}

class Dog extends Animal {
    void eat() {
        System.out.println("dog is eating...");
    }
}

class BabyDog1 extends Dog {
    public static void main(String args[]) {
        Animal a = new BabyDog1();
        a.eat();
    }
}
```

Kết quả:

```
Dog is eating
```

Vì BabyDog1 không ghi đè phương thức eat(), nên phương thức eat() của lớp Dog được gọi.

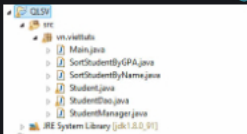
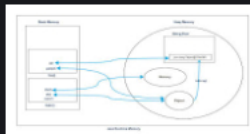
🕒 Overloading và overriding trong java

Lớp Object trong java 🕒

Bài viết liên quan:

- 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



Free 2019 Tarot Reading

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

Cập Nhật Kiến Thức Thời Kỳ 4.0

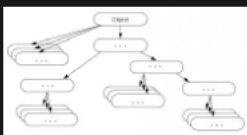
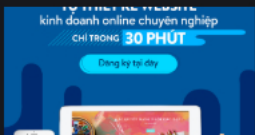
Bài tập quản lý sinh viên trong java - bài tập java có lời giải

QC AstroMary

viettuts.vn

QC FTU - TOPICA

viettuts.vn



Tạo Website Online Nhanh

Kiến trúc Spring

Interface trong java - học Java cơ bản đến nâng cao

Lớp Object trong java - học Java miễn phí hay nhất

QC haravan.com

viettuts.vn

viettuts.vn

viettuts.vn



0 bình luận

Sắp xếp theo Cũ nhất



Thêm bình luận...

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

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

[70 câu hỏi phỏng vấn Struts2](#)

[70 câu hỏi phỏng vấn Spring](#)

[57 câu hỏi phỏng vấn SQL](#)

VietTuts.Vn được tham khảo và tổng

hợp từ các trang <http://javatpoint.com>,

<http://www.tutorialspoint.com>,

<http://docs.oracle.com/en> ...

