

Từ khóa throw trong Java

Từ khóa **throw** trong Java được sử dụng để ném tường minh một exception. Chúng ta có thể ném hoặc Checked Exception hoặc Unchecked Exception trong Java bởi từ khóa throw. Từ khóa throw được sử dụng chủ yếu để ném các Custom Exception (là các exception được định nghĩa bởi lập trình viên). Chúng ta sẽ tìm hiểu về Custom Exception trong chương riêng.

Cú pháp của từ khóa throw trong Java như sau:

```
throw exception;
```

Cùng theo dõi ví dụ đơn giản để ném IOException:

```
throw new IOException("Xin loi: device error");
```

Ví dụ từ khóa throw trong Java

Trong ví dụ này, chúng ta đã tạo phương thức validate mà nhận giá trị nguyên như là một tham số. Nếu **tuoi** là nhỏ hơn 18, chúng ta ném ArithmeticException, nếu không chúng ta sẽ in một thông báo là: Chao mung ban den bo phieu.

```
public class TestThrow1{  
    static void validate(int tuoi){  
        if(tuoi<18)  
            throw new ArithmeticException("Khong hop le");  
        else  
            System.out.println("Chao mung ban den bo phieu");  
    }  
    public static void main(String args[]){  
        validate(13);  
        System.out.println("Phan code con lai...");  
    }  
}
```

Chương trình Java trên sẽ cho kết quả:

```
Exception in thread "main" java.lang.ArithmeticException: Khong hop le
```

Quá trình lan truyền Exception trong Java

Đầu tiên, một exception được ném từ đầu Stack và nếu nó không được bắt, nó chuyển xuống dưới Call Stack tới phương thức trước đó, nếu tại đây nó không được bắt, thì exception lại di chuyển xuống dưới tới phương thức trước đó, ... tới khi nó được bắt hoặc tới khi nó chạm đáy của Call Stack. Đây là quá trình lan truyền exception.

Qui tắc: Theo mặc định, Unchecked Exception được lan truyền trong Calling Chain.

Ví dụ về quá trình lan truyền của exception:

```
class TestExceptionPropagation1{
    void m(){
        int data=50/0;
    }
    void n(){
        m();
    }
    void p(){
        try{
            n();
        }catch(Exception e){System.out.println("Exception duoc xu ly");}
    }
    public static void main(String args[]){
        TestExceptionPropagation1 obj=new TestExceptionPropagation1();
        obj.p();
        System.out.println("Luong chuan...");
    }
}
```

Chương trình Java trên sẽ cho kết quả:

```
Exception duoc xu ly
Luong chuan...
```

Trong ví dụ trên, exception xuất hiện trong phương thức m() nơi nó không được xử lý, vì thế nó được lan truyền tới phương thức n() trước đó, tại đây nó không được xử lý, và tiếp tục nó được truyền tới phương thức p(), tại đây exception này được xử lý.

Exception có thể được xử lý trong bất cứ phương thức nào trong Call Stack hoặc trong phương thức main(), phương thức p(), phương thức n() hoặc phương thức m().

Qui tắc: Theo mặc định, Checked Exception không được lan truyền trong Calling Chain.

Chương trình ví dụ miêu tả Checked Exception không được lan truyền:

```
class TestExceptionPropagation2{
    void m(){
        throw new java.io.IOException("device error");//checked exception
    }
    void n(){
        m();
    }
    void p(){
        try{
            n();
        }catch(Exception e){System.out.println("Exception duoc xu ly");}
    }
    public static void main(String args[]){
        TestExceptionPropagation2 obj=new TestExceptionPropagation2();
        obj.p();
        System.out.println("Luong chuan");
    }
}
```

Chạy chương trình Java trên sẽ cho một lỗi Compile Time Error.