

Khối finally trong Java

Khối finally trong là một khối được sử dụng để thực thi các phần code quan trọng như đóng kết nối, đóng stream, ... Khối finally luôn luôn được thực thi dù cho exception có được xử lý hay không. Khối finally phải được theo sau bởi khối try hoặc khối catch.

Ghi chú: nếu bạn không xử lý ngoại lệ trước khi kết thúc chương trình, JVM sẽ thực thi khối finally (nếu bạn không có).

Sự sử dụng khối finally trong Java

Bạn theo dõi các tình huống khác nhau mà khối try trong Java có thể được sử dụng.

Tình huống 1: ở đây exception không xuất hiện

```
class TestFinallyBlock{
    public static void main(String args[]){
        try{
            int data=25/5;
            System.out.println(data);
        }
        catch(NullPointerException e){System.out.println(e);}
        finally{System.out.println("Khoi finally luon luon duoc thuc thi");}
        System.out.println("Phan code con lai...");
    }
}
```

Chương trình trên sẽ cho kết quả:

```
5
Khoi finally luon luon duoc thuc thi
Phan code con lai...
```

Tình huống 2: exception xuất hiện và không được xử lý.

```
class TestFinallyBlock1{
    public static void main(String args[]){
        try{
            int data=25/0;
```

```
    System.out.println(data);
}
catch(NullPointerException e){System.out.println(e);}
finally{System.out.println("Khoi finally luon luon duoc thuc thi");}
System.out.println("Phan code con lai...");
}
}
```

Chương trình trên sẽ cho kết quả:

```
Khoi finally luon luon duoc thuc thi
Exception in thread "main" java.lang.ArithmeticException: / by zero
```

Tình huống 3: exception xảy ra và được xử lý.

```
public class TestFinallyBlock2{
    public static void main(String args[]){
        try{
            int data=25/0;
            System.out.println(data);
        }
        catch(ArithmeticException e){System.out.println(e);}
        finally{System.out.println("Khoi finally luon luon duoc thuc thi");}
        System.out.println("Phan code con lai...");
    }
}
```

Chương trình trên sẽ cho kết quả:

```
java.lang.ArithmeticException: / by zero
Khoi finally luon luon duoc thuc thi
Phan code con lai...
```

Qui tắc: Với một khối try thì có thể có 0 hoặc nhiều khối catch, nhưng chỉ có một khối finally.

Ghi chú: Khối finally sẽ không được thực thi nếu chương trình thoát ra (hoặc bởi gọi System.exit()) hoặc bởi gây ra một lỗi nghiêm trọng Fatal Error mà làm ngừng tiến trình).

Phân biệt final, finally, và finalize trong Java

Bảng dưới đây liệt kê và miêu tả các điểm khác nhau giữa final, finally và finalize trong Java:

final	finally	finalize
Final được sử dụng để áp dụng các ràng buộc trên lớp, phương thức và biến. Lớp final không thể được kế thừa, phương thức final không thể bị ghi đè, và giá trị biến final không thể bị thay đổi	Finally được sử dụng để đặt các code quan trọng, nó sẽ được thực thi dù exception có được xử lý hay không	Finalize được sử dụng để thực hiện tiến trình dọn sạch
Final là một từ	Finally là một khối	Finalize là một phương thức

khóa		
------	--	--

Ví dụ final trong Java

```
class ViduFinal{
public static void main(String[] args){
final int x=100;
x=200;//Compile Time Error
}}
```

Ví dụ finally trong Java

```
class ViduFinally{
public static void main(String[] args){
try{
int x=300;
}catch(Exception e){System.out.println(e);}
finally{System.out.println("Khoi finally duoc thuc thi");}
}}
```

Ví dụ finalize trong Java

```
class ViduFinalize{
public void finalize(){System.out.println("finalize duoc goi");}
public static void main(String[] args){
ViduFinalize f1=new ViduFinalize();
ViduFinalize f2=new ViduFinalize();
f1=null;
f2=null;
System.gc();
}}
```