

ExceptionHandling và Ghi đè phương thức trong Java

Có một số qui tắc bạn nên biết khi nói đến ghi đè phương thức với xử lý ngoại lệ. Đó là:

Nếu phương thức của lớp cha không khai báo một exception: phương thức ghi đè của lớp con không thể khai báo Checked Exception nhưng có thể khai báo Unchecked Exception.

Qui tắc 1: Nếu phương thức của lớp cha không khai báo một exception: phương thức ghi đè của lớp con không thể khai báo Checked Exception.

```
import java.io.*;

class Parent{

    void msg(){System.out.println("parent");}

}

class TestExceptionChild extends Parent{

    void msg()throws IOException{

        System.out.println("TestExceptionChild");

    }

    public static void main(String args[]){

        Parent p=new TestExceptionChild();

        p.msg();

    }

}
```

Chạy chương trình sẽ cho Compile Time Error.

Qui tắc 2: Nếu phương thức của lớp cha không khai báo một exception: phương thức ghi đè của lớp con không thể khai báo Checked Exception nhưng có thể khai báo Unchecked Exception.

```
import java.io.*;

class Parent{

    void msg(){System.out.println("parent");}

}
```

```
class TestExceptionChild1 extends Parent{
    void msg()throws ArithmeticException{
        System.out.println("child");
    }
    public static void main(String args[]){
        Parent p=new TestExceptionChild1();
        p.msg();
    }
}
```

Chạy chương trình sẽ cho kết quả:

```
child
```

Nếu phương thức lớp cha khai báo một exception: phương thức ghi đè của lớp con có thể khai báo cùng exception đó, exception của lớp con, hoặc không khai báo exception nào, nhưng không thể khai báo exception cha.

1. Ví dụ trường hợp phương thức ghi đè của lớp con khai báo exception cha.

```
import java.io.*;
class Parent{
    void msg()throws ArithmeticException{System.out.println("parent");}
}

class TestExceptionChild2 extends Parent{
    void msg()throws Exception{System.out.println("child");}

    public static void main(String args[]){
        Parent p=new TestExceptionChild2();
        try{
            p.msg();
        }catch(Exception e){}
    }
}
```

```
}
```

Chạy chương trình sẽ cho Compile Time Error.

2. Ví dụ trường hợp phương thức ghi đè của lớp con khai báo cùng exception đó.

```
import java.io.*;

class Parent{

    void msg()throws Exception{System.out.println("parent");}

}

class TestExceptionChild3 extends Parent{

    void msg()throws Exception{System.out.println("child");}

    public static void main(String args[]){

        Parent p=new TestExceptionChild3();

        try{

            p.msg();

        }catch(Exception e){}

    }

}
```

Chạy chương trình sẽ cho kết quả:

```
child
```

3. Ví dụ trường hợp phương thức ghi đè của lớp con khai báo exception của lớp con.

```
import java.io.*;

class Parent{

    void msg()throws Exception{System.out.println("parent");}

}

class TestExceptionChild4 extends Parent{

    void msg()throws ArithmeticException{System.out.println("child");}

}
```

```
public static void main(String args[]){
    Parent p=new TestExceptionChild4();
    try{
        p.msg();
    }catch(Exception e){}
}
```

Chạy chương trình sẽ cho kết quả:

```
child
```

3. Ví dụ trường hợp phương thức ghi đè của lớp con không khai báo exception nào.

```
import java.io.*;
class Parent{
    void msg()throws Exception{System.out.println("parent");}
}

class TestExceptionChild5 extends Parent{
    void msg(){System.out.println("child");}

    public static void main(String args[]){
        Parent p=new TestExceptionChild5();
        try{
            p.msg();
        }catch(Exception e){}
    }
}
```

Chạy chương trình sẽ cho kết quả:

```
child
```