# Task5: Xử lý ngoại lệ khi nhập dữ liệu

## 1. Khi nào cần sử dụng try catch trong Java

Khi thực thi một đoạn code Java nào đó, các lỗi khác nhau có thể xảy ra như:

* Lỗi do chính coder tạo ra
* Lỗi cú pháp
* Lỗi logic
* ... những điều mà chúng ta có thể không lường trước.

Ví dụ như:

* Người dùng nhập dữ liệu không hợp lệ
* Truy cập vượt quá chỉ số mảng
* Chia một số cho 0
* ……

Khi xảy ra lỗi, ngoại lệ, Java thông thường sẽ dừng thực thi chương trình và đưa ra một thông báo, hay nói cách khác là Java ném ra một exception – Ngoại lệ.  
  
  
Xử lý ngoại lệ (Exception Handling) trong java là một cơ chế xử lý các lỗi runtime để có thể duy trì luồng bình thường của ứng dụng.  
  
  
Quá trình xử lý exception được gọi là catch exception (bắt ngoại lệ), nếu Runtime System không xử lý được ngoại lệ thì chương trình sẽ kết thúc.  
  
  
Khối lệnh try trong java được sử dụng để chứa một đoạn code thực thi mà có thế trong quá trình thực thi nó sẽ xảy ra một ngoại lệ.  
  
  
Sau một khối lệnh try, bạn phải khai báo khối lệnh catch hoặc finally, hoặc cả hai.  
  
  
Khối catch trong java được sử dụng để xử lý nếu xảy ra Exception, nếu không thì nó bị bỏ qua.  
  
  
Khối catch phải được sử dụng ngay sau khối try. Bạn có thể sử dụng nhiều khối catch với nhưng chỉ có một khối try duy nhất.

### **Cú pháp của try catch trong Java:**

try{

//code có thể ném ra ngoại lệ

}catch(Exception\_class\_name ex){  
 //code xử lí ngoại lệ

}

### **Cú pháp của try finally trong Java**

try{

//code có thể ném ra ngoại lệ

}finally{

// code trong khối này có thể được thực thi luôn

}

### **Cú pháp của try catch finally trong Java**

Try{

}catch(Exception\_class\_Name\_1 ex){

//code xử lí ngoại lệ 1

}catch(Exception\_class\_Name\_2 ex) {

//code xử lí ngoại lệ 2

}catch(Exception\_class\_Name\_n ex){

//code xử lí ngoại lệ n

}finnaly{

//code trong khối này luôn được thực thi

}

## 2 kiểu ngoại lệ trong Java

Có 2 kiểu ngoại lệ trong Java:

* **Checked Exceptions**: Các ngoại lệ này thường là bị buộc phải bắt hoặc khai báo. Nếu quy tắc này không được tuân theo thì trình biên dịch sẽ không thực thi chương trình.

* **Unchecked Exceptions**: Ngoại lệ này thường là do viết code sai, truyền đối null hoặc tham số không chính xác...

### **Các ngoại lệ kiểu Checked Exceptions phổ biến:**

* **IOException**: Ngoại lệ liên quan đến file input / output
* **SQLException**: Ngoại lệ liên quan đến cú pháp SQL
* **DataAccessException**: Ngoại lệ liên quan đến việc truy cập CSDL
* **ClassNotFoundException**: Bị ném khi JVM không thể tìm thấy một lớp mà nó cần, do lỗi dòng lệnh, sự cố đường dẫn hoặc tệp, class bị thiếu...
* **InstantiationException**: Ngoại lệ khi cố gắng tạo đối tượng của một abstract class hoặc interface

### **Các ngoại lệ kiểu Unchecked Exceptions phổ biến:**

* **NullPointerException**: Ngoại lệ bị ném ra khi cố gắng truy cập một đối tượng có biến tham chiếu có giá trị hiện tại là null
* **ArrayIndexOutOfBound**: Ngoại lệ khi cố gắng truy cập một phần tử vượt quá độ dài của mảng
* **IllegalArgumentException**: Ngoại lệ bị ném ra khi một phương thức nhận được một đối số được định dạng khác với phương thức mong đợi.
* **IllegalStateException**: Ngoại lệ bị ném ra khi trạng thái của môi trường không phù hợp với hoạt động cố gắng thực hiện, ví dụ: Sử dụng Scanner đã bị đóng.
* **NumberFormatException**: Ngoại lệ bị ném khi một phương thức chuyển đổi một Chuỗi thành số nhưng không thể chuyển đổi.
* **ArithmeticException**: Lỗi số học, chẳng hạn như chia cho 0.

Trên đây là cú pháp sử dụng try catch, và các ngoại lệ phổ biến. Còn sử dụng cụ thể như thế nào thì mời bạn sang tiếp phần tiếp theo.

## **2. Ví dụ về cách sử dụng try catch trong java**

Ở phần này mình sẽ giới thiệu cho các bạn một số ví dụ để các bạn làm quen với **try catch trong Java** nhé.  
  
  
Tuy nhiên, đầu tiên, nếu chúng ta không sử dụng try catch, hãy thử chạy một đoạn code lỗi sau, xem trình biên dịch sẽ báo như thế nào nhé.

Public class TrycatchDemo{

Public static void main(String[] args){

Int data =5 /0;

System.our.printlln(“Phép chia cho 0”);

}

}

### Trình biên dịch đã dừng lại ngay tại dòng int data = 5 / 0 và ném ra một ngoại lệ (exception) ở trong màn hình console. Lúc này dòng System.out.println("Phép chia cho 0"); chưa được thực thi.