



Xử lý ngoại lệ



Xử lý lỗi & ngoại lệ

✦ Cách xử lý lỗi trong một số ngôn ngữ lập trình:

```
If (<mẫu số> != 0){  
    // Thực hiện chia  
}else{  
    // Xử lý lỗi  
}
```

```
If(<chỉ số i> hợp lệ){  
    // Truy cập phần tử mảng tại chỉ số i  
}else{  
    // Xử lý lỗi  
}
```



Xử lý lỗi & ngoại lệ

✦ Cách xử lý lỗi trong Java:

```
try{  
    // Thực hiện chia  
    // Truy cập phần tử mảng tại chỉ số i  
}catch(<Ngoại lệ chia cho 0>){  
    // Xử lý lỗi  
}catch(<Ngoại lệ truy cập mảng ngoài chỉ số>){  
    // Xử lý lỗi  
}
```



Xử lý lỗi & ngoại lệ

- ✦ Hai loại lỗi có thể xảy ra trong chương trình là lỗi biên dịch (compile error) và lỗi thực thi (runtime error).
- ✦ Lỗi biên dịch thường xuất phát từ mã nguồn, các lỗi sai cú pháp là một ví dụ. Lỗi biên dịch được bắt bởi trình biên dịch tại thời điểm dịch chương trình. Khi tất cả các lỗi biên dịch đã được sửa mã nguồn mới được dịch sang mã máy hoặc mã bytecode trong Java.
- ✦ Lỗi thực thi là các lỗi phát sinh khi chương trình đang chạy, trình biên dịch không thể bắt các lỗi này tại thời điểm biên dịch như lỗi tràn bộ nhớ, truy cập ngoài chỉ số mảng... Một lỗi thực thi nếu không được người lập trình lường trước và xử lý sẽ khiến chương trình dừng đột ngột.



Khối Try/catch/finally

```
try {  
    // Code có thể phát sinh một exception  
} [catch (Loại Exception-1 e) {  
    // Xử lý lỗi ứng với ngoại lệ 1  
}] [catch (Loại Exception-2 e) {  
    // Xử lý lỗi ứng với ngoại lệ 2  
}]  
...  
} [catch (Loại Exception-n e) {  
    // Xử lý lỗi ứng với ngoại lệ n  
}]  
[finally {  
    // Phần lệnh luôn thực hiện dù ngoại lệ có phát sinh  
}]
```



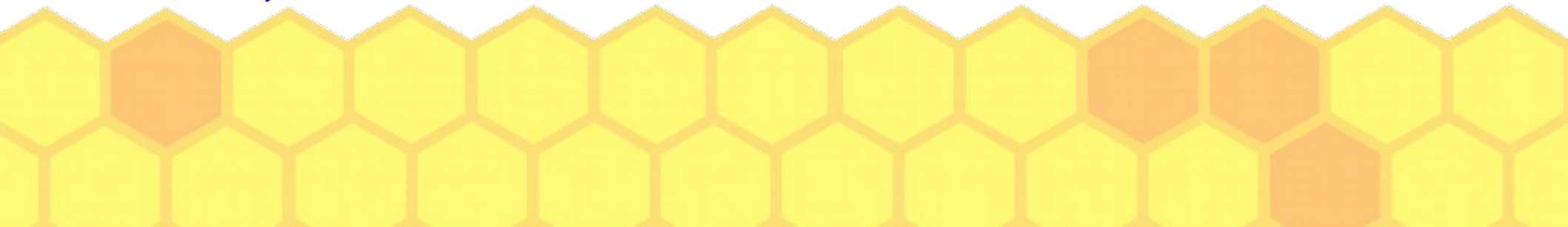
Ví dụ

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    try {  
        System.out.print("Nhập a = ");  
        int a = sc.nextInt();  
        System.out.print("Nhập b = ");  
        int b = sc.nextInt();  
        int c = a/b;  
        System.out.print("Thương c = " + c);  
    } catch (Exception e) {  
        System.out.println("Lỗi phát sinh: " + e.getMessage());  
    }  
}
```



Ví dụ

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    try {  
        System.out.print("Nhap a = ");  
        int a = sc.nextInt();  
        System.out.print("Nhap b = ");  
        int b = sc.nextInt();  
        int c = a/b;  
        System.out.print("Thuong c = " + c);  
    } catch (InputMismatchException e) {  
        System.out.println("Lỗi nhập dữ liệu");  
    } catch (ArithmeticException e) {  
        System.out.println("Lỗi chia cho 0");  
    }  
}
```



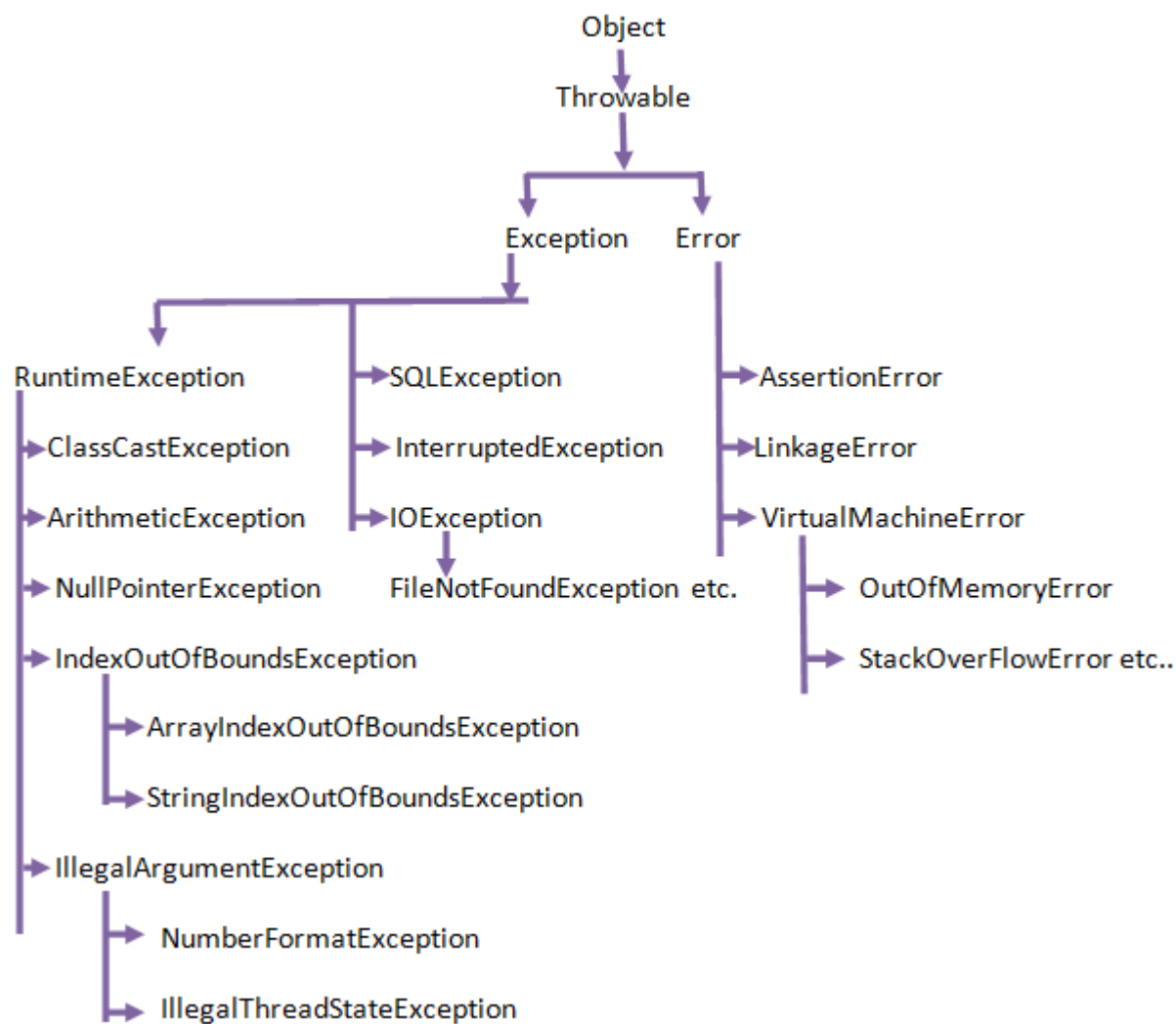
Ví dụ

- +Sử dụng try không catch. Vì một trong hai khối catch/finally phải tồn tại nên điều này cũng có nghĩa là ta sử dụng mẫu try/finally.
- +Mẫu này sử dụng khi ta không cần quan tâm đến vấn đề xử lý ngoại lệ mà chỉ quan tâm đến các hoạt động dọn dẹp diễn ra trong khối finally.
- +Khối finally được thực hiện ngay cả khi trong khối try hoặc catch có chứa lệnh ngắt như return, break, continue, đó là sự khác biệt so với các lệnh khác nằm sau khối try/catch/finally.

```
Connection connection = null;  
try {  
    // mở connection tới database  
    // sử dụng kết nối connection để thực hiện một số tác vụ  
} finally {  
    if (connection != null) {  
        // đóng kết nối  
    }  
}
```



Các lớp ngoại



Ba loại ngoại lệ

- ✦ *Có hai loại ngoại lệ chính là: checked và unchecked. Còn Sun Microsystems nói rằng có ba loại ngoại lệ: Checked Exception, Unchecked Exception, Error*
- ✦ *Ngoại lệ checked*
 - ~ *Các lớp kế thừa từ lớp Throwable ngoại trừ RuntimeException và Error được gọi là ngoại lệ checked (checked exception), ví dụ như Exception, IOException, SQLException,...*
 - ~ *Các ngoại lệ checked được kiểm tra tại thời điểm dịch mã nguồn chương trình.*



✦ Ngoại lệ checked

✦ Ví dụ:

- ~ *FileNotFoundException là một loại của ngoại lệ IOException.*
- ~ *Trong ví dụ sau, dù file “input.txt” có tồn tại hay không, chương trình dịch luôn yêu cầu người lập trình phải xử lý ngoại lệ FileNotFoundException theo hai cách:*



✦ Ngoại lệ checked

✦ **Bắt ngoại lệ sử dụng try/catch**

```
public static void main(String[] args)
{
    FileNotFoundExceptionExample exam = new
    FileNotFoundExceptionExample();
    exam.checkFileNotFoundException();
}
}
```

```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
public class FileNotFoundExceptionExample
{
    public void checkFileNotFoundException()
    {
        try
        {
            FileInputStream in = new
            FileInputStream("input.txt");
            System.out.println("This is not printed");
        }
        catch (FileNotFoundException e)
        {
            e.printStackTrace();
        }
    }
}
```



✦ Ngoại lệ checked

✦ ***Khai báo khả năng ném ra ngoại lệ ở phương thức theo mẫu:***

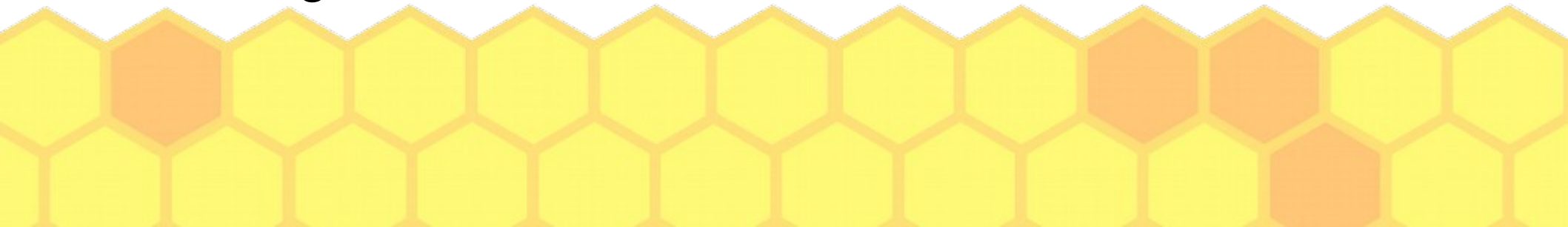
~ ***<tên phương thức>
throws <Ngoại lệ
1>[, <Ngoại lệ 2>,...]***

```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
public class Test {
    public void checkFileNotFound() throws
    FileNotFoundException {
        FileInputStream in = new
        FileInputStream("input.txt");
        System.out.println("This is not printed");
        111
    }
    public static void main(String[] args) throws
    FileNotFoundException {
        Test example = new Test();
        example.checkFileNotFound();
    }
}
```



Ngoại lệ unchecked

- ✦ Các lớp kế thừa từ RuntimeException được gọi là ngoại lệ unchecked (unchecked exception).
- ✦ Ví dụ:
 - ~ ArithmeticException,
 - ~ NullPointerException,
 - ~ ArrayIndexOutOfBoundsException,...
- ✦ Các ngoại lệ unchecked không được kiểm tra tại thời điểm dịch, chúng được kiểm tra tại thời điểm thực thi chương trình.



Error

- ✦ Các ngoại lệ thuộc loại error là những lỗi nghiêm trọng và không dự đoán trước được như ThreadDead, LinkageError,
- ✦ VirtualMachineError...
- ✦ Các ngoại lệ kiểu Error ít được xử lý.



Một số lớp ngoại lệ

✦ Lớp Throwable

~ Lớp Throwable là lớp cha của tất cả các lớp ngoại lệ, nó lưu thông tin chi tiết về ngoại lệ đã xảy ra và cung cấp một số phương thức cơ bản dùng chung cho các lớp ngoại lệ khác như:

- ✦ `Throwable(String s);` // Tạo một đối tượng ngoại lệ kèm thông tin chi tiết
- ✦ `String getMessage();` // Lấy thông tin về ngoại lệ
- ✦ `void printStackTrace();` // In ra tất cả các thông tin liên quan đến ngoại lệ



Một số lớp ngoại lệ

✦ Lớp Exception

- ~ Lớp Exception là lớp cha của các lớp thuộc nhóm RuntimeException và nhóm ngoại lệ checked. Người dùng có thể tạo ra các ngoại lệ kế thừa từ lớp Exception.

✦ Lớp Error

- ~ Lớp Error là lớp cha của các ngoại lệ thuộc nhóm error.



Một số lớp ngoại lệ

✦ Lớp RuntimeException

~ Lớp RuntimeException là lớp cha của các ngoại lệ unchecked. Một số lớp ngoại lệ hay xảy ra thuộc lớp này gồm:

- ✦ ArithmeticException: xảy ra nếu ta chia bất kỳ số nào cho số 0
- ✦ NullPointerException: xảy ra nếu ta có bất kỳ biến nào có giá trị null, thực hiện bất kỳ hoạt động nào bởi biến đó

✦ Ví dụ:

~ `String s=null;`

~ `System.out.println(s.length()); // NullPointerException`



Một số lớp ngoại lệ

- + Lớp RuntimeException

- + Ví dụ:

- ~ NumberFormatException: xảy ra nếu định dạng của giá trị sai

- + Ví dụ:

- ~ String s="abc";
 - ~ int i=Integer.parseInt(s); //NumberFormatException

- ~ ArrayIndexOutOfBoundsException: xảy ra nếu chỉ số truy cập mảng nằm ngoài dải cho phép.

- + Ví dụ:

- ~ int a[]=new int[5];
 - ~ a[5]=50;
 - ~ //ArrayIndexOutOfBoundsException



XÂY DỰNG LỚP NGOẠI LỆ

- ✦ Người lập trình có thể tự định nghĩa lớp ngoại lệ để phù hợp với các tình huống riêng của chương trình bằng cách kế thừa từ một lớp ngoại lệ nào đó, việc cài đặt nội dung thế nào tùy thuộc vào người lập trình.
- ✦ Ví dụ:
 - ~ Trong lớp sinh viên ta cần viết phương thức cho phép thay đổi giá trị của thuộc tính giới tính (gender). Giới tính chỉ có thể nhận một trong hai giá trị kiểu ký tự 'm' (male) hoặc 'f' (female), nếu giá trị truyền vào phương thức sai sẽ phát sinh một ngoại lệ.



XÂY DỰNG LỚP NGOẠI LỆ

✦ Ví dụ:

```
package p1;  
public class InvalidGenderException extends  
Exception{  
    public InvalidGenderException(String msg) {  
        super(msg);  
    }  
}
```



XÂY DỰNG LỚP NGOẠI LỆ

+ Ví dụ:

```
package p1;
...
public class Student {
...
public void setGender(char gender)
throws
InvalidGenderException{
if(gender != 'm' || gender != 'f') {
throw new InvalidGenderException("Ky tu
gioi tinh khong
hop le");
}
this.gender = gender;
}
```

```
public static void main(String[]
args) {
Student std = new Student();
try {
std.setGender('g');
} catch (InvalidGenderException
e) {
System.out.println(e.getMessage
());
}
}
}
```



LAN TRUYỀN NGOẠI LỆ

LAN TRUYỀN NGOẠI LỆ

