BÀI GIẢNG

NGÔN NGỮ LẬP TRÌNH JAVA

CHƯƠNG 9:

XỬ LÝ NGOẠI LỆ

TS. Nguyễn Sĩ Thìn
(nsthin@vku.udn.vn)

Mục tiêu

Hiểu mục đích, ý nghĩa của việc xử lý ngoại lệ

Hiểu và phân loại được các loại ngoại lệ và cách xử lý ngoại lệ

Vận dụng được các kỹ thuật xử lý ngoại lệ

Nội dung

Chương 9

1. Giới thiệu
2. Xử lý ngoại lệ
3. “throw” và “throws”

1. Giới thiệu

- **Ngoại lệ (Exception):** Một sự kiện/ lỗi làm gián đoạn tiến trình thực hiện một chương trình
- **Xử lý ngoại lệ (Exeption handling):** cơ chế quản lý các ngoại lệ để chương trình có thể tiếp tục thực hiện hoặc kết thúc một cách an toàn, không đột ngột.

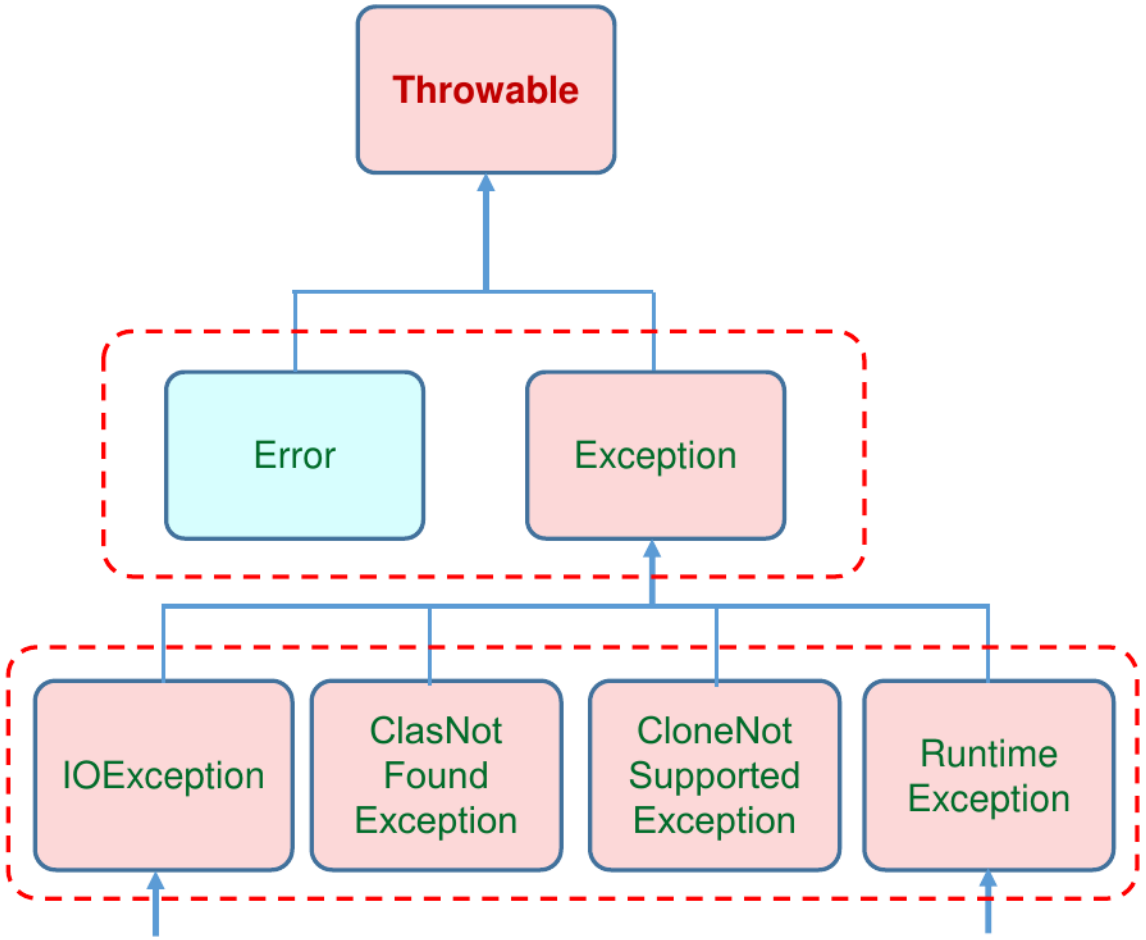
- **Ví dụ:**

```
public static void main(String[] args) {
int i=6/0;
System.out.println("Tiep tuc thuc hien chuong trinh");
}
```

```
Exception in thread "main" java.lang.ArithmeticException: / by zero
at chuong9.Exception.main(Exception.java:6)
```

Phân loại

- **Lớp "Throwable":** tất cả các lớp ngoại lệ được xuất phát từ lớp "**Throwable**"



- **Phân thành 3 loại:**
  - **Checked exeption:**
    - Được đưa ra lúc biên dịch
    - VD: IOException, SQLException,…
  - **Unchecked exeption:**
    - Được đưa ra lúc chạy chương trình
    - VD: NullPointerException,
  - **Error:**
    - Không thể tự khắc phục
    - VD: OutOfMemoryError,…

Một số trường hợp xảy ra "Unchecked exception"

- **Xử lý phép toán:**
  - ArithmeticException
  - Ví dụ: `int a=50/0;`
- **Xử lý con trỏ**
  - NullPointerException
  - Ví dụ:
```
String s=null;
System.out.println(s.length());
```
- **Định dạng số**
  - NumberFormatException
  - Ví dụ:
```
String s="abc";
int i=Integer.parseInt(s);
```
- **Xử lý mảng**
  - ArrayIndexOutOfBoundsException
  - Ví dụ:
```
int a[]=new int[5];
a[10]=50;
```

2. Xử lý ngoại lệ

- **Vấn đề:** Điều gì xảy ra khi chương trình đưa ra/ xuất hiện ngoại lệ nhưng ngoại lệ này không được xử lý ?
  - Chương trình kết thúc và xuất hiện thông tin ngoại lệ trên màn hình console

    → Giải pháp: Xử lý (bắt lấy) ngoại lệ để chương trình được tiếp tục an toàn
- **Cách thức xử lý ngoại lệ**
  - Khối lệnh **“ try / catch ”**
  - Khối lệnh **“try/ catch”** kết hợp **“finally”**

Khối lệnh “try / catch”

- **Khối lệnh “try”**
  - Chứa đoạn lệnh có thể đưa ra ngoại lệ
  - Nếu ngoại lệ xảy ra thì các lệnh tiếp theo của khối lệnh try sẽ bị bỏ qua
- **Khối lệnh “catch”**
  - Xử lý/ bắt lấy ngoại lệ tương ứng xảy ra ở khối lệnh try
  - Gồm một tham số
- **Cú pháp:**

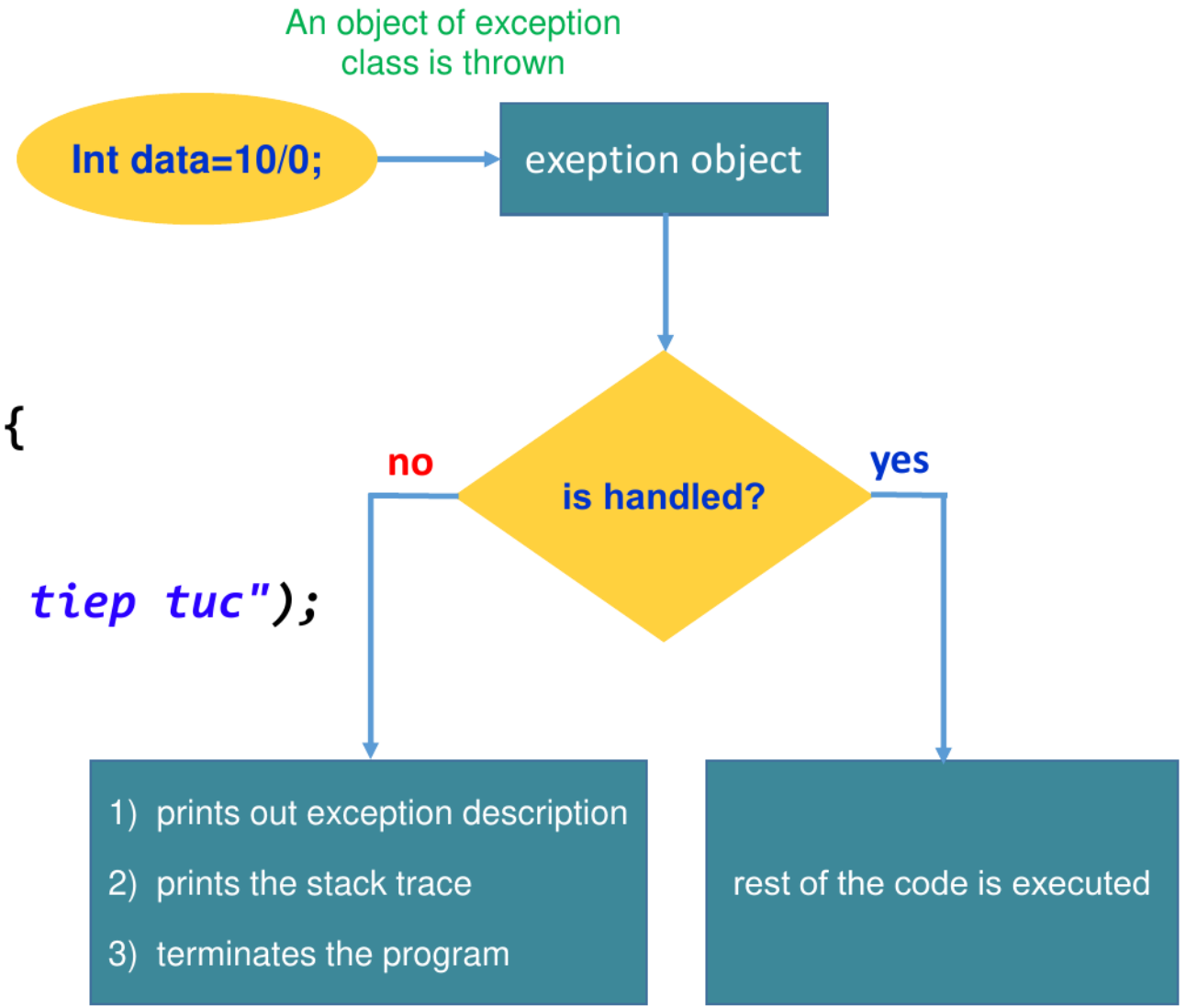
```
try {
      //đoạn code có thể đưa ra ngoại lệ
      } catch (Ten_lop_NgoaiLe e) {

      }
```

Khối lệnh “try / catch” (tt)

- **Ví dụ:**

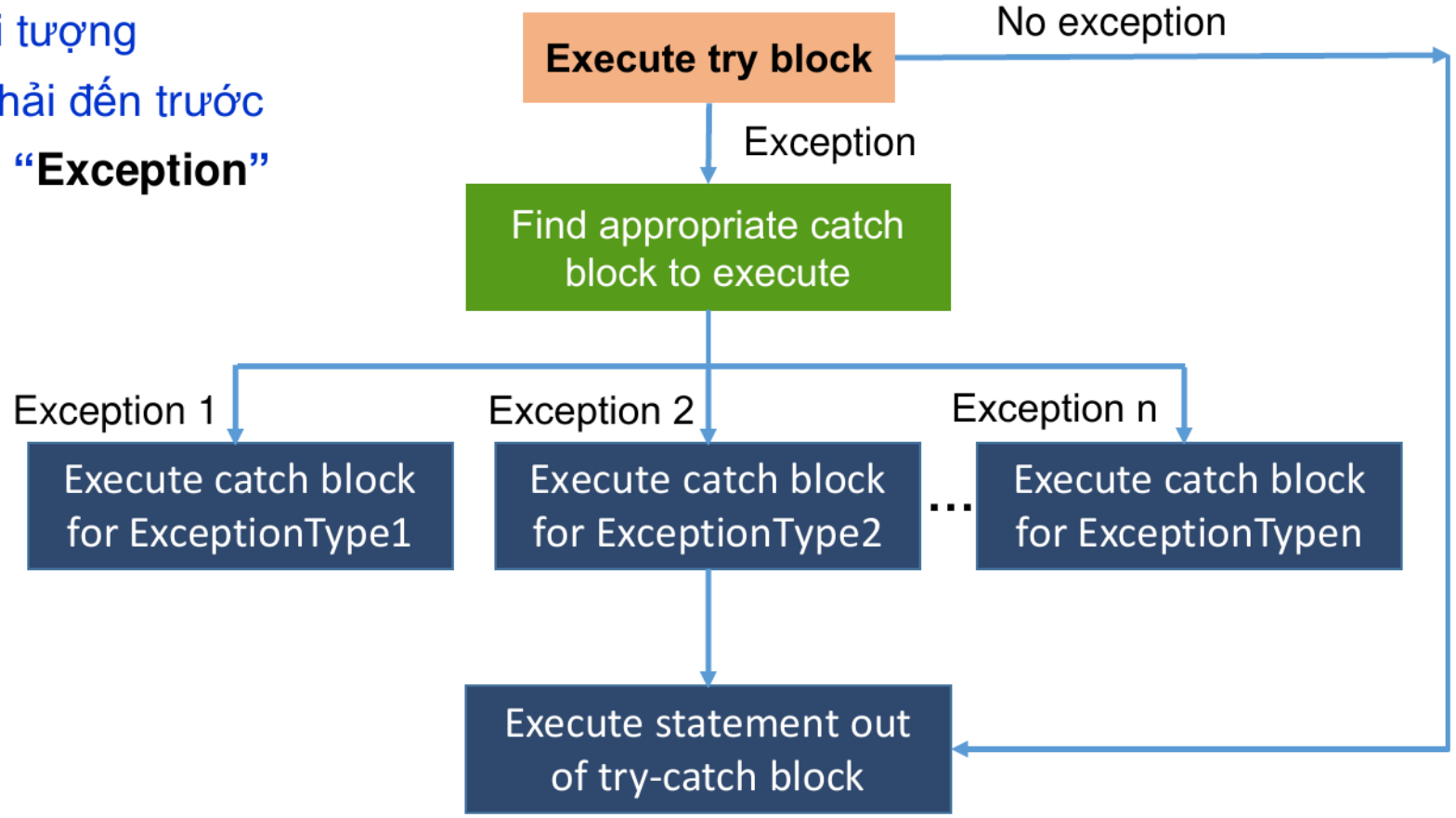
```
try {
    int data =10/0;
 } catch (ArithmeticException e) {
   System.out.println(“Error");
 }
 System.out.println("Chuong trinh tiep tuc");
```



Multi-catch

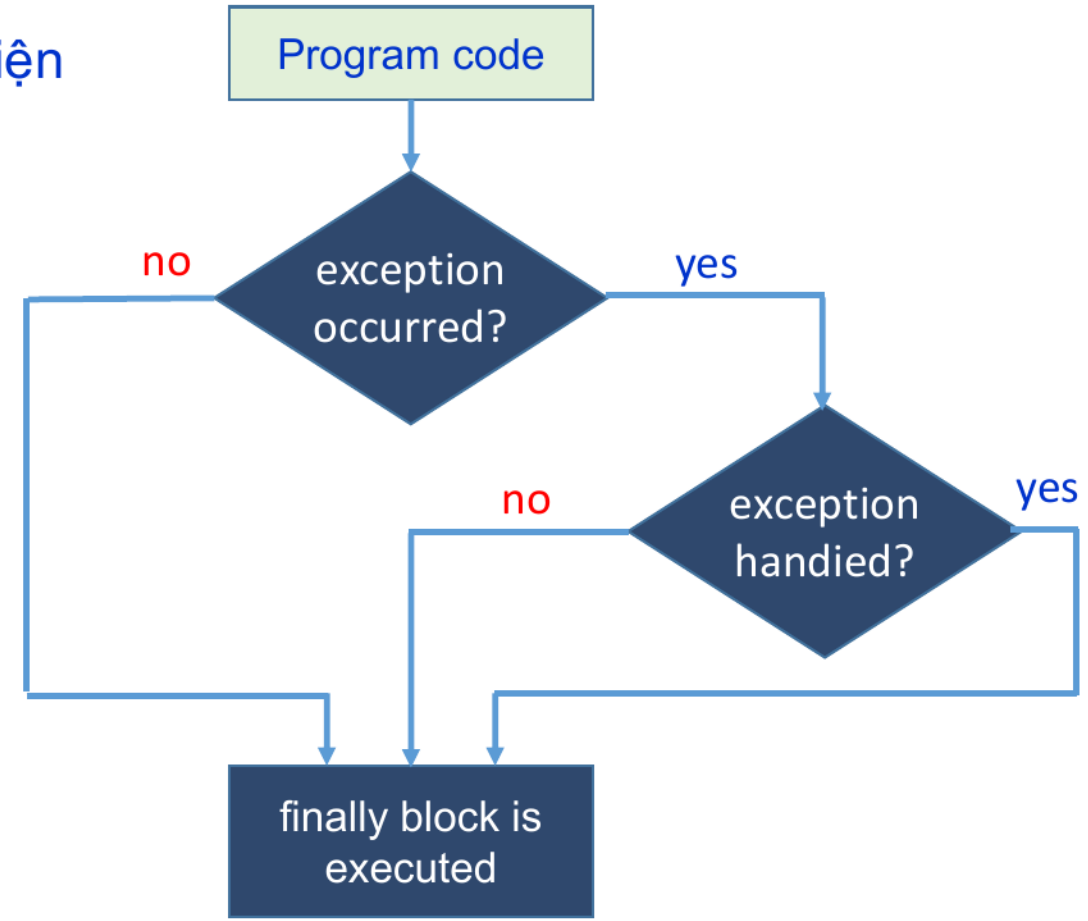
- **Khái niệm:** Thực thi nhiều hơn một khối lệnh **"catch"** theo sau khối lệnh **"try"**
- **Lưu ý:**
  - Mỗi khối lệnh **"catch"** xử lý các ngoại lệ khác nhau
  - Trình bày theo thứ tự các ngoại lệ từ lớp con đến lớp cha

- **Ví dụ:** bắt ngoại lệ cho đối tượng **"ArithmeticException"** phải đến trước bắt ngoại lệ cho đối tượng **"Exception"**



Khối lệnh "try/catch" kết hợp "finally"

- **Sử dụng:** Thường được sử dụng khi thực hiện các lệnh quan trọng như đóng một kết nối.
- **Lưu ý:**
  - Luôn được thực thi dù có phát sinh ngoại lệ hay không.
  - Theo sau khối lệnh "**try**" hoặc khối lệnh "**try/catch**".



Java

Khối lệnh "try/catch" kết hợp "finally" (tt)

```
InputStream in = new FileInputStream(. . .);
try
{
   // 1
      code that might throw exceptions
   // 2
}
catch (IOException e)
{
   // 3
      show error message
   // 4
}
finally
{
   // 5
      in.close();
}
   // 6
```

Có 3 trường hợp thực thi "finally":

1. Không có ngoại lệ xảy ra:

   → 1, 2, 5, 6

2. Có ngoại lệ xảy ra và được xử lý ở "catch":

   → 1, 3, 4, 5, 6

3. Có ngoại lê xảy ra và không được xử lý ở "catch":

   → 1, 5

Truy hồi ngoại lệ

- **Khái niệm:** chương trình cho đến khi gặp phương thức đầu tiên phát sinh ra ngoại lệ đó

```
public class TestExceptionPropagation {
void m(){
    int data=50/0;
  }
void n(){
    m();
  }
void p(){
   try{
    n();
   }catch(Exception e)
       {System.out.println("exception handled");}
}
public static void main(String[] args) {
TestExceptionPropagation obj=new TestExceptionPropagation();
   obj.p();
   System.out.println("normal flow...");
}
```

Result

```
exception handled
normal flow...
```

3. “throw” và “throws”

- **throw:** dùng để đưa ra một đối tượng ngoại lệ một cách rõ ràng
  - **Ví dụ:**

```
throw new IOException("sorry device error");
```

- **throws:** dùng để khai báo một ngoại lệ có thể xảy ra ở phương thức
  - Đặt ở phần tiêu đề phương thức
  - Chỉ nên khai báo “checked exception”

Bài thực hành cuối chương 9

ĐỀ BÀI

- Viết chương trình xây dựng lớp InformationExecption để biểu diễn lỗi định dạng số điện thoại và email đăng nhập không hợp lệ.
- Viết chương trình nhập thông tin gồm số điện thoại và email, kiểm tra nếu không đúng định dạng thì thông báo lỗi.

THANK YOU!