Chương 7
XỬ LÝ NGOẠI LỆ (Exception Handling)

1 - NỘI DUNG

- Định nghĩa một ngoại lệ (Exception)
- Hiểu được mục đích của việc xử lý ngoại lệ
- Hiểu được các kiểu ngoại lệ khác nhau trong Java
- Mô tả mô hình xử lý ngoại lệ
- Hiểu được các khối lệnh chứa nhiều khối xử lý ngoại lệ (catch)
- Mô tả cách sử dụng các khối 'try', 'catch' và 'finally'
- Giải thích cách sử dụng các từ khoá 'throw' và 'throws'
- Tự tạo ra các ngoại lệ

Chương 7
XỬ LÝ NGOẠI LỆ (Exception Handling)

2 - KHÁI NIỆM VÀ MỤC ĐÍCH SỨ DỤNG

- Exception (ngoại lệ) là một đối tượng được tạo ra khi chương trình gặp lỗi được java trả về nhằm mục đích dò tìm và sửa lỗi chương trình
- Ví dụ, việc chia cho 0 sẽ tạo một lỗi trong chương trình.
- Một chương trình nên có cơ chế xử lý ngoại lệ thích hợp. Nếu không, chương trình sẽ bị ngắt khi một ngoại lệ xảy ra. Khi đó, tất cả các nguồn tài nguyên mà hệ thống đã cấp không được giải phóng, gây lãng phí tài nguyên.
- Để tránh trường hợp này, tất cả các nguồn tài nguyên mà hệ thống cấp nên được thu hồi lại. Tiến trình này đòi hỏi cơ chế xử lý ngoại lệ thích hợp.

Chương 7
XỬ LÝ NGOẠI LỆ (Exception Handling)

3 - CÁC KIỂU EXCEPTION

3.1 - Checked exception: Sử dụng để kiểm tra những lỗi phổ biến trong lúc biên dịch (ví dụ truy xuất một tập tin không có trên ổ đĩa thì sẽ nhận được một java.io.FileNotFoundException). Checked exception kế thừa từ java.lang.Exception và chúng được dùng trong try-catch-finally.

3.2 – Error: Là một loại lỗi nằm ngoài phạm vi ứng dụng, được java trả về trong đối tượng java.io.IOError.

Ví dụ ứng dụng mở một tập tin thành công nhưng khi đọc dữ liệu lại không đọc được do ổ cứng bị hỏng.

3.3 - Unchecked exception: Thường được hiểu là những exception được sinh ra do lỗi logic khi lập trình và được trả về lúc runtime. Unchecked exception kế thừa từ java.lang.RuntimeException

Chương 7
XỬ LÝ NGOẠI LỆ (Exception Handling)

4 - XỬ LÝ EXCEPTION

✓ Khi một ngoại lệ xảy ra, đối tượng tương ứng với ngoại lệ đó được tạo ra.

✓ Đối tượng này chứa thông tin chi tiết về ngoại lệ. Thông tin này có thể được nhận về và được xử lý. Các môi trường runtime như 'IllegalAccessException', 'EmptyStackException' v.v... có thể tạo ra ngoại lệ.

✓ Chương trình đôi khi có thể tự tạo ra ngoại lệ.

✓ Lớp 'Throwable' được Java cung cấp là lớp trên cùng của lớp Exception, là lớp cha của tất cả các ngoại lệ khác.

Chương 7
XỬ LÝ NGOẠI LỆ (Exception Handling)

5 - MÔ HÌNH XỬ LÝ EXCEPTION

✓ Mô hình xử lý ngoại lệ của Java được gọi là 'catch and throw'. Trong mô hình này, khi một ngoại lệ xảy ra, ngoại lệ sẽ bị chặn và chương trình chuyển đến một khối xử lý ngoại lệ. Người lập trình phải xử lý các ngoại lệ khác nhau có thể phát sinh trong chương trình. Các ngoại lệ phải được xử lý, hoặc thoát khỏi chương trình khi nó xảy ra.

✓Ngôn ngữ Java cung cấp 5 từ khoá sau để xử lý các ngoại lệ:

- try
- catch
- throw
- throws
- finally

Chương 7
XỬ LÝ NGOẠI LỆ (Exception Handling)

5 - MÔ HÌNH XỬ LÝ EXCEPTION

✓ Cấu trúc của mô hình xử lý ngoại lệ:

```
try{
        // đoạn mã có khả năng gây ra ngoại lệ
}
catch(Exception e1)
{
        // Nếu các lệnh trong khối 'try' tạo ra ngoại lệ có loại
//e1, thì xử lý ngoại lệ nếu không chuyển xuống khối 'catch'
//tiếp theo
}
catch(Exception e2)
{
        // Nếu các lệnh trong khối 'try' tạo ra ngoại lệ có loại
//e2, thì xử lý ngoại lệ, nếu không chuyển xuống khối
//'catch' tiếp theo
}
```

Chương 7
XỬ LÝ NGOẠI LỆ (Exception Handling)

5 - MÔ HÌNH XỬ LÝ EXCEPTION

```
catch(Exception eN)
{
// Nếu các lệnh trong khối 'try' tạo ra ngoại lệ có loại eN, thì thực hiện
//xử lý ngoại lệ nếu không chuyển xuống khối 'catch' tiếp theo
}
finally
{
        // khối lệnh này luôn được thực hiện cho dù ngoại lệ có xảy ra hay không.
}
```

Chương 7
XỬ LÝ NGOẠI LỆ (Exception Handling)

6 - CÁC KHỐI TRY VÀ CATCH

VD1:

```
try
{
        doFileProcessing(); // phương thức do người sử dụng
                            //định nghĩa
        displayResults();
}
catch (Exeption e) // thể hiện của ngoại lệ
{
System.err.println("Error :" + e.toString());
        e.printStackTrace();
}
```

- Ở đây, 'e' là đối tượng của lớp 'Exception'. Ta sử dụng đối tượng này để in các chi tiết về ngoại lệ. Các phương thức 'toString' và 'printStackTrace' được sử dụng để mô tả các ngoại lệ xảy ra.

Chương 7
XỬ LÝ NGOẠI LỆ (Exception Handling)

6 - CÁC KHỐI TRY VÀ CATCH

VD2:

```
try
{
        doFileProcessing();
        displayResults(); }
catch(LookupException e)
{
        handleLookupException(e); // phương thức xử lý lỗi
                                  // do người sử dụng định nghĩa
}
catch(Exception e)
{
        System.err.println("Error:" + e.printStackTrace());
}
}
```

Chương 7
XỬ LÝ NGOẠI LỆ (Exception Handling)

6 - CÁC KHỐI TRY VÀ CATCH

VD3: Try long nhau

```
try
{
        statement 1;
        statement 2;
        try
        {
                statement1;
                statement2;
        }
        catch(Exception e) // của khối try trong
        { }
}
catch(Exception  e) // của khối try ngoài
{ }
…
```

Chương 7
XỬ LÝ NGOẠI LỆ (Exception Handling)

6 - CÁC KHỐI TRY VÀ CATCH

CtrinhDemo:

```
class TryClass
{
        public static void main(String args[])
        {
                int demo=0;
                try
                {
                        System.out.println(20/demo);
                }
                catch(ArithmeticException a)
                {
                System.out.println("Cannot Divide by zero");
                }
        }
}
```

Chương 7
XỬ LÝ NGOẠI LỆ (Exception Handling)

7 - KHỐI FINALLY

✓ Khi một ngoại lệ xuất hiện, phương thức đang được thực thi có thể bị dừng mà không được hoàn thành. Nếu điều này xảy ra, thì các đoạn mã phía sau (ví dụ như đoạn mã có chức năng thu hồi tài nguyên, như các lệnh đóng tập viết ở cuối phương thức) sẽ không bao giờ được gọi.

✓ Java cung cấp khối 'finally' để thực hiện tất cả các việc thu dọn khi một ngoại lệ xảy ra. Khối này có thể được sử dụng kết hợp với khối 'try'. Khối 'finally' chứa các câu lệnh thu hồi tài nguyên về cho hệ thống hay lệnh in ra các câu thông báo.

✓ Các lệnh này bao gồm:

Chương 7
XỬ LÝ NGOẠI LỆ (Exception Handling)

7 - KHỐI FINALLY

- Đóng tập tin.
- Đóng ResultSet (được sử dụng trong chương trình cơ sở dữ liệu).
- Đóng lại các kết nối được tạo trong cơ sở dữ liệu.

VD: try

```
{
        doSomethingThatMightThrowAnException();
}
finally
{
        cleanup();
}
```

✓Phương thức 'cleanup()' được gọi nếu phương thức 'doSomethingThatMightThrowAnException()' gây ra ngoại lệ và ngay kể cả khi không có ngoại lệ nào xảy ra, sau đó thực hiện tiếp phần sau khối lệnh 'finally'.

Chương 7
XỬ LÝ NGOẠI LỆ (Exception Handling)

7 - KHỐI FINALLY

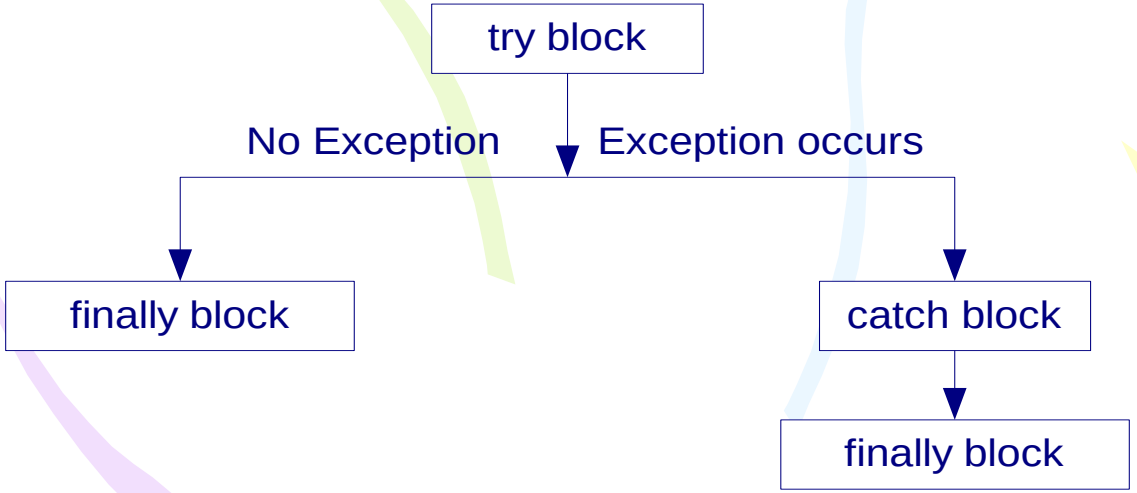
✓ Khối 'finally' là tuỳ ý, không bắt buộc.

✓ Khối này được đặt sau khối 'catch' cuối cùng.

✓ Chương trình sẽ thực thi câu lệnh đầu tiên của khối 'finally' ngay sau khi gặp câu lệnh 'return' hay lệnh 'break' trong khối 'try'.

✓ Khối 'finally' bảo đảm lúc nào cũng được thực thi, bất chấp có ngoại lệ xảy ra hay không.

✓ Sự thực hiện của các khối 'try', 'catch' và 'finally'.



Chương 7
XỬ LÝ NGOẠI LỆ (Exception Handling)

8 - Định nghĩa ngoại lệ với lệnh 'throw' và 'throws'

- Các ngoại lệ có thể được tạo ra bằng cách sử dụng từ khoá 'throw'.
- Từ khóa 'throw' chỉ ra một ngoại lệ vừa xảy ra.
- Từ khoá 'throws' để tạo nhiều ngoại lệ. Các ngoại lệ cách nhau bởi dấu phẩy

✓ VD:

```
try
{
   if (flag<0)
   {
        throw new MyException(); // user-defined
   }
}
```

Chương 7
XỬ LÝ NGOẠI LỆ (Exception Handling)

8 - Định nghĩa ngoại lệ với lệnh 'throw' và 'throws'

```
public class Example
{ public void exceptionExample() throws ExException,
   LookupException
   {
        try
        {
                // các lệnh
        }
        catch(ExException exmp)
        {
        }
        catch(LookupException lkpex)
        {
        }
}}
```