

Quản lý Exceptions

GV: ThS. Phan Nguyệt Minh
minhpn@uit.edu.vn

<http://courses.uit.edu.vn>

Nội dung

- Giới thiệu về Exception
- Kiểm soát Exception
- Ví dụ minh họa
- Thư viện phân cấp các lớp Exception

Giới thiệu về Exception

Ví dụ 1:

```
...  
int x = 10;  
int y = 0;  
float z = x/y;  
System.out.print("Ket qua la:" + z);  
...
```

Dòng lệnh thứ 3 có lỗi chia cho 0, vì vậy đoạn chương trình kết thúc và dòng lệnh thứ 4 xuất kết quả ra màn hình không thực hiện được.

Giới thiệu về Exception (tt)

Ví dụ 2:

```
...  
void docfile(String filename)  
{  
    ...  
    FileInputStream fin = new FileInputStream(filename);  
    ...  
}
```

Dòng lệnh trên có khả năng xảy ra lỗi đọc file (chẳng hạn khi file không có trên đĩa)

Giới thiệu về Exception (tt)

- **Exception**
 - ✓ Dấu hiệu của lỗi trong khi thực hiện chương trình
 - ✓ ví dụ: lỗi chia cho 0, đọc file không có trên đĩa, ...
- **Quản lý Exception (Exception handling)**
 - ✓ Kiểm soát được lỗi từ những thành phần chương trình
 - ✓ Quản lý Exception theo 1 cách thống nhất trong những project lớn
 - ✓ Hạn chế, bỏ bớt những đoạn source code kiểm tra lỗi trong chương trình.

Kiểm soát Exception

Ví dụ 1:

```
...  
try {  
    int x = 10;  
    int y = 0;  
    float z = x/y;  
    System.out.print("Ket qua la:" + z);  
}  
catch(ArithmeticException e) {  
    System.out.println("Loi tinh toan so hoc")  
}  
...
```

Kiểm soát Exception (tt)

Ví dụ 2:

```
...  
void docfile(String filename) throws IOException {  
    ...  
    FileInputStream fin = new FileInputStream(filename);  
    ...  
}
```

Kiểm soát Exception (tt)

Hoặc

```
...  
void docfile(String filename) { ...  
    try {  
        ...  
        FileInputStream fin = new  
        FileInputStream(filename);  
        ...  
    }  
    catch (IOException e) {  
        System.out.println("Loi doc file");  
    }  
}
```


Kiểm soát Exception (tt)

- Khi có lỗi phương thức sẽ ném ra một exception
- Việc kiểm soát exception giúp chương trình kiểm soát được những trường hợp ngoại lệ và xử lý lỗi.
- Những lỗi không kiểm soát được sẽ có những ảnh hưởng bất lợi trong chương trình.
- Dùng từ khóa **throws** để chỉ định những loại exception mà phương thức có thể ném ra.
- ▶ <tiền tố> <tên phương thức>(<đối số>) **throws** <các exceptions>

Kiểm soát Exception (tt)

- Đoạn code có thể sinh ra lỗi cần đặt trong khối lệnh bắt đầu bằng **try**.
- Đoạn code để kiểm tra, xử lý trong trường hợp có lỗi xảy ra đặt trong khối lệnh **catch**.

```
try {  
    // Đoạn mã có thể sinh ra lỗi ...  
}  
catch (<Kiểu Exception>){  
    // Đoạn mã kiểm soát lỗi  
}
```

Kiểm soát Exception (tt)

- Khối lệnh đặt trong **finally** luôn được thực thi cho dù có Exception hay không.
- Thường dùng để giải phóng tài nguyên

```
try { // Đoạn mã có thể sinh ra lỗi ...  
}  
Catch (<Kiểu Exception>) { // Đoạn mã kiểm soát lỗi  
}  
finally {  
    // Đoạn mã luôn luôn được thực thi  
}
```

Kiểm soát Exception (tt)

```
try {  
    // Khối lệnh trước dòng lệnh sinh ra lỗi  
    // Dòng lệnh sinh ra lỗi (Exception)  
    ...  
}  
catch (<Kiểu Exception>){  
    // Đoạn mã kiểm soát lỗi  
}  
finally { ...  
}
```

Khối lệnh sau dòng lệnh sinh ra lỗi sẽ bị bỏ qua và không thực hiện khi có exception

Ví dụ kiểm soát Exception chia cho 0

```
import java.io.*;

public class MainClass {

    public static void main(String[] args) {

        try {
            int num_1, num_2;

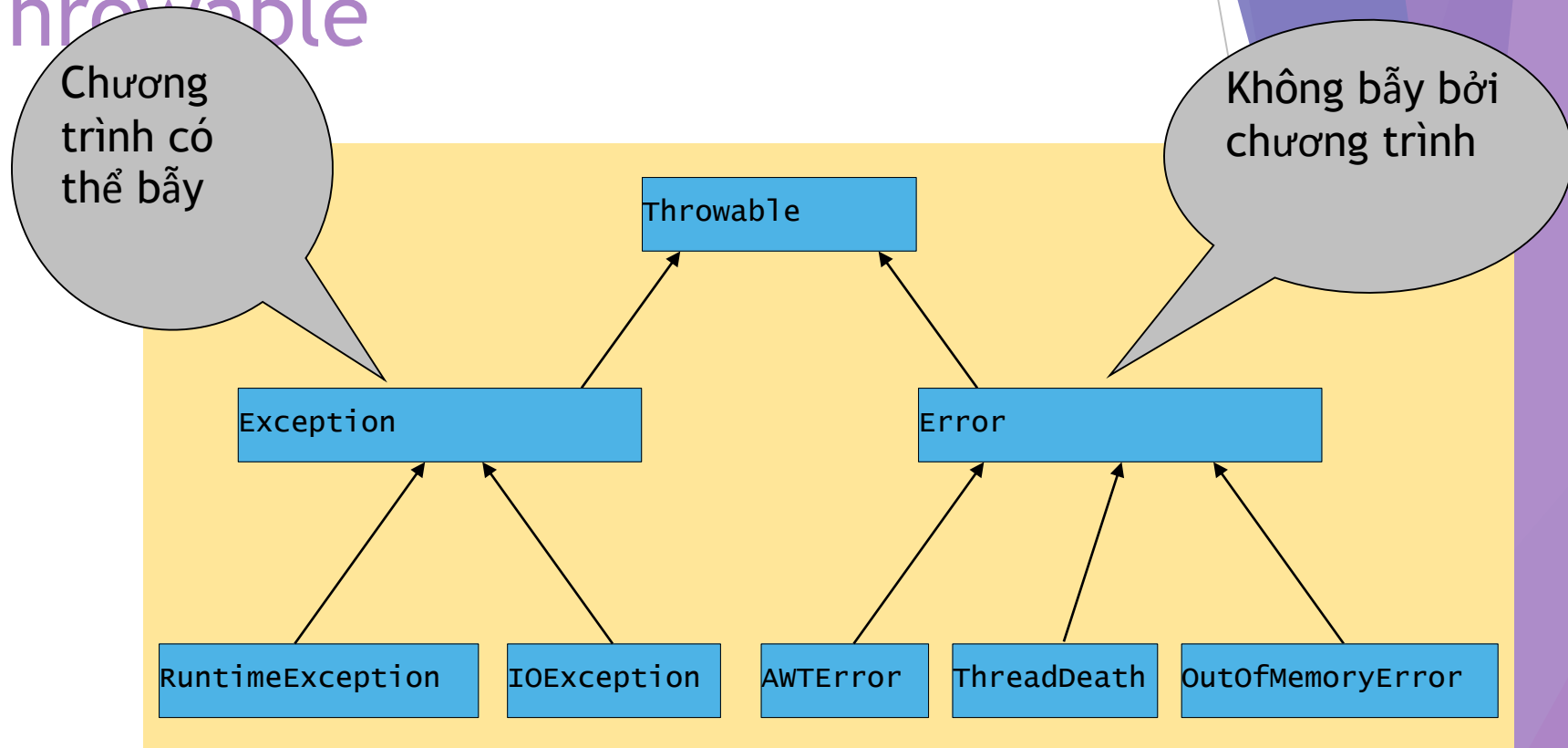
            BufferedReader in = new BufferedReader(new
            InputStreamReader(System.in));

            System.out.print("\n Nhap so thu 1:");
            num_1 = Integer.parseInt(in.readLine());
            System.out.print("\n Nhap so thu 2:");
            num_2 = Integer.parseInt(in.readLine());
            float rs = num_1/num_2;
            System.out.print("\n Ket qua:" + rs);
        }
    }
}
```

Ví dụ kiểm soát Exception chia cho 0 (tt)

```
        catch (ArithmeticException e) {  
            System.out.print("Loi chia cho 0");  
        }  
        catch (IOException e) {  
            System.out.print("Loi xuat nhap");  
        }  
        catch (Exception e) {  
            System.out.print("Loi khac");  
        }  
        System.out.print("Kiem soat duoc loi hay  
        Khong co loi");  
    }  
}
```

Phân cấp thư viện của lớp Throwable



- Có thể định nghĩa các exception mới bằng cách dẫn xuất (**extends**) từ những lớp Exception đang có.

Q/A