

LẬP TRÌNH JAVA

Bài 13: Ngoại lệ trong java (Exception)

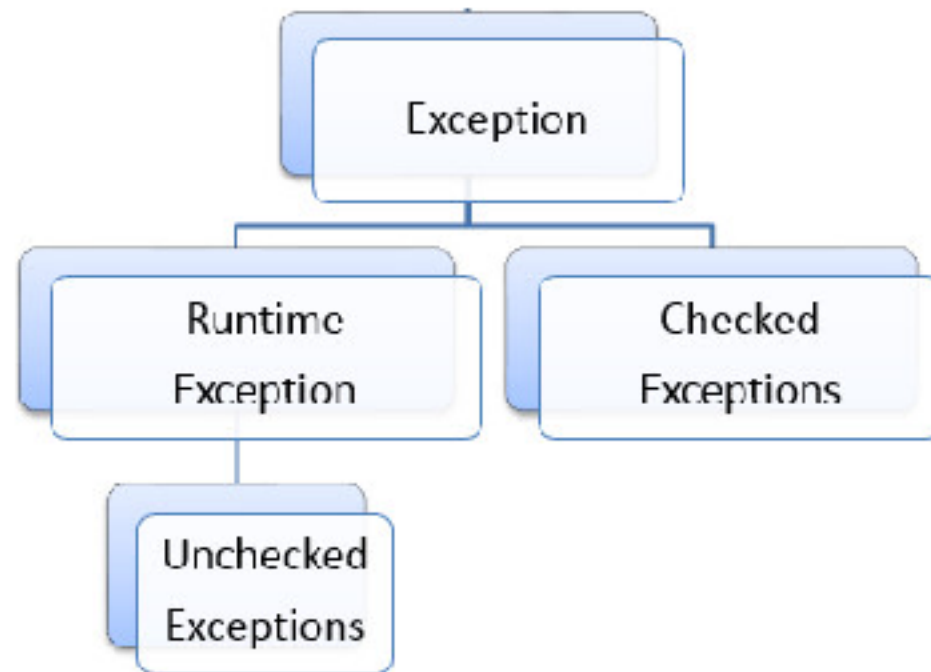
Khái niệm

- ❑ Ngoại lệ (exception) là thuật ngữ chỉ tình trạng sai hoặc bất thường xảy ra khi một chương trình đang chạy
- ❑ Xảy ra khi nào?
 - Chia cho 0
 - Đọc kiểu số thực vào một biến kiểu nguyên
 - Truy cập phần tử bên ngoài mảng
 - Ghi vào file khi chưa mở file, ghi vào một file chưa được tạo ra

Các loại exception

Ngoại lệ '*unchecked*': Là các ngoại lệ không cần phải 'catch' khi viết mã

Ngoại lệ '*checked*': Là các ngoại lệ phải được 'catch' khi viết mã



Ví dụ

❑ Chương trình này sẽ gặp lỗi ở đâu?

```
import java.util.*;

public class TestException {

    public static void main (String args[]) {
        Scanner scanner = new Scanner(System.in);

        System.out.print( "Numerator: " );
        int numerator = scanner.nextInt();
        System.out.print( "Denominator: " );
        int denominator = scanner.nextInt();

        int result = numerator/denominator;
        System.out.printf("\nResult: %d / %d = %d\n",
                           numerator, denominator, result );
    }
}
```

Nhập kiểu không phải số nguyên

Có thể bị chia cho 0

Ví dụ

```
% java TestException
Numerator: 3
Denominator: d
Exception in thread "main" java.util.InputMismatchException
    at java.util.Scanner.throwFor(Unknown Source)
    at java.util.Scanner.next(Unknown Source)
    at java.util.Scanner.nextInt(Unknown Source)
    at java.util.Scanner.nextInt(Unknown Source)
    at TestException.main(TestException.java:11)

% java TestException
Numerator: 2
Denominator: 0
Exception in thread "main" java.lang.ArithmeticException: / by zero
    at TestException.main(TestException.java:13)
```

Xử lý ngoại lệ

- ❑ Sử dụng khối `try/catch` và `try/catch/finally`
- ❑ Ném ngoại lệ `throw`
- ❑ Né ngoại lệ `throws`

Sử dụng try/catch

```
import java.util.Scanner;
import java.util.InputMismatchException;
import java.lang.ArithmeticException;

public class TwoCatchBlocks {
    public static void main (String args[]) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print( "Numerator: " );
            int numerator = scanner.nextInt();
            System.out.print( "Denominator: " );
            int denominator = scanner.nextInt();
            int result = numerator/denominator;
            System.out.printf("Result: %d / %d = %d\n",
                             numerator, denominator, result );
        }
        catch (InputMismatchException e) {
            System.out.println("Input error. ");
        }
        catch (ArithmeticException e) {
            System.out.println("Arithmetic error.");
        }
    }
}
```

```
% java TwoCatchBlocks
Numerator: e
Input error.
```

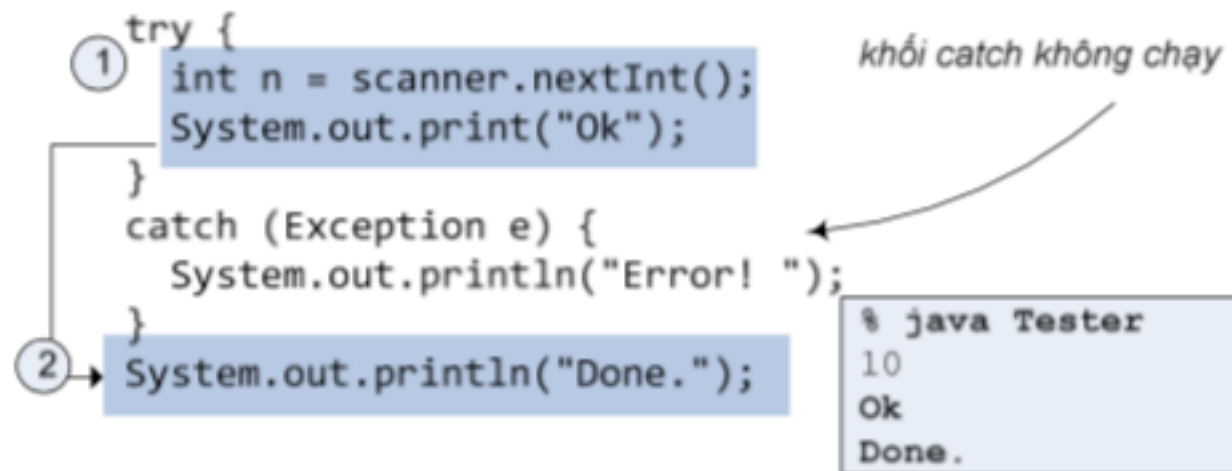
```
% java TwoCatchBlocks
Numerator: 2
Denominator: 0
Arithmetic error.
```

Hoạt động khối try/catch

- ❑ Phương thức được gọi thành công và không sinh ngoại lệ
- ❑ Phương thức được gọi ném ngoại lệ và khối catch bắt được ngoại lệ đó
- ❑ Phương thức được gọi ném ngoại lệ nhưng khối catch không bắt được ngoại lệ đó

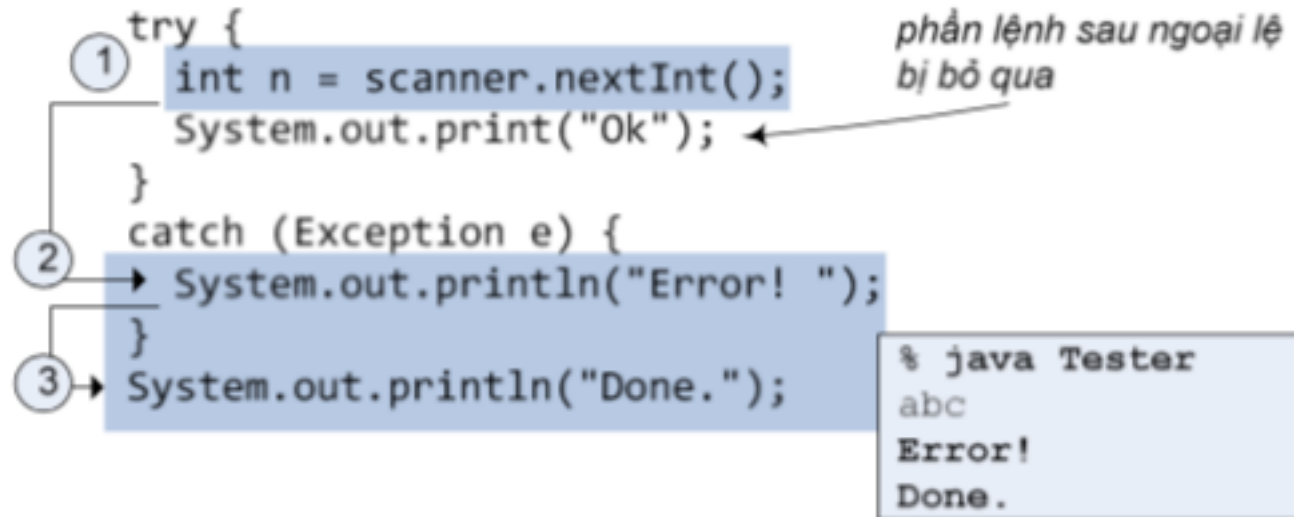
Hoạt động khối try/catch

Phương thức không sinh ngoại lệ



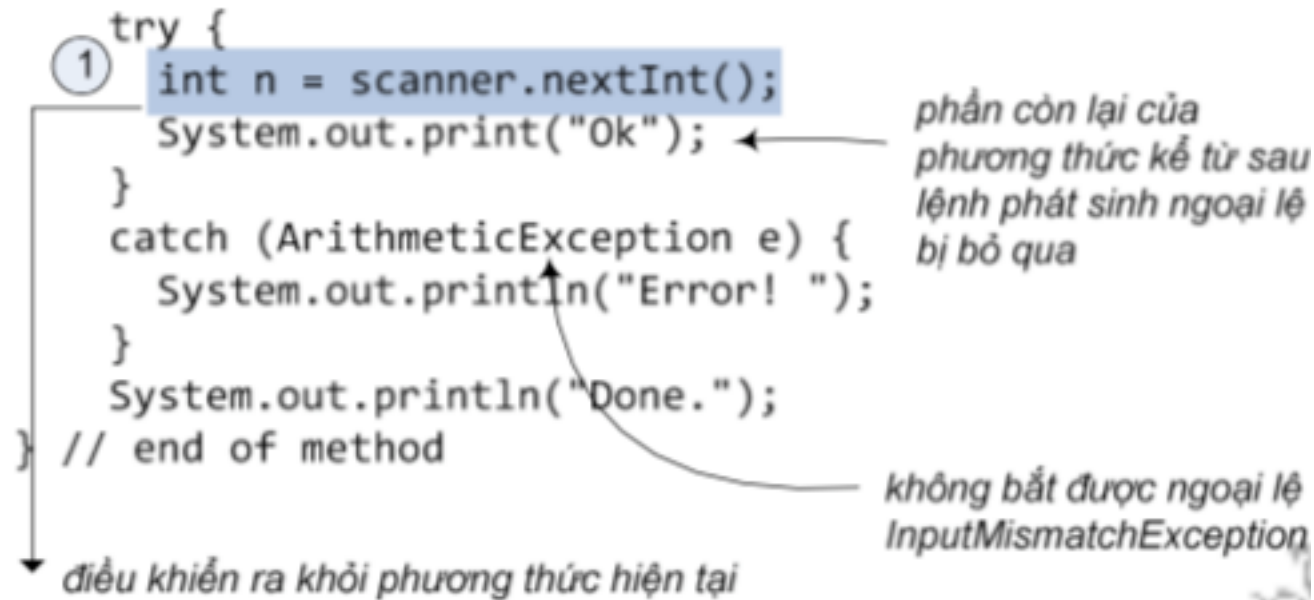
Hoạt động khối try/catch

- ❑ Phương thức được gọi ném ngoại lệ và khối catch bắt được ngoại lệ đó



Hoạt động khối try/catch

- ❑ Phương thức được gọi ném ngoại lệ nhưng khối catch không bắt được ngoại lệ đó



Try/catch/finally

Khối finally - những việc dù thế nào cũng phải làm

Ví dụ: Nếu trứng luộc thành công, ta cũng tắt lò. Tóm lại, dù chuyện gì xảy ra thì ta cũng đều phải tắt lò

```
try {
    turnOvenOn();
    egg.boil();
}
catch (EggException e) {
    System.out.println("Error! ");
}
finally {
    turnOvenOff();
}
```

Ném ngoại lệ - throw - throws

Các bước

- Tại dòng khai báo phương thức rằng nó có thể ném loại ngoại lệ đó (dùng từ khóa throws);
- Tạo một ngoại lệ và ném nó (bằng lệnh throw)

Ném ngoại lệ - throw - throws

```
public class Fraction {  
    private int numerator, denominator;  
  
    public Fraction (int n, int d) throws Exception {  
        if (d==0) throw new Exception();  
  
        numerator = n;  
        denominator = d;  
    }  
}
```

tuyên bố rằng phương thức này có thể ném ngoại lệ loại Exception

tạo một đối tượng Exception mới và ném nó tới nơi gọi phương thức

```
public class TestFraction {  
    public static void main(String [] args) {  
        try {  
            Fraction f = new Fraction (2,0);  
        }  
        catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

nếu không khắc phục được sự cố thì ít nhất cũng in ra thông tin lỗi vết sự cố

Né ngoại lệ - throws

Khi sử dụng một hàm nào đó có thể sinh ra ngoại lệ ta có thể dùng từ khóa **throws** để né ngoại lệ

Ví dụ:

Phương thức sleep trong lớp Thread luôn yêu cầu try/catch, nếu không muốn xử lý ngoại lệ tại hàm đó thì dùng throws để né ngoại lệ tại nơi phương thức đó và đưa cho hàm

```
public static void startThread() {  
    Thread thread = new Thread();  
    thread.sleep( millis: 1000);  
}
```

```
public static void startThread() {  
    Thread thread = new Thread();  
    try {  
        thread.sleep( millis: 1000);  
    } catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
}
```

```
public static void startThread() throws InterruptedException {  
    Thread thread = new Thread();  
    thread.sleep( millis: 1000);  
}
```

```
public static void main(String[] args) {  
    try {  
        startThread();  
    } catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
}
```

```
public static void startThread() throws InterruptedException {  
    Thread thread = new Thread();  
    thread.sleep( millis: 1000);  
}
```

Bài tập

1) Tạo class tên SinhVien có các thuộc tính như mã sinh viên, họ tên, điểm, xếp loại (Xếp loại dựa vào điểm, nếu điểm ≥ 8 thì xếp loại giỏi, điểm ≥ 7 thì xếp loại khá, điểm ≥ 5 thì xếp loại trung bình, điểm < 5 thì xếp loại kém).

Viết các setter, getter, constructor, toString.

Nhập dữ liệu cho sinh viên từ bàn phím. Viết code xử lý nhập dữ liệu hợp lệ. Nếu nhập sai kiểu dữ liệu thì thông báo lỗi và yêu cầu nhập lại. Ví dụ nhập điểm là **10a** thì chương trình sẽ hiển thị thông báo “Bạn phải nhập dữ liệu là kiểu số”

In thông tin sinh viên ra màn hình.