

Chương 3: LUỒNG NHẬP XUẤT TRONG JAVA

Khoa CNTT

ĐH GTVT TP.HCM

- ❶ Biệt lệ (Exception)
- ❷ Khái niệm luồng (Stream)
- ❸ Nhập/Xuất chuẩn
- ❹ Truy xuất File

Exception là gì?

Định nghĩa (Oracle's java document)

An exception is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions.

Ví dụ

- * Thực hiện phép toán chia cho 0.
- * Parse chuỗi "abc" thành số nguyên.
- * ...

Phân loại Exception

Checked Exception

- * Được sử dụng trong signature của method
- * Được kiểm tra tại thời điểm compile-time
- * Có ý nghĩa là: đây là phương thức có thể gây ra "lỗi" vì vậy **khi sử dụng nó** cần phải "bắt - catch" ngoại lệ.

Phân loại Exception

Checked Exception - Ví dụ: method có khai báo exception

```
static public int chia(int s1, int s2) throws Exception {  
    return s1 / s2;  
}
```

Checked Exception - Ví dụ: gọi method ở trên

```
try {  
    double kq = ExcepDemo.chia(10, 0); //method call  
    System.out.println("ket qua = " + kq);  
} catch (Exception ex) {  
    System.out.println(ex.getMessage());  
}
```

Phân loại Exception

Unchecked Exception

- * Được sử dụng trong khối lệnh `try ... catch ...`
- * Được kiểm tra tại thời điểm run-time

Lệnh try catch

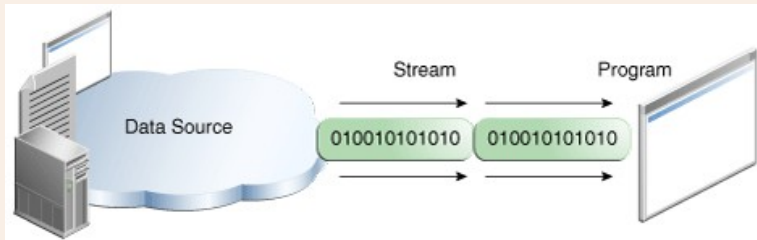
```
try {  
    //khởi lệnh có khả năng xảy ra lỗi  
} catch (Exception ex) {  
    //khởi lệnh xử lý khi xảy ra lỗi  
}  
finally{  
    //khởi lệnh luôn được thực hiện dù xảy ra lỗi hay không  
}
```

Stream là gì?

- * In computer science Streams are defined as the sequence of data elements that is made available over time. It represents a Source (which generates the data in the form of Stream) and a destination (which consumes or read data available as Stream).
- * In other simple words it is a flow of data from which you can read or write data to it.

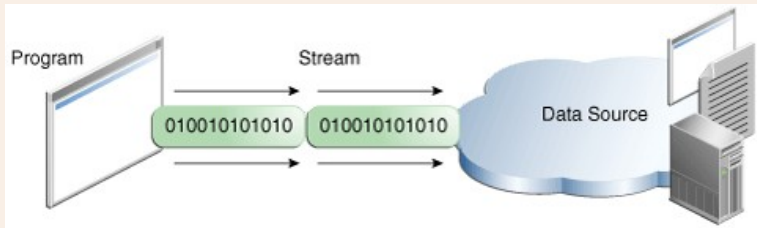
Stream trong Java

Input stream



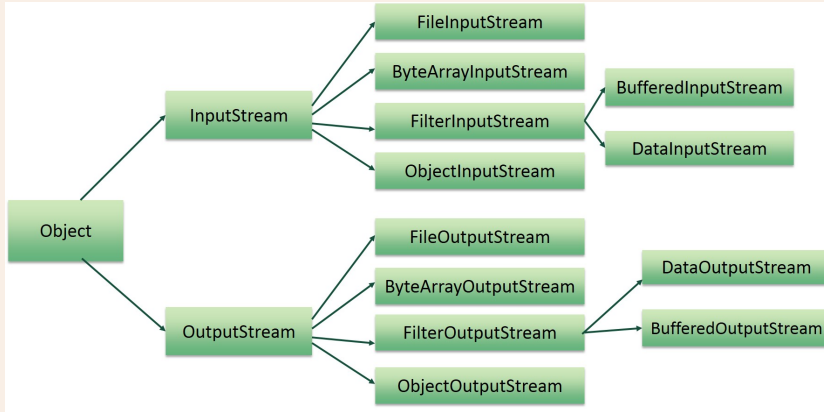
Stream trong Java

Output stream



Stream trong Java

Hierarchy of classes to deal with Input and Output streams:



Nhập/Xuất chuẩn trong Java (Standard Streams)

Các packages xuất nhập chuẩn

- * `System.in` = bàn phím
- * `System.out` = màn hình
- * `System.err` → thông báo lỗi

Nhập dữ liệu từ bàn phím

```
int a, b, c;  
Scanner s = new Scanner(System.in); // sử dụng System.in  
System.out.print("Nhập a, b, c: ");  
a = s.nextInt();  
b = s.nextInt();  
c = s.nextInt();
```

Xuất dữ liệu ra màn hình

```
System.out.println("a = " + a);
```

Hoặc:

```
String kq = String.format("max(%d,%d,%d) = %d", a,b,c,m);  
System.out.println(kq);
```

Các trường hợp truy xuất file:

- * Truy xuất file nhị phân: dữ liệu trong file được xem như dãy các bytes nhị phân.
- * Truy xuất file văn bản: dữ liệu trong file được xem như dãy các mã Unicode của các kí tự.
- * Serialize & Deserialize: truy xuất file lưu trữ các objects.

Ghi file nhị phân

```
public static void ghiFile() {  
    FileOutputStream out = null;  
    try {  
        File f = new File("tho.dat");  
        out = new FileOutputStream(f);  
        String s = "Khi ta o chi la noi dat o";  
        out.write(s.getBytes()); // ghi day bytes.  
    } catch (Exception ex) { /* xu ly exception o day */ }  
    finally {  
        try { out.close(); }  
        catch (IOException ex) { /* xu ly exception o day */ }  
    }  
}
```

Truy xuất File

Đọc file nhị phân

```
public static void docFile() {  
    FileInputStream in = null;  
    try {File f = new File("tho.dat");  
        in = new FileInputStream(f);  
        int kq;  
        do {kq = in.read();  
            System.out.print((char) kq);  
        } while (kq != -1);  
    } catch (Exception ex) { /*...*/ }  
    finally {try {in.close();  
        } catch (IOException ex) { /*...*/ }  
    }  
}
```


Ghi file văn bản

```
public static void ghiFile() {  
    File f = new File("vanban.txt");  
    FileWriter fw = null;  
    try {  
        fw = new FileWriter(f);  
        fw.write("Khi ta o chi la noi dat o");  
        fw.write(System.getProperty("line.separator"));  
        fw.write("Khi ta di dat bong hoa tam hon");  
    } catch (Exception ex) { /*...*/ }  
    finally { try { fw.close();  
        } catch (IOException ex) { /*...*/ }  
    }  
}
```

Truy xuất File

Đọc file văn bản: **đọc từng ký tự**

```
public static void docFile() {  
    File f = new File("vanban.txt");  
    FileReader fr = null;  
    try {if (!f.exists()) {throw new Exception("File không tồn tại");}  
        fr = new FileReader(f);  
        int kq;  
        while ((kq = fr.read()) != -1) {System.out.print((char) kq);}  
    } catch (Exception ex) {/*...*/}finally {  
        try {fr.close();  
        } catch (IOException ex) {/*...*/}  
    }  
}
```

Đọc file văn bản: **đọc từng dòng**

```
public static void docFileByScanner() {  
    try {File f = new File("vanban.txt");  
        if (!f.exists()) {  
            throw new Exception("File không tồn tại");  
        }  
        Scanner scan = new Scanner(f);  
        String line;  
        while (scan.hasNextLine()) {  
            line = scan.nextLine();  
            System.out.println(line);  
        }scan.close();  
    } catch (Exception ex) { /* ... */ }  
}
```

Đọc file văn bản: sử dụng buffer

```
public static void docFileByBufferChar() {  
    try {  
        File f = new File("vanban.txt");  
        FileReader read = new FileReader(f);  
        BufferedReader b = new BufferedReader(read);  
        String d;  
        while ((d = b.readLine()) != null) {  
            System.out.println(d);  
        }  
        b.close();  
    } catch (Exception ex) { /*...*/ }  
}
```

Serialize

```
public static void ghiDSHHSerialize() {  
    File f = new File("hh.dat");  
    FileOutputStream fo = null;  
    ObjectOutputStream out = null;  
    try {  
        fo = new FileOutputStream(f);  
        out = new ObjectOutputStream(fo);  
        out.writeObject(HangHoa.dsHangHoa()); //note!  
    } catch (Exception ex) { /*...*/ } finally {  
        try {  
            out.close();  
            fo.close();  
        } catch (IOException ex) { /*...*/ }  
    }  
}
```

Deserialize

```
public static void docDSHHSerialize() {  
    File f = new File("hh.dat");  
    FileInputStream fo = null;  
    ObjectInputStream out = null;  
    try {  
        fo = new FileInputStream(f);  
        out = new ObjectInputStream(fo);  
        ArrayList<HangHoa> lst;  
        lst = (ArrayList<HangHoa>)out.readObject();  
        for(HangHoa h:lst)  
            System.out.println(h.toString());  
    } catch (Exception ex) { /*...*/ } finally {  
        try {  
            out.close();fo.close();  
        } catch (IOException ex) { /*...*/ }  
    }  
}
```

Bài tập

Sử dụng các kiến thức của chương này (exception & streams), xây dựng chương trình minh họa quá trình mua bán hàng hóa. Gồm các chức năng sau:

- 1 Hiển thị danh sách hàng hóa.
- 2 Cho phép chọn chức năng mua hàng.
- 3 Hiển thị thông tin giỏ hàng.
- 4 Cho phép cập nhật giỏ hàng.
- 5 Lưu hóa đơn mua hàng.

—Hết—