



Bài 6

Lỗi ngoại lệ - Exception

Đa luồng – Multi-threading

- Giới thiệu lỗi ngoại lệ - Exception
- Quản lý lỗi ngoại lệ
- Giới thiệu luồng và đa luồng
- Tạo và quản lý luồng

Lỗi ngoại lệ - Exception

- Là **một sự kiện xảy ra** trong quá trình **chương trình thực thi một câu lệnh** nào đó làm **phá vỡ luồng làm việc bình thường** của chương trình
- Khi gặp lỗi ngoại lệ, chương trình **lập tức ngừng lại và xuất hiện thông báo lỗi**

```
14 public class ExceptionDemo {
15
16     public static void main(String[] args) {
17         System.out.println("BKAP - Exception Demo - Start");
18         int a, b;
19         Scanner sc = new Scanner(System.in);
20         System.out.println("Nhap vao so thuc thu 1: ");
21         a = sc.nextInt();
22         System.out.println("Nhap vao so thuc thu 2: ");
23         b = sc.nextInt();
24         int c = a / b;
25         System.out.println("Thuong cua 2 so la: " + c);
26         System.out.println("BKAP - Exception Demo - End");
27     }
28
29 }
30
```

Output x

Debugger Console x ExceptionDemo (run) x

run:

BKAP - Exception Demo - Start

Nhap vao so thuc thu 1:

5

Nhap vao so thuc thu 2:

2

Thuong cua 2 so la: 2

BKAP - Exception Demo - End

BUILD SUCCESSFUL (total time: 3 seconds)

```
14 public class ExceptionDemo {
15
16     public static void main(String[] args) {
17         System.out.println("BKAP - Exception Demo - Start");
18         int a, b;
19         Scanner sc = new Scanner(System.in);
20         System.out.println("Nhap vao so thuc thu 1: ");
21         a = sc.nextInt();
22         System.out.println("Nhap vao so thuc thu 2: ");
23         b = sc.nextInt();
24         int c = a / b;
25         System.out.println("Thuong cua 2 so la: " + c);
26         System.out.println("BKAP - Exception Demo - End");
27     }
28
29 }
30
```

Output x

Debugger Console x ExceptionDemo (run) x

run:

BKAP - Exception Demo - Start

Nhap vao so thuc thu 1:

5

Nhap vao so thuc thu 2:

0

Exception in thread "main" java.lang.ArithmeticException: / by zero
at exceptiondemo.ExceptionDemo.main(ExceptionDemo.java:24)
C:\Users\ADMIN\AppData\Local\NetBeans\Cache\8.2\executor-snippets\run.xml:53: Java returned: 1

BUILD FAILED (total time: 2 seconds)

- Các loại lỗi ngoại lệ

- Checked Exception:**

- Xảy ra tại thời điểm compile – thời điểm chương trình biên dịch.
- Thường liên quan đến lỗi cú pháp (syntax)
- Bắt buộc phải quản lý

- Unchecked Exception:**

- Xảy ra trong thời điểm runtime – thời điểm chương trình đang chạy
- Thường liên quan đến lỗi logic
- Không bắt buộc phải quản lý

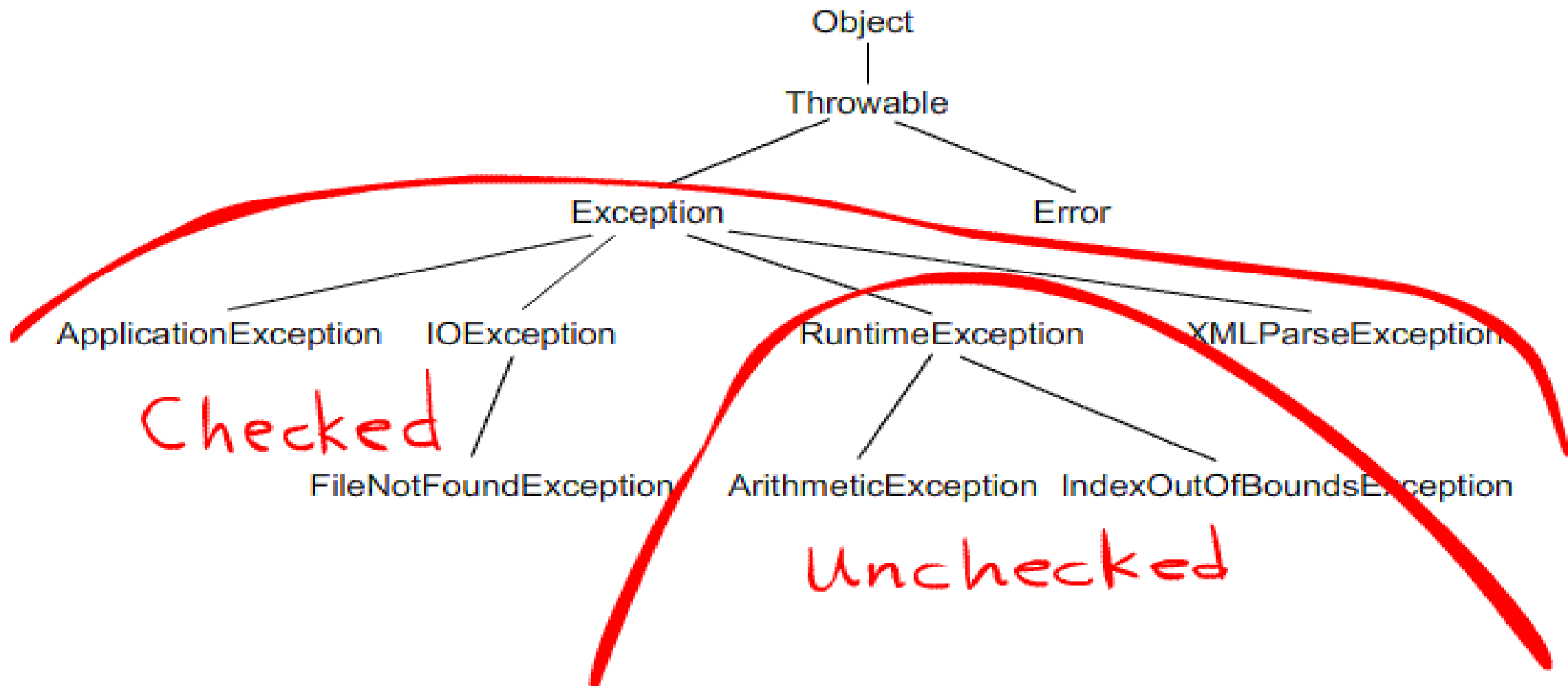
```
12 public class CheckedException {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         System.out.println(ABC);
19     }
20 }
```

```
12 public class UncheckedException {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         System.out.println("Start");
19         int a = 10, b = 0;
20         int c = a/b;
21         System.out.println("Thuong 2 so la: "+c);
22         System.out.println("End");
23     }
24 }
25
26
```

Output ×

Debugger Console × ExceptionDemo (run) ×

run:
Start
Exception in thread "main" java.lang.ArithmeticException: / by zero
at exceptiondemo.UncheckedException.main(UncheckedException.java:20)
C:\Users\ADMIN\AppData\Local\NetBeans\Cache\8.2\executor-snippets\run.xml:53: Java returned: 1
BUILD FAILED (total time: 1 second)



- Quản lý lỗi ngoại lệ
 - Sử dụng khối **try...catch...finally**

try{

Block Statement;

catch(**ExceptionType** exceptionObjectName){

Block Statement (Handling exception – Error Message);

finally{

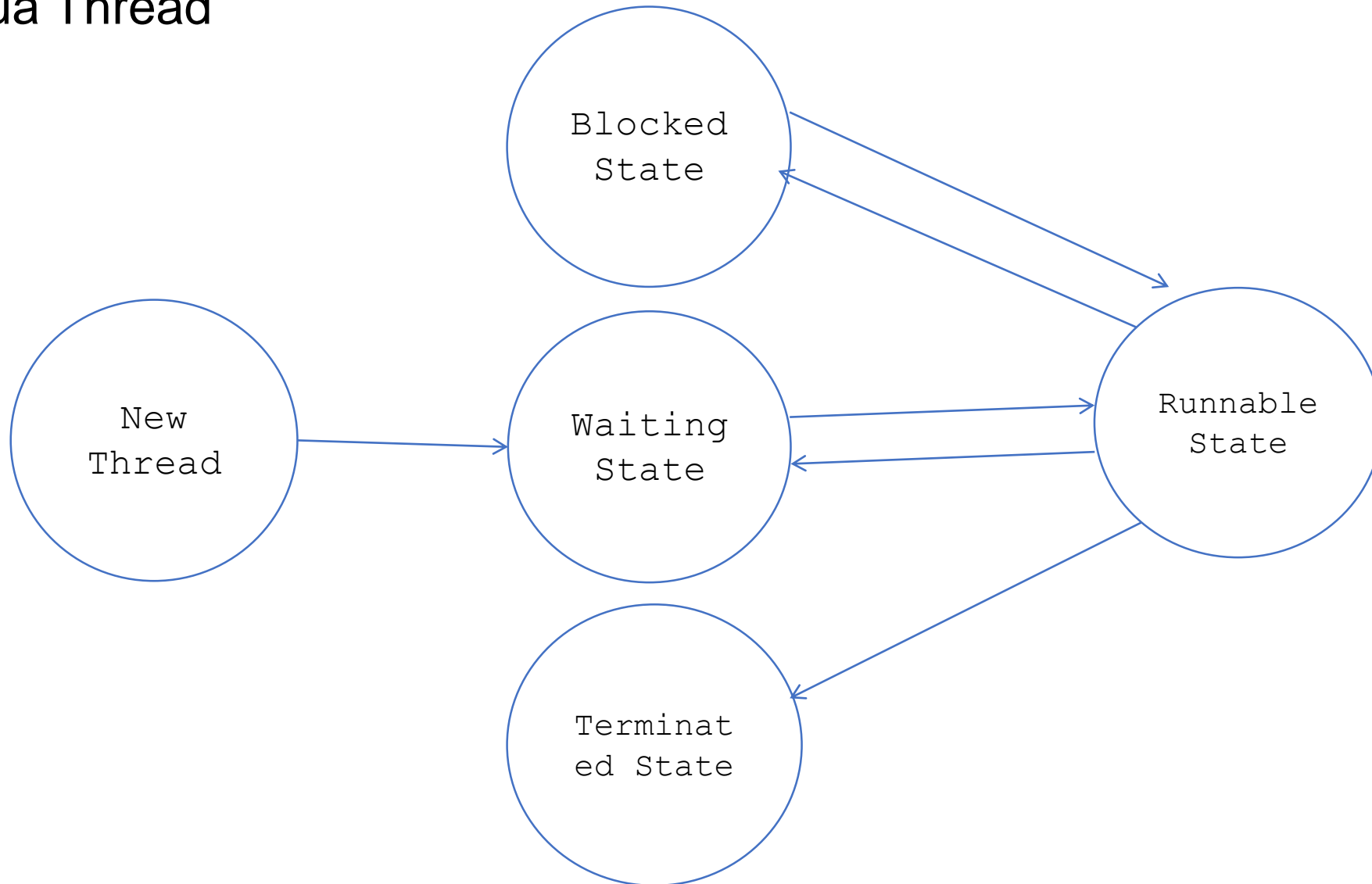
Block Statement;

}

- Sử dụng từ khóa **throw** và **throws**
 - **Throw**: Tự tung một lỗi ngoại lệ - bên trong phương thức
 - **Throws**: Đẩy Exception cho nơi khác xử lý

- **Multitasking** – Đa nhiệm: là khả năng chạy đồng thời một hoặc nhiều chương trình cùng một lúc trên một hệ điều hành. Có 2 cách:
 - **Multiprocessing**: Đa nhiệm dựa trên tiến trình – Đa tiến trình
 - Mỗi tiến trình có địa chỉ riêng trong bộ nhớ
 - Tiến trình xử lý nặng
 - Sự giao tiếp giữa các tiến trình yêu cầu cao
 - Chuyển đổi tiến trình đòi hỏi thời gian để đăng ký việc lưu và tải các bộ nhớ, danh sách cập nhật
 - **MultiThreading**: Đa nhiệm dựa trên luồng – Đa luồng
 - Các luồng chia sẻ không gian địa chỉ ô nhớ
 - Luồng xử lý nhẹ
 - Sự giao tiếp giữa các luồng có yêu cầu thấp

- Vòng đời của Thread



- Cách tạo luồng
 - Kế thừa (Extend) từ lớp **Thread**
 - B1: Khai báo một lớp mới kế thừa từ lớp **Thread**
 - B2: Override phương thức **run** – phương thức chạy khi luồng bắt đầu chạy, kết thúc thì luồng tự hủy
 - B3: Tạo instance cho lớp vừa khai báo
 - B4: Gọi phương thức **start** của đối tượng để bắt đầu thực thi
 - Kế thừa (implement) từ **Runnable** Interface
 - B1: Khai báo lớp mới kế thừa interface **Runnable**
 - B2: Triển khai phương thức **run** - phương thức chạy khi luồng bắt đầu chạy, kết thúc thì luồng tự hủy
 - B3: Tạo instance cho lớp vừa khai báo
 - B4: Gọi phương thức **start** của đối tượng để bắt đầu thực thi

- **Các phương thức quan trọng trong thread:**

- `getName()`: Trả về tên của thread
- `start()`: Bắt đầu chạy luồng
- `run()`: Thực hiện chức năng của luồng
- `suspend()`: Tạm thời dừng hoạt động của luồng bằng cách ngưng cung cấp CPU cho luồng
- `resume()`: cho luồng chạy lại khi luồng bị dừng bằng `suspend`
- `stop()`: ép buộc luồng kết thúc
- `destroy()`: dừng hẳn luồng
- `isAlive()`: kiểm tra luồng còn active ko
- `sleep(long)`: tạm dừng luồng trong một khoảng thời gian milisecond
- `interrupt()`: kiểm tra thread hiện tại đã bị ngắt chưa
- `setPriority(int)`: set độ ưu tiên
- `getPriority()`: trả lại độ ưu tiên
- `join()`: thông báo luồng cha phải chờ luồng con chạy xong rồi mới được chạy
- `join(long)`: Thread cha cần đợi millisecond mới được tiếp tục chạy

- Độ ưu tiên là số nguyên dao động giữa: **MIN_PRIORITY** và **MAX_PRIORITY**.
- Luồng có chỉ số ưu tiên cao được CPU ưu tiên xử lý hơn.
- Các mức ưu tiên:
 1. **Thread.MAX_PRIORITY**: Hằng số 10, cao nhất.
 2. **Thread.NORM_PRIORITY**: Hằng số 5, trung bình.
 3. **Thread.MIN_PRIORITY**: Hằng số 1. thấp nhất.

HỎI ĐÁP





TRẢI NGHIỆM THỰC HÀNH



TRÂN TRỌNG CẢM ƠN!



238 Hoàng Quốc Việt, Bắc Từ Liêm, Hà Nội



0968.27.6996



tuyensinh@bachkhoa-aptech.edu.vn



www.bachkhoa-aptech.edu.vn