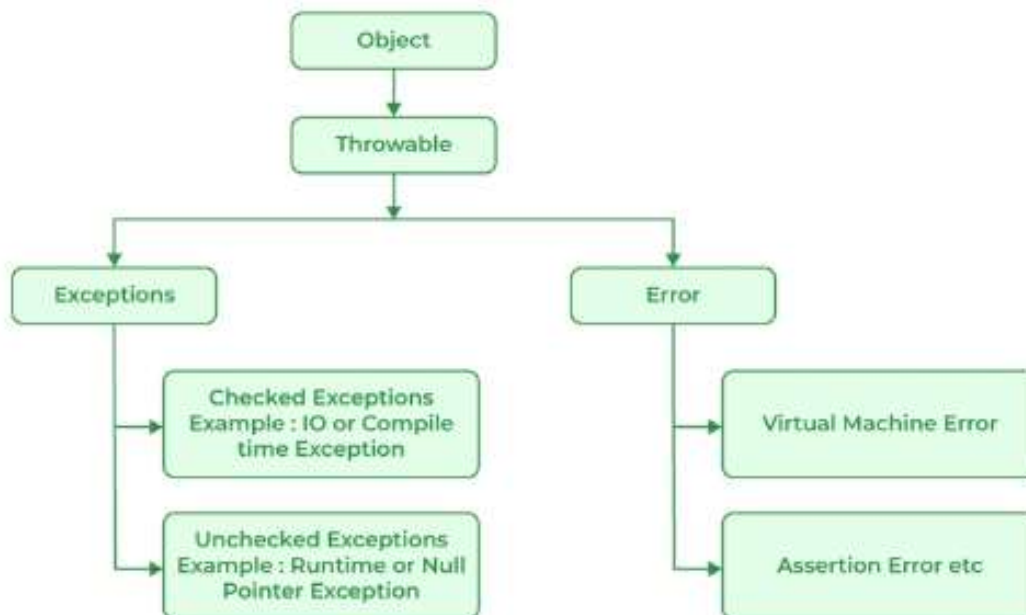


🕒 Cập nhật tháng 8 năm 2024

[Bài Đọc] Phân loại ngoại lệ trong java

Ngoại lệ (Exception) trong Java được chia thành nhiều loại để hỗ trợ lập trình viên xử lý lỗi hiệu quả hơn. Tất cả các ngoại lệ trong Java đều là các đối tượng kế thừa từ lớp gốc Throwable. Dựa trên tính chất và cách xử lý, ngoại lệ trong Java được phân loại thành 3 loại chính: **Checked Exception**, **Unchecked Exception**, và **Error**.

1. Phân cấp các lớp Ngoại lệ trong Java



2. Checked Exception

- Khái niệm:
 - Checked Exception là các ngoại lệ được kiểm tra trong quá trình biên dịch (compile-time)
 - Java yêu cầu lập trình viên phải xử lý hoặc khai báo các ngoại lệ này, nếu không chương trình sẽ không biên dịch được
- Ví dụ:
 - IOException
 - SQLException
 - FileNotFoundException

- `ClassNotFoundException`
- Đặc điểm:
 - Thường xảy ra khi có lỗi liên quan đến I/O, cơ sở dữ liệu, hoặc các lỗi phụ thuộc bên ngoài
 - Lập trình viên phải sử dụng từ khóa `try-catch` hoặc `throws` để xử lý
- Ví dụ:

```
public class CheckedExceptionExample { new *
    public static void main(String[] args) { new *
        try {
            File file = new File( pathname: "nonexistent.txt");
            FileReader fr = new FileReader(file); // Gây ra FileNotFoundException
        } catch (Exception e) {
            System.out.println("File không tồn tại: " + e.getMessage());
        }
    }
}
```

3. Unchecked Exception

- Khái niệm:
 - Unchecked Exception là các ngoại lệ xảy ra tại thời gian chạy (runtime)
 - Không bắt buộc phải xử lý hoặc khai báo trong mã, nhưng nên tránh các lỗi này bằng cách viết mã cẩn thận
- Ví dụ:
 - `ArithmeticException`
 - `NullPointerException`
 - `ArrayIndexOutOfBoundsException`
 - `IllegalArgumentException`
- Đặc điểm:
 - Là kết quả của lỗi logic trong chương trình (ví dụ: chia cho 0, truy cập phần tử mảng không hợp lệ)
 - Không bắt buộc xử lý, nhưng lập trình viên cần đảm bảo tránh để xảy ra các lỗi này
- Ví dụ:

```
public class UncheckedExceptionExample { new *
    public static void main(String[] args) { new *
        int[] numbers = {1, 2, 3};
        try {
            System.out.println(numbers[5]); // Gây ra ArrayIndexOutOfBoundsException
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Truy cập chỉ mục không hợp lệ: " + e.getMessage());
        }
    }
}
```

4. Error

- Khái niệm:

- Error là các lỗi nghiêm trọng liên quan đến hệ thống hoặc môi trường thực thi
- Các lỗi này không thể hoặc không nên xử lý trong chương trình
- Ví dụ:
 - StackOverflowError
 - OutOfMemoryError
 - VirtualMachineError
- Đặc điểm:
 - Thường liên quan đến lỗi hệ thống hoặc tài nguyên, ví dụ như bộ nhớ bị đầy hoặc vòng lặp đệ quy quá sâu
 - Không được thiết kế để xử lý bằng try-catch
- Ví dụ:

```
public class ErrorExample { new *
    public static void main(String[] args) { new *
        causeStackOverflow();
    }

    public static void causeStackOverflow() { 2usages new *
        causeStackOverflow(); // Gây ra StackOverflowError
    }
}
```

5. So sánh Checked Exception, Unchecked Exception và Error

Đặc điểm	Checked Exception	Unchecked Exception	Error
Thời điểm xảy ra	Compile-time	Runtime	Runtime
Bắt buộc xử lý	Có	Không	Không
Nguyên nhân phổ biến	Lỗi nhập/xuất, cơ sở dữ liệu	Lỗi logic, lập trình sai	Lỗi hệ thống, tài nguyên
Ví dụ	IOException, SQLException	NullPointerException, ArithmeticException	OutOfMemoryError, StackOverflowError

Link tài nguyên đọc thêm: <https://www.geeksforgeeks.org/exceptions-in-java/>