

🕒 Cập nhật tháng 8 năm 2024

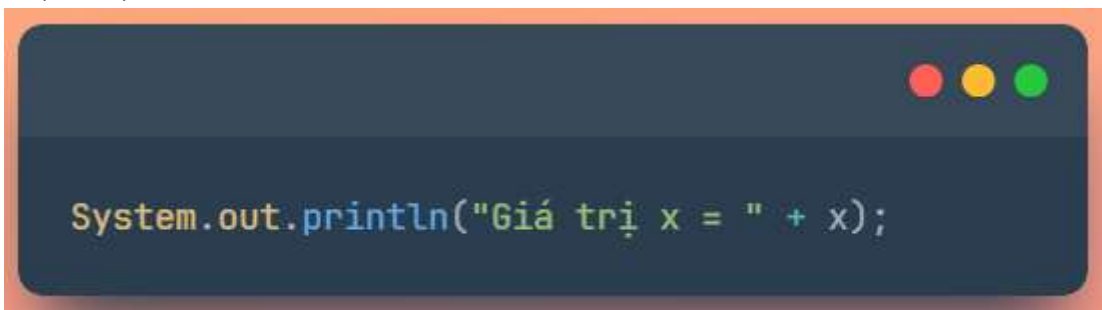
[Bài đọc] Debug

1. Debug là gì?

- Định nghĩa:
 - **Debug** là quá trình tìm kiếm, phân tích và sửa lỗi (bugs) trong mã nguồn chương trình.
 - Thay vì chỉ in ra giá trị bằng `System.out.println()`, debug cho phép bạn **quan sát trạng thái của chương trình từng bước một**

2. Các cách debug trong Java

- Sử dụng `System.out.println()`
 - Cách thủ công, đơn giản nhất để theo dõi giá trị biến hoặc dòng chương trình nào đang được thực thi



- Dễ dùng - Khó kiểm soát với chương trình lớn, không nên dùng trong sản phẩm thực tế

• Sử dụng **Debugger** của IDE (Eclipse, IntelliJ, VS Code)

- Các bước cơ bản:
 - **Đặt breakpoint**: là điểm dừng tại dòng code muốn quan sát
 - **Chạy chế độ Debug** (F5 hoặc Shift+F9 tùy IDE)
 - **Sử dụng các chức năng**:
 - **Step Over** (F6): đi đến dòng tiếp theo, không vào hàm
 - **Step Into** (F5): đi sâu vào bên trong hàm đang gọi
 - **Step Out** (F7): thoát khỏi hàm hiện tại
 - **Resume** (F8): chạy tiếp đến breakpoint kế tiếp
- Quan sát được:
 - Biến cục bộ và toàn cục
 - Giá trị tham số truyền vào hàm
 - Stack trace (ngăn xếp lệnh)
 - Cấu trúc object (mở rộng từng thuộc tính của object)

3. Ví dụ nhỏ

```
public class DebugExample {
    public static void main(String[] args) {
        int a = 5, b = 0;
        int result = divide(a, b); // breakpoint tại dòng này
        System.out.println("Kết quả: " + result);
    }

    public static int divide(int x, int y) {
        return x / y;
    }
}
```

- Nếu debug, bạn sẽ thấy chương trình **gây lỗi chia cho 0** tại `x / y`
- IDE sẽ **dừng tại breakpoint** và báo lỗi `ArithmeticException`

4. Lợi ích khi dùng Debug

Lợi ích	Mô tả
Quan sát giá trị biến	Biết chính xác tại sao kết quả sai
Hiểu luồng chạy	Đặc biệt hữu ích với các hàm đệ quy, vòng lặp
Tìm lỗi logic	Không phải lỗi cú pháp, mà lỗi tư duy
Tiết kiệm thời gian	So với việc đoán lỗi qua in log

5. Công cụ & Mẹo debug nâng cao

- **Conditional Breakpoint**: dừng khi biến đạt giá trị nhất định
- **Watch Expressions**: theo dõi biểu thức hoặc biến cụ thể
- **Evaluate Expression**: thử chạy đoạn code nhỏ ngay khi đang debug
- **Logging Breakpoint**: thay vì dừng, ghi log ra console mà không cần viết `System.out.println()`