

Bài 15

Debug

Module: ADVANCED PROGRAMMING WITH JAVA

Mục tiêu

- Giải thích các bước để debug một ứng dụng Java
- Debug được ứng dụng Java

Debug ứng dụng Java

Debug

- Debug (dò lỗi) là quá trình tìm kiếm và sửa chữa các lỗi trong một chương trình
- Các loại lỗi thường gặp:
 - Lỗi cú pháp (syntax error) dễ phát hiện bởi vì compiler đưa ra thông báo
 - Lỗi thực thi (runtime error) cũng dễ phát hiện bởi vì interpreter cũng hiển thị lỗi trên màn hình
 - Lỗi logic thường khó phát hiện hơn
- Các lỗi logic còn được gọi là bug
- Thông thường, cần kết hợp nhiều cách để tìm được bug

Một số phương pháp debug

- Đọc mã nguồn (hand-trace)
- Chèn các lệnh in ra các giá trị trong từng đoạn của chương trình để kiểm tra các giá trị và việc thực thi các câu lệnh
- Sử dụng debugger: một chương trình cho phép quan sát quá trình thực thi của một ứng dụng
- Java cung cấp sẵn một debugger là **jdb** (cũng là một ứng dụng Java)
- Các IDE thông thường cũng tích hợp sẵn debugger

Các tính năng của debugger

- Thực thi riêng lẻ từng câu lệnh tại một thời điểm
- Theo dõi hoặc bỏ qua luồng thực thi vào sâu bên trong một phương thức
- Đặt các breakpoint: các điểm mà chương trình sẽ dừng lại để lập trình viên quan sát trạng thái hiện tại của chương trình
- Hiển thị giá trị của các biến tại từng thời điểm
- Hiển thị ngăn xếp các phương thức được gọi
- Thay đổi giá trị của biến tại thời điểm thực thi (chỉ một số debugger hỗ trợ tính năng này)

Stack trace

- Stack trace thể hiện lần lượt các phương thức được gọi
- Sử dụng stack trace hỗ trợ việc tìm ra loại lỗi và vị trí lỗi
- Ví dụ:
 - Lỗi chia cho 0
 - Vị trí lỗi: dòng số 7, bên trong phương thức *divide()*


```
public static int divide(int a, int b){  
    return 1 / (a - b);  
}
```

```
public static void main(String[] args) {  
    int a = 1;  
    int b = 1;  
    int c = divide(a, b);  
}
```

```
Exception in thread "main" java.lang.ArithmeticException: / by zero  
    at SystemTime.divide(SystemTime.java:7)  
    at SystemTime.main(SystemTime.java:14)
```

```
Process finished with exit code 1
```

Debug với IntelliJ IDEA

- Khởi chạy chế độ debug bằng phím (^D) hoặc nút 
- Đặt breakpoint
- Xem thông tin chi tiết tại điểm breakpoint
- Step into (quan sát sâu vào bên trong một phương thức được gọi)
- Step over (không quan sát bên trong một phương thức được gọi)

[Thực hành] Giải phương trình bậc nhất

[Thực hành] Tính số ngày của tháng

[Bài tập] Giải phương trình bậc hai

[Bài tập] Đọc số thành chữ

[Bài tập] Debug ứng dụng

Tổng kết

- switch-case là một cấu trúc điều kiện cho phép lựa chọn thực thi các khối lệnh khác nhau dựa trên kết quả của việc so sánh
- Cần lựa chọn sử dụng switch-case và if phù hợp với từng tình huống
- switch-case có thể sử dụng với kiểu dữ liệu String
- switch-case có thể sử dụng với kiểu dữ liệu enum
- Debug là quá trình tìm kiếm và sửa lỗi
- Java cung cấp sẵn debugger
- IntelliJ IDEA và các IDE khác thường tích hợp sẵn debugger

Hướng dẫn

- Hướng dẫn làm bài thực hành và bài tập
- Chuẩn bị bài tiếp: Exception Handling