Bài 8.4: Các khuyến nghị trong xử lý ngoại lệ

- ✓ Các tip để có best practices trong xử lý ngoại lệ

Best practices 1

Sử dụng try-catch-finally để khắc phục lỗi, giải phóng tài nguyên khi chương trình đang hoạt động mà gặp sự cố.

- ✓Chỉ bắt ngoại lệ khi bạn biết cách khắc phục sự cố xảy ra bởi ngoại lệ. Nếu không hãy để các phương thức trong chuỗi lời gọi lưu trong stack thực hiện điều này nếu nó có thể.
- ✓Các kiểu ngoại lệ càng nằm sâu trong chuỗi kế thừa càng cần phải đặt ở đầu trong các khối catch. Bởi vì khi ngoại lệ cha đứng trước, nó sẽ được xử lý trước và ngoại lệ cụ thể hơn không được xử lý.
- ✓Dọn dẹp và giải phóng tài nguyên bằng sử dụng câu lệnh using hoặc khối finally. Ưu tiên sử dụng using để tự động dọn dẹp khi ngoại lệ được kích hoạt. Nếu tài nguyên đang sử dụng mà không triển khai interface Idispoable thì sử dụng khối finally để dọn dẹp.

Best practices 2

Xử lý các điều kiện thông thường không cần văng ngoại lệ.

- ✓Với các hành động xảy ra có thể dẫn tới ngoại lệ, cố gắng tìm cách giải quyết nó mà không cần kích hoạt ngoại lệ. Ví dụ như cố đóng 1 kết nối đã đóng sẽ dẫn đến ngoại lệ. Để tránh, ta dùng câu lệnh if kiểm tra trước khi thực hiện điều này.
- ✓Nếu sự kiện nào đó ít khi xảy ra và nó thực sự chỉ ra một lỗi nào đó, sử dụng ngoại lệ.
- ✓Nếu sự kiện thường xuyên xảy ra, ta có thể coi đó là 1 phần của các hành động thực thi thông thường khác để hạn chế ngoại lệ.

Best practices 3

Thiết kế các lớp để có thể tránh ngoại lệ.

- ✓Với các lớp, có thể cung cấp các phương thức, thuộc tính cho phép tránh các lời gọi có thể xảy ra ngoại lệ. Ví dụ lớp đọc file có phương thức kiểm tra sự kết thúc của file trước khi đọc.
- ✓Một cách khác là trả về giá trị null khi xảy ra lỗi thay vì văng ngoại lệ. Làm vậy sẽ giảm ảnh hưởng tối đa của sự cố tới hiệu năng của ứng dụng.

Best practices 4

Văng ngoại lệ thay vì trả về một mã lỗi.

✓Ngoại lệ đảm bảo rằng sự cố xảy ra sẽ được thông báo rõ ràng vì ngay cả khi nơi gọi phương thức không kiểm tra giá trị trả về.

Best practices 5

Sử dụng ngoại lệ có sẵn của thư viện .NET.

- ✓Chỉ nên tạo lớp ngoại lệ mới khi các ngoại lệ có sẵn không đáp ứng được mục đích sử dụng.
- ✓Ví dụ, văng ngoại lệ InvalidOperationException khi setter hoặc lời gọi không phù hợp để cung cấp trạng thái hiện tại của đối tượng.
- ✓Sử dụng ArgumentException hoặc một lớp con của nó nếu tham số không hợp lệ được truyền vào lời gọi.

Best practices 6

Kết thúc tên lớp ngoại lệ tự định nghĩa với hậu tố Exception.

✓Khi tạo ngoại lệ tự định nghĩa, luôn kết thúc tên của lớp ngoại lệ đó bằng Exception.

Best practices 7

Trong lớp ngoại lệ tự định nghĩa phải chứa ít nhất 3 constructor.

- ✓Một constructor không tham số sử dụng các giá trị mặc định.
- ✓Một constructor nhận 1 string làm tham số.
- ✓Một constructor nhận 1 string và 1 inner exception làm tham số.
- ✓Ngoài ra nếu ứng dụng hoạt động trong môi trường remote ta cần đảm bảo siêu dữ liệu về ngoại lệ có sẵn tại thời điểm cần sử dụng. Lúc này cần phải tuần tự hóa, bổ sung thêm constructor có serialiable.

Best practices 8

Sử dụng các thông báo lỗi đúng ngữ pháp.

✓Viết câu từ thông báo lỗi ngắn gọn, rõ nghĩa và kết thúc bằng dấu chấm.

✓Ví dụ: “Chỉ số phần tử đã vượt biên của mảng”.

Best practices 9

Bao gồm thông báo cục bộ trong mỗi exception.

✓Thông báo lỗi mà bạn nhìn thấy là được kế thừa từ lớp cha, không phải từ lớp ngoại lệ.

✓Với các ứng dụng cục bộ, bạn nên cung cấp thông điệp cục bộ cho mỗi ngoại lệ mà ứng dụng có thể sinh ra.

Best practices 10

Trong ngoại lệ tự định nghĩa bạn có thể cung cấp thêm các thuộc tính nếu cần thiết.

✓Bổ sung các thuộc tính cho ngoại lệ chỉ khi các thông tin đó là hữu ích.

✓Ví dụ lớp FileNotFoundException bổ sung thuộc tính tên file.

Best practices 11

Sử dụng phương thức builder để tạo ngoại lệ.

- ✓Nếu trong một lớp văng cùng một ngoại lệ tại nhiều nơi khác nhau trong triển khai của nó. Có thể cân nhắc sử dụng phương thức hỗ trợ tạo và trả về ngoại lệ đó.
- ✓Mục đích là để tránh việc lặp lại code nhiều lần.

Best practices 12

Khôi phục trạng thái khi phương thức chạy không hoàn chỉnh do ngoại lệ.

- ✓Nơi gọi phương thức nên giả sử rằng không bị ảnh hưởng bởi ngoại lệ xảy ra trong phương thức đang gọi.
- ✓Ví dụ điển hình là giao dịch rút tiền. Nếu đang xử lý rút tiền mà bị lỗi, ta không nhận được tiền, do đó tài khoản phải đảm bảo không bị trừ tiền.

```
private static void TransferFunds(Account from, Account to, decimal amount)
{
    string withdrawalTrxID = from.Withdrawal(amount);
    try
    {
        to.Deposit(amount);
    }
    catch
    {
        from.RollbackTransaction(withdrawalTrxID);
        throw;
    }
}
```

Nội dung tiếp theo

Thao tác với file text, JSON, CSDL