

Bài 8.1: Tổng quan về ngoại lệ

- ✓ Khái niệm và mục đích sử dụng
- ✓ Các đặc trưng của ngoại lệ
- ✓ Ví dụ minh họa

Khái niệm và mục đích sử dụng

- ✓ Ngoại lệ(exception) là các lỗi không mong muốn xảy ra trong quá trình chương trình ứng dụng đang chạy và có thể xử lý được bằng chương trình lập trình.
- ✓ Error(lỗi) là các sự cố nghiêm trọng không thể khắc phục bằng chương trình như hỏng ổ cứng, cháy CPU khi đang hoạt động...
- ✓ Xử lý ngoại lệ sử dụng các keyword throw, try, catch, finally để tạo, khoanh vùng, xử lý ngoại lệ và dọn dẹp sau khi ngoại lệ xảy ra.
- ✓ Ngoại lệ có thể được sinh ra bởi trình chạy của ngôn ngữ CLR, bởi thư viện .NET hoặc thư viện bên thứ ba hoặc mã nguồn của ứng dụng.
- ✓ Khi ngoại lệ xảy ra, CLR sẽ tìm khối catch trong chuỗi các lời gọi phương thức để xem ngoại lệ đó có được xử lý tường minh không.
- ✓ Nếu ngoại lệ xảy ra và được xử lý, chương trình vẫn tiếp tục hoạt động bình thường ngoại trừ chức năng xảy ra ngoại lệ. Nếu không, chương trình sẽ kết thúc ngay lập tức với một thông báo ngoại lệ tương ứng.

Các đặc trưng của ngoại lệ

- ✓ Tất cả các lớp ngoại lệ đều là hậu duệ của `System.Exception`.
- ✓ Sử dụng khối try để bao bọc đoạn lệnh có thể xảy ra ngoại lệ.
- ✓ Khi ngoại lệ xảy ra trong khối try, luồng chương trình sẽ chuyển cho trình xử lý ngoại lệ đầu tiên được liên kết với ngoại lệ đó trong chuỗi các lời gọi tính từ phương thức xảy ra ngoại lệ về trước.
- ✓ Khối catch là nơi xử lý ngoại lệ khi ngoại lệ tương ứng xảy ra trong khối try.
- ✓ Nếu không có khối catch nào ứng với ngoại lệ đã xảy ra trong khối try, chương trình sẽ kết thúc với một thông điệp lỗi.
- ✓ Không bắt một ngoại lệ trừ khi bạn có thể xử lý nó và đặt chương trình vào trạng thái có thể kiểm soát.
- ✓ Nếu bạn bắt ngoại lệ `System.Exception`, hãy ném lại nó ở cuối khối catch với `throw`.

Các đặc trưng của ngoại lệ

- ✓ Nếu khối catch định nghĩa một biến có kiểu của ngoại lệ, bạn có thể sử dụng nó để quản lý các thông tin liên quan đến kiểu ngoại lệ đã xảy ra.
- ✓ Sinh ngoại lệ tường minh bằng keyword throw.
- ✓ Các đối tượng ngoại lệ có chứa thông tin chi tiết về lỗi đã xảy ra như trạng thái của stack chứa lời gọi phương thức khi xảy ra ngoại lệ, thông điệp mô tả của ngoại lệ.
- ✓ Các lệnh trong khối finally luôn được thực hiện cho dù có xảy ra ngoại lệ hay không. Do đó khối finally là nơi để giải phóng tài nguyên và dọn dẹp (đóng luồng file, kết nối database,...).
- ✓ Việc quản lý ngoại lệ trong .NET được triển khai ở trên đỉnh của cấu trúc kỹ thuật Win32.

Ví dụ

```
1 reference
static double Div(int a, int b)
{
    if(b == 0)
    { // tại đây tạo và ném ra một ngoại lệ
        throw new DivideByZeroException("Loi chia cho 0");
    }
    return a / b;
}

0 references
static void Main()
{
    int a = 20;
    int b = 0;
    try // gói đoạn lệnh có thể xảy ra ngoại lệ
    {
        Console.WriteLine($"{a} / {b} = {Div(a, b)}");
    } catch(DivideByZeroException e) // bắt và xử lý ngoại lệ
    {
        Console.WriteLine(e.Message);
        Console.WriteLine(e.StackTrace);
    }
}
```

```
C:\WINDOWS\system32\cmd.exe
Loi chia cho 0
   at Chapter8.Lesson81.Div(Int32 a, Int32 b)
   in C:\Users\trieu\source\repos\CSharpCourse\Chapter8\Lesson81.cs:line 11
   at Chapter8.Lesson81.Main() in C:\Users\trieu\source\repos\CSharpCourse\Chapter8\Lesson81.cs:line 22
Press any key to continue . . .
```



Nội dung tiếp theo

Xử lý ngoại lệ