

C# Programming

Xử lý lỗi ngoại lệ trong C#

By Hoàng Hữu Việt

Email: viethh@vinhuni.edu.vn

Viện Kỹ thuật và Công nghệ, Đại học Vinh

Cao Thanh Sơn

Email: sonct@vinhuni.edu.vn

Viện Nghiên cứu và Đào tạo trực tuyến

Vinh, 2022

Mục đích, chuẩn đầu ra và nội dung

- Mục đích
 - Giới thiệu về kỹ thuật xử lý ngoại lệ (exception handling) trong C#.
- Chuẩn đầu ra
 - Giải quyết các lỗi ngoại lệ sử dụng các lớp của .NET framework
 - Phát triển kỹ năng lập trình, kỹ năng tìm kiếm và đọc hiểu tài liệu
- Nội dung
 - Lỗi ngoại lệ
 - Thủ tục xử lý lỗi ngoại lệ
 - Lớp Exception của .NET

Lỗi ngoại lệ

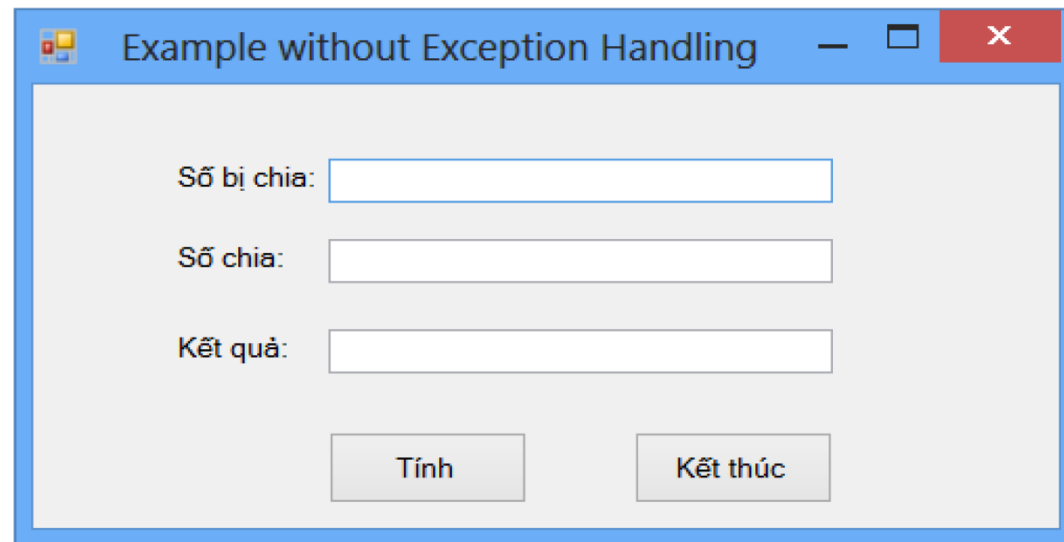
- Một lỗi ngoại lệ là lỗi không mong đợi xảy ra khi ứng dụng đang thực hiện.
- Một số nguyên nhân xảy ra lỗi ngoại lệ
 - ❑ Truy nhập phần tử ngoài kích thước mảng.
 - ❑ Thực hiện các phép toán tràn số.
 - ❑ Thực hiện phép chia cho 0.
 - ❑ Xung đột khi gọi các lệnh của các thư viện lập trình.
 - ❑ Tài nguyên không đủ khi thực hiện chương trình.
- Xử lý lỗi ngoại lệ cho phép tạo ra các ứng dụng mà có thể giải quyết các vấn đề lỗi ngoại lệ.

Lỗi ngoại lệ

- Với các ngôn ngữ lập trình không hỗ trợ kỹ thuật xử lý lỗi ngoại lệ:
 - Người lập trình thường phải dành nhiều thời gian để xử lý các lỗi có thể xảy ra trong ứng dụng.
 - Kéo dài thời gian phát triển ứng dụng.
- Ngôn ngữ lập trình C# hỗ trợ kỹ thuật xử lý các lỗi ngoại lệ dựa trên lớp Exception thuộc không gian tên System.

Lỗi ngoại lệ

- Ví dụ chia 2 số nguyên:

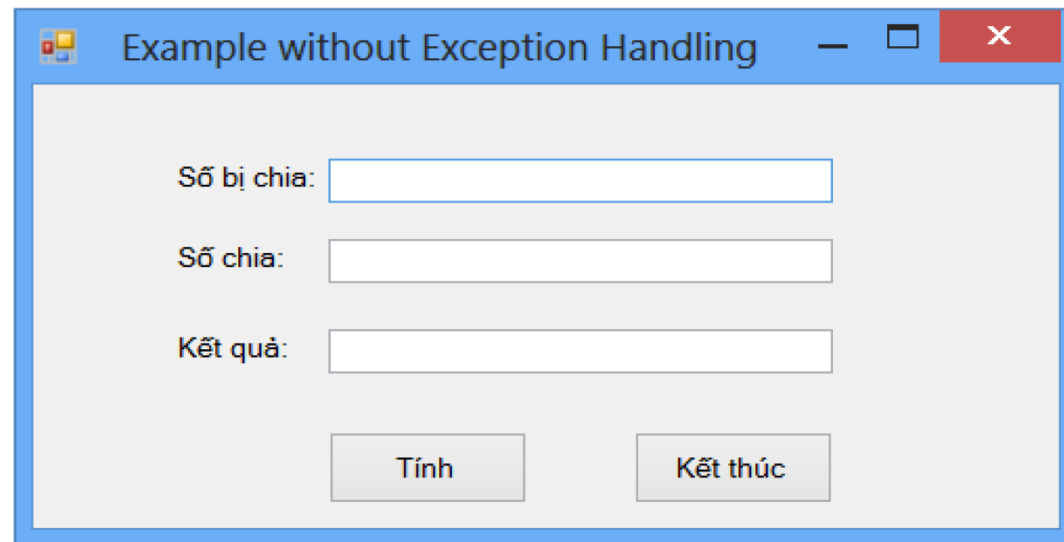


The screenshot shows a standard Windows application window with a blue title bar. The title bar text is "Example without Exception Handling". Inside the window, there is a light gray background. Three text input fields are arranged vertically, each preceded by a label: "Số bị chia:", "Số chia:", and "Kết quả:". Below these fields, there are two buttons: "Tính" on the left and "Kết thúc" on the right.

```
private void btnDivide_Click(object sender, EventArgs e)
{
    int a = Convert.ToInt32(txtNumber1.Text);
    int b = Convert.ToInt32(txtNumber2.Text);
    int c = a/b;
    txtResult.Text = c.ToString();
}
```

Lỗi ngoại lệ

- Ví dụ chia 2 số nguyên:



The screenshot shows a standard Windows application window with a blue title bar containing the text 'Example without Exception Handling' and standard window control buttons (minimize, maximize, close). The main content area is light gray and contains three text input fields stacked vertically. The first field is preceded by the label 'Số bị chia:', the second by 'Số chia:', and the third by 'Kết quả:'. Below these fields are two rectangular buttons. The left button is labeled 'Tính' and the right button is labeled 'Kết thúc'.

- Nhập dữ liệu cho các trường hợp:
 - ❑ Số bị chia là 50 và số chia là 4, kết quả?
 - ❑ Số bị chia là số nguyên và số chia bằng 0, kết quả?
 - ❑ Số bị chia là số nguyên và số chia là xâu ký tự, kết quả?

Thủ tục xử lý lỗi ngoại lệ

```
try
{
    <các lệnh có thể xảy ra lỗi>;
} catch (exception 1)
{
    <xử lý lỗi>;
} catch (exception 2)
{
    <xử lý lỗi>;
} ... catch (exception n)
{
    <xử lý lỗi>;
} finally
{
    <các lệnh thực hiện cho cả khối try và khối catch>;
}
```

Thủ tục xử lý lỗi ngoại lệ

■ Ví dụ chia 2 số nguyên

```
private void btnDivide_Click(object sender, EventArgs e){
try
{
    int a = Convert.ToInt32(txtNumber1.Text);
    int b = Convert.ToInt32(txtNumber2.Text);
    int c = a/b;
    txtResult.Text = result.ToString();
} catch (DivideByZeroException myDivideByZeroException)
{
    MessageBox.Show(myDivideByZeroException.Message, "Zero..");
} catch (FormatException myFormatException)
{
    MessageBox.Show(myFormatException.Message, "Format..");
}
}
```


Lớp Exception của .NET

- Lớp Exception trong không gian tên System là lớp cơ sở của các lớp xử lý lỗi ngoại lệ.
- Các lớp xử lý lỗi ngoại lệ đều phải được kế thừa từ lớp Exception
 - Lớp DivideByZeroExceptions và FormatExceptions.
- Tính chất đa hình của lớp kế thừa
 - Một khối catch có tham số là lớp Exception có thể nhận biết tất cả các lỗi của các lớp dẫn xuất của nó.
- Ví dụ chia 2 số có thể có các lỗi khác chưa được xử lý như lỗi tràn số (OverflowException):
 - Sử dụng lớp Exception làm tham số của khối catch là một cách để xử lý các lỗi.

Lớp Exception của .NET

- Xử lý lỗi ngoại lệ dùng lớp Exception:

```
private void btnDivide_Click(object sender, EventArgs e)
{
    try
    {
        int a = Convert.ToInt32(txtNumber1.Text);
        int b = Convert.ToInt32(txtNumber2.Text);
        int c = a/b;
        txtResult.Text = c.ToString();
    } catch (Exception myException)
    {
        MessageBox.Show(myException.Message, "Exception");
    }
}
```

Lớp Exception của .NET

- Các thuộc tính thường dùng của lớp Exception
 - Message: lưu trữ thông báo lỗi.
 - StackTrace: lưu trữ vị trí dòng lệnh và thủ tục sinh ra lỗi.
 - InnerException: lưu trữ vết của các vị trí dòng lệnh và các thủ tục sinh ra lỗi.
- Ví dụ tính căn bậc 2 - xem trong tài liệu tham khảo.

SquareRootTest

Nhập giá trị tính căn bậc 2:

Giá trị căn bậc 2:

Tính Kết thúc

Lớp Exception của .NET

- Một số lớp ngoại lệ thường được sử dụng
 - `ArgumentException`: Lỗi tham số hàm.
 - `ArithmeticException`: Lỗi liên quan đến phép toán.
 - `DivideByZeroException`: Lỗi chia cho 0.
 - `FormatException`: Lỗi định dạng dữ liệu.
 - `IndexOutOfRangeException`: Lỗi truy nhập chỉ số mảng.
 - `InvalidCastException`: Lỗi phép toán gán không hợp lệ.
 - `OutOfMemoryException`: Lỗi tràn bộ nhớ.
 - `OverflowException`: Lỗi tràn số.
 - `StackOverflowException`: Lỗi tràn stack.

Kết thúc



HỎI & TRẢ LỜI