

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG



CHƯƠNG 7 XỬ LÝ NGOẠI LỆ (EXCEPTION HANDLING)

ThS: Phạm Nguyễn Sơn Tùng

Email: pnstung@fit.hcmus.edu.vn

NỘI DUNG BÀI HỌC

1

Dẫn nhập

2

Xử lý bằng phương pháp try-throw-catch

3

Một vài ví dụ minh họa

4

Ứng dụng trong thực tế

5

Bài tập tại lớp

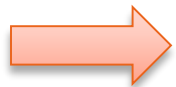
GIỚI THIỆU EXCEPTION HANDLING



GIỚI THIỆU EXCEPTION HANDLING

➤ **Báo lỗi:** Một số lỗi hay gặp khi lập trình.

- Syntax Error → Compiler
- Logical Error → Debugger
- Runtime Error.
 - Dữ liệu đầu vào xấu.
 - Tài nguyên thiếu hụt.



Try...Catch

GIỚI THIỆU EXCEPTION HANDLING

➤ **Cú pháp:** Một số lỗi hay gặp khi lập trình.

```
try
{
    // protected code
    throw
}
catch( ExceptionType e )
{
    // code to handle ExceptionType exception
}
```

GIỚI THIỆU EXCEPTION HANDLING

- **Try** (thử): Giống như một người giám sát (trọng tài), tìm kiếm những Exception (ngoại lệ) từ người dùng và Throw (ném) nó về một nơi để xử lý.
- **Throw** (ném): Dùng để cho người dùng ném một ngoại lệ. Người dùng có thể throw một câu thông báo hoặc 1 biến bất kỳ. Mỗi "throw" phải có ít nhất một "catch".
- **Catch** (bắt): Khi ngoại lệ được ném đi thì đây là nơi bắt ngoại lệ để xử lý hoặc thông báo.

GIỚI THIỆU EXCEPTION HANDLING

- **Ví dụ 1:** Cho chương trình tính căn của một số, nếu số đó là số âm thì báo không tính căn được, ngược lại thì tính căn đưa kết quả người dùng.

GIỚI THIỆU EXCEPTION HANDLING

```
#include <iostream>
using namespace std;
int main()
{
    float x;
    cout << "Nhap so x: ";
    cin >> x;
    try
    {
        if (x < 0.0)
            throw 1;
        std::cout << "Ket qua " << x << " la: " << sqrt(x) << endl;
    }
    catch (int)
    {
        cout << "Loi: So am khong tinh can duoc" << endl;
    }
}
```


GIỚI THIỆU EXCEPTION HANDLING

- **Ví dụ 2:** Cho chương trình chia 2 số nguyên, nếu có trường hợp chia cho số 0 thì không thực hiện lệnh này và tiếp tục thực hiện các lệnh sau đó, khi hết chương trình thì dừng.

GIỚI THIỆU EXCEPTION HANDLING

```
#include <iostream>
using namespace std;
int division(int a, int b)
{
    if (b == 0)
        throw 1;
    return a / b;
}
int main()
{
    int x = 10, z;
    for (int i = -3; i < 3; i++)
    {
        try
        {
            z = division(x, i);
            cout << z << endl;
        }
        catch (int e)
        {
            cout << "Division by zero " << e << endl;
        }
    }
}
```

GIỚI THIỆU EXCEPTION HANDLING

- **Ví dụ 3:** Bắt lỗi khi không cấp phát được vùng nhớ của một kiểu dữ liệu cơ bản, ví dụ như **string** trong c++.

```
#include <iostream>
#include <exception>
#include <string>
using namespace std;
int main()
{
    try
    {
        string s;
        s.resize(-1);
    }
    catch (exception &e)
    {
        cout << "Standard exception: " << e.what() << '\n';
    }
    return 0;
}
```

GIỚI THIỆU EXCEPTION HANDLING

➤ Nhiều loại **catch** khác nhau

```
catch (int a)
{
    cout << a << endl;
}
catch (float b)
{
    cout << b << endl;
}
catch (...)
{
    cout << "chuoi";
}
```

GIỚI THIỆU EXCEPTION HANDLING

- **Ví dụ 4: Exception là một lớp** thay vì throw ra các kiểu dữ liệu cơ bản, có thể gói Exception lại thành một lớp, có thể throw ra đối tượng hoặc con trỏ của đối tượng.

```
#include<iostream>
#include <string>
using namespace std;

class MyException
{
public:
    string message()
    {
        return string("Chia cho so 0");
    }
};
```

GIỚI THIỆU EXCEPTION HANDLING

```
class CPhanSo
{
    int Tu, Mau;
public:
    CPhanSo(const int &t, const int &m) : Tu(t), Mau(m) {}
    CPhanSo operator/(CPhanSo &ps)
    {
        if (ps.Tu == 0)
            throw MyException();
        return CPhanSo(Tu*ps.Mau + Mau *ps.Tu, Mau*ps.Mau);
    }
};
```

GIỚI THIỆU EXCEPTION HANDLING

```
int main()
{
    CPhanSo ps1(1, 2), ps2(0, 1);
    try
    {
        CPhanSo ps3 = ps1 / ps2;
    }
    catch(MyException e)
    {
        cout << e.message() << endl;
    }
    return 0;
}
```

GIỚI THIỆU EXCEPTION HANDLING

- **Ví dụ 5: Kế thừa lớp Exception** thay lớp Exception trong C++ có thể được kế thừa để custom lại để bắt nhiều lỗi mang tính “cá nhân” hơn.

```
#include<iostream>
#include <string>
using namespace std;

class MyException: public exception
{
public:
    MyException(): exception("Do dai password wifi it nhat 8 ky tu") {}
};
```


GIỚI THIỆU EXCEPTION HANDLING

```
int main()
{
    string pass;
    cout << "Nhap pass: ";
    cin >> pass;
    try
    {
        if (pass.size() < 8)
            throw MyException();
    }
    catch (MyException me)
    {
        cout<<me.what();
    }
    return 0;
}
```

MỘT SỐ LƯU Ý EXCEPTION HANDLING

Cơ chế thông báo lỗi và xử lý ngoại lệ cho phép tách biệt thành phần xử lý lỗi với các đoạn mã chương trình bình thường.

Trong nhiều trường hợp lỗi chỉ phát sinh khi bắt đầu chạy chương trình, các lỗi được phát hiện nhưng không biết xử lý như thế nào. Cơ chế Exception Handling sẽ giúp chương trình xử lý các tình huống này.

BÀI TẬP EXCEPTION HANDLING

Bài 01: Khai báo một cấu trúc mystack định nghĩa lại lớp stack. Cho một số ngẫu nhiên được tạo ra từ 1 -> 10. Nếu số đó là hợp số thì cho vào stack, nếu số đó là số nguyên tố thì thông báo cho người dùng số vừa tạo ra là số nguyên tố và không thêm vào stack.

Stack chỉ chứa tối đa 10 số, nếu quá 10 số thì thông báo ra hết vùng nhớ và không thêm vào nữa và thoát chương trình.

BÀI TẬP EXCEPTION HANDLING

Bài 02: Viết chương trình mô phỏng đồng hồ kilomet, khi người dùng nhấn vào phím “mũi tên bên phải” chương trình sẽ tăng lên 1 km/h. Khi đồng hồ tăng lên 60 km/h đồng hồ sẽ cảnh báo “QUÁ TỐC ĐỘ” và sẽ hạ xuống từ từ còn 50 km/h lúc đó người dùng có thể bấm tiếp phím mũi tên để tăng thêm tốc độ. (Dùng try catch)

BÀI TẬP EXCEPTION HANDLING

Bài 03: Một máy ATM ban đầu có 10 triệu 100 ngàn VNĐ. Được chia làm các loại tiền như sau:

- 500.000 VNĐ có 10 tờ → tổng: 5.000.000 VNĐ
- 200.000 VNĐ có 20 tờ → tổng: 4.000.000 VNĐ
- 100.000 VNĐ có 10 tờ → tổng: 1.000.000 VNĐ
- 50.000 VNĐ có 2 tờ → tổng: 100.000 VNĐ

Khi người dùng rút, mỗi loại phải có ít nhất 1 tờ tiền trong 4 loại trên. Khi 1 loại trong 4 loại trên hết, thì thông báo xin lỗi người dùng biết không thể đổi tiền lẻ mệnh giá đó.

BÀI TẬP EXCEPTION HANDLING

Bài 04: Kiểm tra tính hợp lệ của một Email và Password.

- Email Không chứa ký tự đặc biệt “+”, “-”, “*”, “/”. Email phải chứa 1 ký tự @ có chứa 1 dấu chấm phía sau @ để đúng đó là 1 tên miền.
- Password: dài ít nhất 8 ký tự, có 1 ký tự viết hoa, có cả chữ lẫn số.

TÀI LIỆU THAM KHẢO

- 1. Lập trình hướng đối tượng, Trần Đan Thư, Đinh Bá Tiến, Nguyễn Tấn Trần Minh Khang, NXB Khoa Học Kỹ Thuật, 2010.
- 2. Lập trình hướng đối tượng, Trần Văn Lãng, NXB Thống Kê, 2004.
- 3. Object-oriented Programming in c++, 4th Edition, Robert Lafore, SAMS, 1997.
- 4. C++ Primer, Fifth Edition, Stephen Prata, SAMS, 2004.
- 5. Slide bài giảng của: Thầy Nguyễn Minh Huy, Thầy Hồ Tuấn Thanh, Thầy Đinh Bá Tiến, Thầy Trần Văn Lãng, Thầy Đặng Bình Phương, Cô Đặng Thị Thanh Nguyên.
- 6. Các website về lập trình:
 - <http://www.cplusplus.com/>
 - <http://stackoverflow.com/>
 - <http://www.codeproject.com/>