

CSC10003 – Phương pháp lập trình hướng đối tượng

Tuần 07: Exception Handling

I. Ví dụ về Exception Handling thông qua kế thừa class exception

VD 1 :

```
#include<iostream>
#include<exception>
#include<string>
using namespace std;
class DividedByZeroException : public exception {
public:
    const char* what() const throw() {
        return "Exception: Cannot divide by zero!";
    }
};
void main() {
    int numerator = 10;
    int denominator = 0;
    int ans;
    try {
        if (denominator == 0)
            throw DividedByZeroException();
        ans = numerator / denominator;
        cout << ans << endl;
    }
    catch (DividedByZeroException& ex) {
        cout << ex.what() << endl;
    }
}
```

VD 2 :

```
#include<iostream>
#include<exception>
#include<string>
#include<sstream>
using namespace std;
class IndexOutOfRangeException : public exception {
private:
    int index, capacity;
public:
    IndexOutOfRangeException(int index, int capacity) {
        this->index = index;
        this->capacity = capacity;
    }
    const char* what() const throw() {
        stringstream stream;
        stream << "Exception: Index out of range" << endl;
        stream << "Detail: index = " << index << ", capacity = "
<< capacity << endl;
        return _strdup(stream.str().c_str());
    }
};
int main() {
    int arr[] = { 2, -7, 8, 9 };
    int size = 4;
    int index = 7;
    try {
        if (index < 0 || index >= size)
            throw IndexOutOfRangeException(index, size);
        cout << arr[index] << endl;
    }
    catch (IndexOutOfRangeException& ex) {
        cout << ex.what() << endl;
    }
    return 0;
}
```

II. Bài tập thực hành

Bài tập 1: Tạo Class Fraction chứa hai thuộc tính tử số và mẫu số. Định nghĩa và cài đặt phương thức SetValue với 2 tham số được truyền vào. Kiểm tra tính hợp lệ của mẫu số, nếu mẫu số bằng 0 thì throw đối tượng DividedByZeroException. Try/catch xử lý exception ở hàm main.

Bài tập 2: Tạo Class Array chứa các thuộc tính cần thiết (mảng kiểu int, size). Định nghĩa và cài đặt phương thức getValueAt với tham số truyền vào là index cần lấy trong mảng. Kiểm tra tính hợp lệ hàm này và throw đối tượng IndexOutOfRangeException nếu không hợp lệ. Try/catch xử lý exception ở hàm main.

Bài tập 3: Định nghĩa và cài đặt class FileNotFoundException để xử lý ngoại lệ cho các thao tác mở/đóng file nếu đường dẫn không hợp lệ.

Bài tập 4: IntegerFormatException. VD: “123” có thể chuyển sang số nguyên 123, nhưng “one-two-three” thì không hợp lệ.

Bài tập 5: DateFormatException. Các giá trị dd/mm/yyyy không hợp lệ.

Bài tập 6: TimeFormatException. Các giá trị giờ/phút/giây không hợp lệ.

Bài tập 7: Sử dụng lại class Stack và Queue ở tuần 3. Xử lý ngoại lệ cho các phương thức push, pop, getTop trong class Stack và enqueue, dequeue, getFront trong class Queue

Yêu cầu khác:

- Các Class cần được tách ra 2 phần:
 - o Phần định nghĩa thuộc tính và phương thức ở file .h
 - o Phần cài đặt nằm ở file .cpp
- Hàm main nằm ở file Main.cpp, chứa các luồng demo các yêu cầu của từng bài tập như mẫu bên dưới

```
//File Main.cpp nằm Source File. Chứa hàm main

void main()
{
    //Bài 1
    ...

    //Bài 2
    ...

    //Bài 3
    ...
}
```

Các trường hợp không tuân thủ sẽ bị trừ từ 5%-20% số điểm tùy theo mức độ.

III. Nộp bài

Trên lớp: Hoàn thành ở lớp: ít nhất 3 bài tập.

Về nhà: Hoàn thành tất cả bài tập.

Tổ chức thư mục theo cấu trúc sau:

- **Source code:** Thư mục chứa mã nguồn. Cần xóa các tập tin trung gian của quá trình biên dịch cho nhẹ bớt (Build > Clean solution và xóa đi thư mục ẩn .vs)

- **Release:** Thư mục chứa tập tin thực thi .exe biên dịch ra từ mã nguồn.
- **Other:** Thư mục chứa các tài liệu nộp thêm ví dụ ảnh sơ đồ lớp.

Nén tất cả ở dạng <MSSV>_Week<X>.zip/rar.

- **MSSV:** mã số sinh viên
- **X:** mã tuần
- **Y:** số bài đã làm
- Ví dụ: **21127001_Week07_07.zip**