KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
227 Nguyễn Văn Cừ, Phường 4, Quận 5, TP.HCM
Điện Thoại: (08) 38.354.266 - Fax:(08) 38.350.096

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

BÀI TẬP TUẦN 9

2022 – 2023

Inheritance & Polymorphism, Exception Handle

Bài 1 Nhập hai số nguyên, viết hàm thực hiện chia hai số nguyên. Sử dụng Exception handle để xử lý trong trường hợp số chia là zero.

Quá trình sử dụng Exception Handle được minh họa trong demo sau:

```
/*********************************************************
 * Exception Handle, File exception.h
 * Thiết kế hàm chia 2 số có xử lý ngoại lệ: 'Chia bởi zero'
 *********************************************************/
#include <iostream>
#include <stdexcept>
using namespace std;

/*
 * Exception Handling
 * One of the avantages of the exception handling approach is
 * that we can design functions that can throw an exception.
 * It is the responsibility of the caller to handle the exception.
*/

/* Problem: Division by zero! */
// (1) Design functions that can throw an exception
// Solution 1: Using the try-catch block to detect an error
// throw by a function
int quotient(int first, int second) {
	if (second == 0) {
		throw 0;
	}
	return (first / second);
}

// Solution 2: Using an object of invalid_argument
// to detect division by zero in a function
// stdexcept: Standard Exception Classes
int stdQuotient(int first, int second) {
	if (second == 0) {
		throw invalid_argument("Error! Devide by zero!");
	}
	return first / second;
}
```

```
/*********************************************************
 * processing Exception in caller program,
 * using try-catch block
*********************************************************/
#include "exception.h"

int main() {
	int num1, num2, result;
	for (int i = 0; i < 3; i++)
	{
		cout << "Enter an integer: ";
		cin >> num1;
		cout << "Enter another integer: ";
		cin >> num2;

		// Chon giai pháp giải quyết
		int chon = 0;
		while (1) {
			/*1.Solution 1, 2.Solution 2*/
			cout << "Nhap: (1 || 2 || 0 thoat)";
			cin >> chon;
			/* 1.Solution 1 */
			if (chon == 1) {
				// try-catch block
				try {
					cout << "Result:"
						<< quotient(num1, num2) << endl;
				}
				catch (...){
					cout << "Phep chia cho zero!" << endl;
				}
			}
			/* 2.Solution 2 */
			else if (chon == 2) {
				// try-catch block
				try
				{
					cout << "Result of division: "
						<< stdQuotient(num1, num2);
					cout << endl;
				}
				catch (invalid_argument ex) {
					cout << ex.what() << endl;
				}
			}
			else
				break;  // Thoat khoi vong lap while
		}
	}

	return 0;
}
-----------------------------------------------------------
```

Bài 2: Exception Handle in Class

Tạo một lớp xử lý ngoại lệ như sau:

Sinh viên tạo một project và thực hành theo demo sau:

```
/*********************************************************
* File giao dien cho lop myexception, myexception.h
*********************************************************/
#ifndef MYEXCEPTION_H
#define MYEXCEPTION_H
#include <iostream>
#include <string>
using namespace std;

class myException {
private:
	string message;
public:
	myException() {
		message = "";
	}
	myException(string msg) {
		message = msg;
	}
	string getMessage() {
		return message;
	}
};
#endif
```

Tạo một lớp nhập hai số nguyên và xử lý ngoại lệ phép chia cho không trong phép chia hai số, lớp được định nghĩa như sau:

```
/*********************************************************
* Divide two integer and exception handle 'Divide by zero'
* File divde_two_integer.h
*********************************************************/
#include <iostream>
using namespace std;
#include "myexception.h";

class Divide {
	int num1;
	int num2;
public:
	Divide();
	~Divide();

	void input();
	double division();
};
```

Bài 3 - Exception Handle in Class

Tạo một lớp AgeInput để ràng buộc về dữ liệu nhập tuổi trong các trường hợp khác nhau sử dụng một lớp xử lý ngoại lệ 'myException'.

Tip: Tạo lớp 'Xử lý ngoại lệ' như Bài 2.

```
/*****************************************************
* File hien thuc AgeInput, ageinput.cpp
*****************************************************/
#ifndef AGEINPUT_CPP
#define AGEINPUT_CPP

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

#include "myexception.h"

class AgeInput : public myException {
private:
	string DEFAULT_MESSAGE = "Your age: ";
	int DEFAULT_LOWER_BOUND = 0;
	int DEFAULT_UPPER_BOUND = 99;
	int lowerBound;
	int upperBound;
public:
	// Constructors
	AgeInput();
	AgeInput(int low, int high);

	int getAge() {
	    return getAge(DEFAULT_MESSAGE); }
	// Method: enter age and checking entrace data bound
	int getAge(string promt);
};
#endif
```

---------- Testing AgeInput class ----------

Your Age: Enter age: 17

Input out of bound!

Your Age: Enter age: 15

Input out of bound!

Your Age: Enter age: 33

Input Okay. Age = 33

Bài 4 Công ty ABC cần xây dựng ứng dụng quản lý thông tin và tính lương cho nhân viên, sử dụng Kế thừa và Đa hình trong lập trình hướng đối tượng của C++. Thông tin mỗi nhân viên bao gồm: mã nhân viên (C-string), họ tên (C-string), ngày sinh, địa chỉ (C-string).

Công ty có 2 loại nhân viên với cách tính lương như sau:

- Lương(Nhân viên sản xuất): số sản phẩm * 50.000 vnđ

- Lương(Nhân viên công nhật): số ngày * 500.000 vnđ

Yêu cầu

I. Vẽ sơ đồ quan hệ kế thừa giữa các lớp, với mỗi lớp có thông tin chi tiết gồm thành phần dữ liệu và hàm thành phần.

Note: *Tương tự như Hình 1.1, Bài 1*.

II. Áp dụng tính kế thừa, khai báo class NVSanXuat và NVCongNhat kế thừa class NhanVien và cài đặt các hàm sau:

1. Cài đặt các Constructors cho mỗi Class
2. Nhập thông tin nhân viên, với ràng buộc dữ liệu nhập như sau:
   - 2.1. Mã nhân viên là chuỗi gồm 5 kí tự số
   - 2.2. Tuổi nhân viên với qui định sau:
     + Nam: 18 – 60 tuổi
     + Nữ: 18 – 55 tuổi
   - 2.3. Số sản phẩm: 10-15 sản phẩm
   - 2.4. Số ngày công: 22 – 26 ngày
3. Xuất thông tin nhân viên ra màn hình
4. Tính lương nhân viên

III. Cài đặt class CongTy sử dụng mảng động hoặc danh sách liên kết. Viết các Constructors và Assignment Operator (toán tử gán bằng), cài đặt các phương thức để thực hiện các chức năng sau.

1. Ghi danh sách nhân viên vào file ‘ds_NhanVien.dat’.
2. Đọc danh sách nhân viên từ file ‘ds_NhanVien.dat’.
3. Xuất danh sách nhân viên ra màn hình.
4. Tính tổng tiền lương của tất cả nhân viên.
5. Tìm nhân viên có lương cao nhất
6. Tính lương trung bình trong công ty
7. Nhập vào mã, tìm nhân viên tương ứng
8. Nhập vào tên, tìm nhân viên tương ứng
9. Có bao nhiêu nhân viên sinh trong tháng 5
10. Thêm một nhân viên vào danh sách, và cập nhật lại file ‘ds_NhanVien’.

11. Xóa một nhân viên khỏi danh sách, và cập nhật lại file ‘ds_NhanVien’.
12. Ghi tất cả các nhân viên có lương nhỏ hơn lương trung bình của công ty lên file ‘emp_LowerAvgSalary.dat’.

==========