

Lab 13

Đa hình

Lập trình hướng đối tượng

Mục tiêu
<ol style="list-style-type: none">1. Hiểu về cơ chế đa hình2. Vận dụng cơ chế đa hình để giải quyết vấn đề

1 Hướng dẫn khởi đầu – Bài toán tính lương nhân viên

Mô tả bài tập

Trong công ty có 3 loại nhân viên như sau:

+ Nhân viên **sản xuất**: trả lương theo số sản phẩm làm ra.

Giả sử mỗi sản phẩm có giá 5000đ, nhân viên làm ra 120 sản phẩm thì số tiền phải trả là:

$$5.000 \times 1200 = 6.000.000 \text{ đ}$$

+ Nhân viên **công nhật**: trả lương theo số ngày làm việc.

Giả sử mỗi ngày công là 250.000 đ, nhân viên đã làm việc 20 ngày, số tiền lương nhận là:

$$250.000 \times 20 = 5.000.000 \text{ đ}$$

+ Nhân viên **quản lý**: trả lương theo lương cơ bản x hệ số

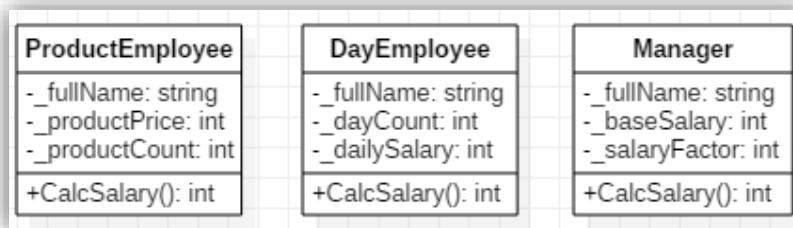
Giả sử lương cơ bản là 1.400.000 đ, hệ số 3 thì lương của nhân viên quản lý là

$$1.400.000 \times 3 = 4.200.000 \text{ đ}$$

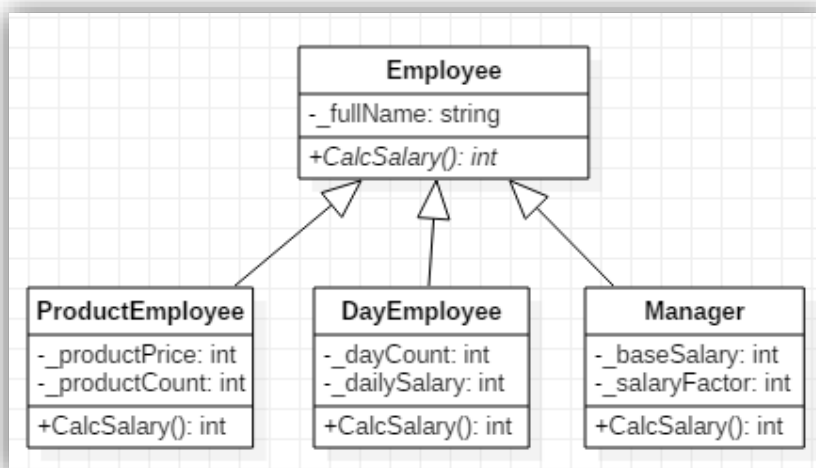
Hãy thiết kế sơ đồ lớp và viết hàm tính lương cho từng nhân viên cụ thể khi đến khi báo cáo lương.

Hướng dẫn cài đặt

Bước 1: Tạo mới dự án

Bước 2: Phân tích sơ bộ ban đầu**Bước 3: Cải tiến sơ đồ lớp**

Ta nhận thấy có thể thiết kế sơ đồ lớp tốt hơn bằng cách vận dụng kế thừa như sau:

**Bước 4: Cài đặt cụ thể các lớp như sau:**

Cài đặt lớp cha **Employee**, là một nhân viên tổng quát

```

class Employee {
private:
    string _fullname;
public:
    string FullName() { return _fullname; }
    void SetFullName(string value) { _fullname = value; }
public:
    Employee(string fullname) {
        _fullname = fullname;
    }
    virtual int CalcSalary() = 0;
};
  
```

Lần lượt cài đặt các lớp con như sau:

```
class ProductEmployee: public Employee {  
private:  
    int _productPrice;  
    int _productCount;  
public:  
    ProductEmployee(string fullname, int price, int count) : Employee(fullname) {  
        _productPrice = price;  
        _productCount = count;  
    }  
  
    int CalcSalary() {  
        return _productPrice * _productCount;  
    }  
};
```

```
class DailyEmployee : public Employee {  
private:  
    int _dayCount;  
    int _dailyPrice;  
public:  
    DailyEmployee(string fullname, int price, int count) : Employee(fullname) {  
        _dailyPrice = price;  
        _dayCount = count;  
    }  
  
    int CalcSalary() {  
        return _dailyPrice * _dayCount;  
    }  
};
```

```

class Manager : public Employee {
private:
    int _baseSalary;
    int _salaryFactor;
public:
    Manager(string fullname, int base, int factor) : Employee(fullname) {
        _baseSalary = base;
        _salaryFactor = factor;
    }

    int CalacSalary() {
        return _baseSalary * _salaryFactor;
    }
};

```

Bước 5: Viết hàm main để kiểm tra cài đặt các lớp

```

int main()
{
    const int count = 6;
    auto a = new Employee*[count];

    a[0] = new ProductEmployee("Long", 2500, 400);
    a[1] = new ProductEmployee("Minh", 3000, 150);
    a[2] = new DailyEmployee("Tuan", 250000, 25);
    a[3] = new DailyEmployee("Kiet", 380000, 18);
    a[4] = new Manager("Chinh", 3300000, 2 );
    a[5] = new Manager("Vu", 3700000, 3);

    cout << "Luong cua cac nhan vien la:" << endl;

    for (auto i = 0; i < count; i++) {
        cout << a[i]->FullName() << ":";
        cout << a[i]->CalcSalary() << endl;
    }

    for (auto i = 0; i < count; i++) {
        delete a[i];
    }
    delete[] a;
}

```

Kết quả như sau:

```
Luong cua cac nhan vien la:  
Long:1000000  
Minh:450000  
Tuan:6250000  
Kiet:6840000  
Chinh:6600000  
Vu:11100000
```

Bước 6: Giải thích ý nghĩa

Cơ chế đa xạ cho phép gọi đúng những phương thức của đối tượng mà không cần biết trước kiểu dữ liệu của đối tượng đó nhờ vào một con trỏ ảo (virtual pointer) bên trong đối tượng. Con trỏ này giữ địa chỉ của một bảng ảo (v) trong đó chứa thông tin về kiểu dữ liệu và địa chỉ các phương thức tương ứng của đối tượng. Đây được xem là điểm then chốt của cơ chế đa xạ.

2 Bài tập vận dụng

Yêu cầu

1. Thực hiện định nghĩa lớp theo thiết kế cho trước vào tập tin .h.
2. Thực hiện cài đặt lớp trong tập tin .cpp cho lớp tương ứng.
3. Viết các đoạn mã nguồn kiểm tra việc định nghĩa lớp trong hàm main.

Danh sách các lớp cần cải tiến cụ thể

1. Cải tiến bài tập hướng dẫn, thêm vào hàm tính tiền thưởng

CalcBonus(): `int`

Với yêu cầu như sau:

Nếu nhân viên sản xuất làm hơn 100 sản phẩm: thưởng thêm 500đ cho mỗi sản phẩm thêm

Nếu nhân viên công nhật làm hơn 20 ngày, thưởng thêm 100.000 đ cho mỗi ngày làm thêm

Nhân viên quản lí không thưởng gì.

2. Cải tiến bài tập hướng dẫn, thêm vào hàm phạt

CalcPenalty(): `int`

Với yêu cầu như sau

Nếu nhân viên sản xuất làm ít hơn 100 sản phẩm: phạt 600đ cho mỗi sản phẩm thiếu chỉ tiêu.

Nếu nhân viên công nhật làm ít hơn 15 ngày, phạt 150.000 đ cho mỗi ngày thiếu chỉ tiêu.

4 Hướng dẫn nộp bài

- + Mỗi bài tương ứng với 1 folder. Chỉ chứa file .h và .cpp và Makefile
- + Tổng hợp tất cả folder vào folder MSSV.
- + Nén lại tất cả thành một tập tin duy nhất MSSV.zip.

Để nộp bài, nén tất cả lại và đặt tên với định dạng MSSV.zip

-- HẾT --