

BÀI 6: QUẢN LÝ DỰ ÁN

Khai báo lớp Nhân viên (mã NV, Họ tên, Địa chỉ, Số ĐT, chuyên môn), mã NV là một số nguyên có 5 chữ số, tự động tăng. Khai báo lớp Dự án gồm các thuộc tính (mã dự án, Tên dự án, Kiểu dự án, Tổng kinh phí) – trong đó kiểu dự án có thể gồm: nhỏ, trung bình hoặc lớn. Mã dự án là một số nguyên có 5 chữ số, tự động tăng. Khai báo lớp Bảng phân công có quan hệ kết hợp (association) với lớp Nhân viên và lớp Dự án trong đó một nhân viên có thể được gán cho một hoặc nhiều dự án một lúc với số ngày tham gia khác nhau và vị trí công việc khác nhau.

Viết chương trình trong ngôn ngữ C++ thực hiện các yêu cầu sau:

1. Nhập danh sách dự án. In ra danh sách dự án đã có
2. Nhập danh sách nhân viên. In ra danh sách nhân viên đã có
3. Nhập danh sách phân công dự án cho mỗi nhân viên đã có và in danh sách ra màn hình (chú ý: cùng một nhân viên thì không thể tham gia cùng một dự án với hai vị trí khác nhau nhưng có thể tham gia cùng lúc vào nhiều dự án)
4. Sắp xếp danh sách phân công
 - a. Theo Họ tên nhân viên
 - b. Theo Số ngày tham gia (giảm dần)
5. Tìm kiếm và hiển thị danh sách phân công theo tên nhân viên

CODE

```
#include <bits/stdc++.h>
using namespace std;

class NhanVien {
private:
    int mnv;
    string hoten, diachi, sdt, chuyenmon;
public:
```

```

    static int dem;
    NhanVien();
    void nhapNV();
    void inNV();
    string getHoten() const { return hoten; };
    int getMNV() const { return mnv; };
};

int NhanVien::dem = 0;

class DuAn {
private:
    int mda;
    string tenda, kieuda;
    long long tongkp;

public:
    static int soda;
    DuAn();
    void nhapDA();
    void inDA();
};

class BangPhanCong {
private:
    NhanVien nhanVien;
    DuAn duAn;
    pair<NhanVien, DuAn> data;
    int soNgayThamGia;
    string viTriCongViec;
public:
    BangPhanCong();
    BangPhanCong(NhanVien nv, DuAn da, int days, string position) :
data(make_pair(nhanVien, duAn)), soNgayThamGia(days), viTriCongViec(position)
{}

    NhanVien getNhanVien() const { return data.first; }
    DuAn getDuAn() const { return data.second; }
    int getSoNgayThamGia() const { return soNgayThamGia; }
    string getViTriCongViec() const { return viTriCongViec; }
    string getHoten() const { return data.first.getHoten(); }
    int getMNV() const { return data.first.getMNV(); };
    void nhapPhanCong();
    void inPhanCong();
    bool comparePhanCong(const BangPhanCong& a, const BangPhanCong&
b);

    bool operator<(const BangPhanCong& other) const {
return nhanVien.getHoten() < other.getNhanVien().getHoten();
}
    bool hasEmployeeName(const string& searchName) const {
return nhanVien.getHoten() == searchName;
}
};

```

```

int DuAn::soda = 0;

NhanVien::NhanVien() {
    mnv = 10000 + dem++;
    hoten = diachi = chuyenmon = sdt = "";
}

void NhanVien::nhapNV() {
    cout << "Ten nhan vien: ";
    getline(cin, hoten);
    cout << "Dia chi: ";
    getline(cin, diachi);
    cout << "So dien thoai: ";
    getline(cin, sdt);
    cout << "Chuyen mon: ";
    getline(cin, chuyenmon);
}

void NhanVien::inNV() {
    cout << "Ten nhan vien: " << hoten << endl;
    cout << "Ma nhan vien: " << getMNV() << endl;
    cout << "Dia chi: " << diachi << endl;
    cout << "So dien thoai: " << sdt << endl;
    cout << "Chuyen mon: " << chuyenmon << endl;
}

DuAn::DuAn() {
    mda = 10000 + soda++;
    tenda = kieuda = "";
    tongkp = 0;
}

void DuAn::nhapDA() {
    cout << "Ten du an: ";
    getline(cin, tenda);
    cout << "Kieu du an: ";
    getline(cin, kieuda);
    cout << "Tong kinh phi: ";
    cin >> tongkp;
    cin.ignore();
}

void DuAn::inDA() {
    cout << "Ten du an: " << tenda << endl;
    cout << "Ma du an: " << mda << endl;
    cout << "Kieu du an: " << kieuda << endl;
    cout << "Tong kinh phi: " << tongkp << endl;
}

BangPhanCong::BangPhanCong() {
    soNgayThamGia = 0;
    viTriCongViec = "";
}

```

```

void BangPhanCong::nhapPhanCong() {
    nhanVien.nhapNV();
    duAn.nhapDA();

    cout << "So ngay tham gia: ";
    cin >> soNgayThamGia;
    cin.ignore();

    cout << "Vi tri cong viec: ";
    getline(cin, viTriCongViec);
}

void BangPhanCong::inPhanCong() {
    nhanVien.inNV();
    duAn.inDA();
    cout << "So ngay tham gia: " << soNgayThamGia << endl;
    cout << "Vi tri cong viec: " << viTriCongViec << endl;
}

bool BangPhanCong::comparePhanCong(const BangPhanCong& a, const BangPhanCong&
b) {
    return a.getSoNgayThamGia() > b.getSoNgayThamGia();
}

int main() {
    vector<NhanVien> dsnv;
    vector<DuAn> dsda;
    vector<BangPhanCong> bpc;
    int n, m;

    //Nhap danh sach nhan vien
    cout << "So luong nhan vien: ";
    cin >> n;
    cin.ignore();
    for (int i = 0; i < n; i++) {
        NhanVien nv;
        nv.nhapNV();
        dsnv.push_back(nv);
    }
    if(!dsnv.empty()){
        cout<<"=====Danh sach nhan vien===== "<<endl;
        for (auto &nv : dsnv) {
            nv.inNV();
            cout << endl;
        }
    }

    //Nhap danh sach du an
    cout << "So du an: ";
    cin >> m;
    cin.ignore();

```

```

for (int i = 0; i < m; i++) {
    DuAn da;
    da.nhapDA();
    dsda.push_back(da);
}
if(!dsda.empty()){
    cout<<"=====Danh sach du an===== "<<endl;
    for (auto &da : dsda) {
        da.inDA();
        cout << endl;
    }
}

//Nhap danh sach phan cong du an
cout<<"Nhap danh sach phan cong du an"<<endl;
for(int i=0;i<m*n;i++){
    BangPhanCong phanCong;
    phanCong.nhapPhanCong();
    bpc.push_back(phanCong);
}
if(!bpc.empty()){
    /*sort(bpc.begin(), bpc.end(), [](const BangPhanCong& a, const
BangPhanCong& b) { //Sap xep theo so ngay tham gia giam dan
        return a.getSoNgayThamGia() > b.getSoNgayThamGia();
    });
    */
    sort(bpc.begin(), bpc.end()); //Sap xep theo ten

    cout<<"\n=====Bang phan cong du an===== "<<endl;
    for ( auto& phanCong : bpc) {
        phanCong.inPhanCong();
        cout << endl;
    }
}

//Tim kiem nhan vien
string searchName;
cout << "Nhap ten nhan vien can tim: ";
getline(cin, searchName);

auto it = find_if(bpc.begin(), bpc.end(), [searchName](const
BangPhanCong& phanCong) {
    return phanCong.hasEmployeeName(searchName);
});
if (it != bpc.end()) {
    cout << "\n=====Bang phan cong du an cho nhan vien " <<
searchName << "===== \n";
    it->inPhanCong();
    cout << endl;
} else {
    cout << "Khong tim thay nhan vien " << searchName << endl;
}
}

```

```
    return 0;
}
```

```
So luong nhan vien: 2
Ten nhan vien: dan
Dia chi: 1
So dien thoai: 1
Chuyen mon: 1
Ten nhan vien: binh
Dia chi: 2
So dien thoai: 2
Chuyen mon: 2
=====Danh sach nhan vien=====
Ten nhan vien: dan
Ma nhan vien: 10000
Dia chi: 1
So dien thoai: 1
Chuyen mon: 1

Ten nhan vien: binh
Ma nhan vien: 10001
Dia chi: 2
So dien thoai: 2
Chuyen mon: 2

So du an: 1
Ten du an: tes
Kieu du an: to
Tong kinh phi: 3
=====Danh sach du an=====
Ten du an: tes
Ma du an: 10000
Kieu du an: to
Tong kinh phi: 3

Nhap danh sach phan cong du an
Ten nhan vien: dan
Dia chi: 1
```

```
Ten nhan vien: dan
Dia chi: 1
So dien thoai: 1
Chuyen mon: 1
Ten du an: tes
Kieu du an: to
Tong kinh phi: 3
So ngay tham gia: 1
Vi tri cong viec: l
Ten nhan vien: binh
Dia chi: 2
So dien thoai: 2
Chuyen mon: 2
Ten du an: tes
Kieu du an: to
Tong kinh phi: 3
So ngay tham gia: 2
Vi tri cong viec: j

=====Bang phan cong du an=====
Ten nhan vien: dan
Ma nhan vien: 10002
Dia chi: 1
So dien thoai: 1
Chuyen mon: 1
Ten du an: tes
Ma du an: 10001
Kieu du an: to
Tong kinh phi: 3
So ngay tham gia: 1
Vi tri cong viec: l

Ten nhan vien: binh
Ma nhan vien: 10004
Dia chi: 2
```

```
So dien thoai: 2
Chuyen mon: 2
Ten du an: tes
Ma du an: 10003
Kieu du an: to
Tong kinh phi: 3
So ngay tham gia: 2
Vi tri cong viec: j

Nhap ten nhan vien can tim: dan

=====Bang phan cong du an cho nhan vien dan=====
Ten nhan vien: dan
Ma nhan vien: 10002
Dia chi: 1
So dien thoai: 1
Chuyen mon: 1
Ten du an: tes
Ma du an: 10001
Kieu du an: to
Tong kinh phi: 3
So ngay tham gia: 1
Vi tri cong viec: l

-----
Process exited after 96.73 seconds with return value 0
Press any key to continue . . . |
```