

**BÁO CÁO BÀI TẬP 1**

**Môn học: Lập trình hướng đối tượng**

**Tên chủ đề: LAB01-BT2**

*GVHD: Nguyễn Hữu Quyền*

*Ngày báo cáo: 18/03/2023*

**Nhóm: 01**

**1. THÔNG TIN CHUNG:**

Lớp: IT002.N28.1

STT	Họ và tên	MSSV	Email
1	Ngô Anh Quang	22521201	22521201@gm.uit.edu.vn
2	Mai Nguyễn Nam Phương	22521164	22521164@gm.uit.edu.vn

**2. NỘI DUNG THỰC HIỆN:**

STT	Công việc	Kết quả tự đánh giá	Người đóng góp
1	Kịch bản 01	100%	1+2
2	Kịch bản 02	100%	1+2
3	Kịch bản 03	100%	1+2

## BÁO CÁO CHI TIẾT

### 1. Kịch bản 01

- Tài nguyên: Thiết lập lớp PhanSo để biểu diễn khái niệm phân số với hai thành phần dữ liệu tử số, mẫu số và các hàm thành phần cộng, trừ, nhân, chia hai phân số, các hàm thành phần xuất, nhập, định giá trị cho phân số
- Mục tiêu: Viết chương trình cho phép nhập vào hai phân số, in ra kết quả các phép toán cộng, trừ, nhân, chia hai phân số kể trên
- Các bước thực hiện:

i) tạo hàm tìm UCLN của 2 số a, b có dạng :

```
int gcd(int a, int b) {  
    if (a == 0) return b;  
    else return gcd( b%a, a );  
}
```

- ii) gán giá trị i=UCLN của 2 tử và mẫu, lấy tử và mẫu chia cho i để rút gọn phân số
- iii) Tạo class PhanSo gồm 2 thành phần tử mẫu là private, hàm nhập xuất cộng trừ nhân chia là public
- iv) Trong public khởi tạo constructor cho PhanSo, đặc mặc định tử là 0, mẫu là 1

```
PhanSo()  
{  
    tu = 0;  
    mau = 1;  
}
```

- v) Định nghĩa cộng trừ nhân chia đều là 1 phân số mới. Làm như bình thường

```
PhanSo Cong(PhanSo b)
{
    PhanSo c;
    c.tu = tu * b.mau + mau * b.tu;
    c.mau = mau * b.mau;
    return c;
}

PhanSo Tru(PhanSo b)
{
    PhanSo c;
    c.tu = tu * b.mau - mau * b.tu;
    c.mau = mau * b.mau;
    return c;
}

PhanSo Nhan(PhanSo b)
{
    PhanSo c;
    c.tu = tu * b.tu;
    c.mau = mau * b.mau;
    return c;
}

PhanSo Chia(PhanSo b)
{
    PhanSo c;
    c.tu = tu * b.mau;
    c.mau = mau * b.tu;
    return c;
}

vi) Xuất phân số dưới dạng rút gọn.
void Xuat()
{
    int i = gcd(tu, mau);
    tu /= i;
    mau /= i;
    int z;
    z = (int)tu / mau;
    if (tu % mau == 0) cout << z << endl;
    else cout << tu << '/' << mau << endl;
}
```

## 2. Kịch bản 02

- Tài nguyên: Xây dựng lớp biểu diễn khái niệm số phức với hai thành phần dữ liệu thực, ảo và các hàm thành phần xuất, nhập, định giá trị cho số phức, cộng, trừ, nhân, chia hai số phức
- Mục tiêu: Viết chương trình cho phép nhập vào hai số phức, in ra kết quả các phép toán cộng, trừ, nhân, chia hai số phức kể trên

- Các bước thực hiện:

- Tạo class tên là con gồm thành phần thuc, ao là private, các hàm cộng trừ nhân chia là public
- Trong đó, định nghĩa hàm cộng là 1 con tên “cong”, rồi truyền vào 1 con khác tên b. Khi đó, ta gọi 1 con trong hàm main, truy cập vào đối tượng “cong” và truyền vào số phức ta cần thực hiện phép cộng. Khi xuất ra sẽ là 1 số phức. Gán c = số phức đó r xuất

```

con cong(con b)
{
    con c;
    c.thuc = thuc + b.thuc;
    c.ao = ao + b.ao;
    return c;
}

c = a.cong(b);
cout << "Tong 2 so phuc= ";

```

- Tương tự với các toán tử khác

```

con tru(con b)
{
    con c;
    c.thuc = thuc - b.thuc;
    c.ao = ao - b.ao;
    return c;
}

con nhan(con b)
{
    con c;
    c.thuc = thuc * b.thuc - ao * b.ao;
    c.ao = thuc * b.ao + ao * b.thuc;
    return c;
}

con chia(con b)
{
    con c;
    c.thuc = (thuc * b.thuc + ao * b.ao) / (pow(b.thuc, 2) + pow(b.ao, 2));
    c.ao = (ao * b.thuc - thuc * b.ao) / (pow(b.thuc, 2) + pow(b.ao, 2));
    return c;
}

```

### 3. Kịch bản 03

- Tài nguyên: Xây dựng lớp Candidate (Thí sinh) gồm các thuộc tính: mã, tên, ngày tháng năm sinh, điểm thi Toán, Văn, Anh và các phương thức cần thiết
- Mục tiêu: Xây dựng lớp TestCandidate để kiểm tra lớp trên:
  - Nhập vào n thí sinh (n do người dùng nhập)
  - In ra thông tin về các thí sinh có tổng điểm lớn hơn 15
- Các bước thực hiện:

- i) Tạo class Candidate bao gồm các private theo đề bài đã cho, các thuộc tính bao gồm hàm khởi tạo, hàm xuất và hàm tính tổng điểm 3 môn toán văn anh

```
public:
    Candidate(string ma, string name, string ntns, float toan, float van, float anh)
    {
        this->ma = ma;
        this->name = name;
        this->ntns = ntns;
        this->toan = toan;
        this->van = van;
        this->anh = anh;
    }
    int sum()
    {
        return toan + van + anh;
    }
    void Xuat()
    {
        cout << "Ma: " << ma << endl;
        cout << "Ten: " << name << endl;
        cout << "Ngày tháng năm sinh: " << ntns << endl;
        cout << "Điểm toán: " << toan << endl;
        cout << "Điểm văn: " << van << endl;
        cout << "Điểm anh: " << anh << endl;
    }
};
```

- ii) Tạo thêm 1 lớp class TestCandidate chỉ có public

- iii) Tạo một hàm vector tĩnh tên là Nhap\_ds\_ts để :

```
static vector<Candidate> Nhap_ds_ts()
{
    int n;
    cout << "Nhập vào số thí sinh: " ;
    cin >> n;
```

- (1) Nhập vào số thí sinh

- (2) Khởi tạo 1 vector theo dạng class Candidate có tên là danh\_sach để nhập và lưu thông tin số thí sinh

```

vector<Candidate> danh_sach;
for (int i = 0; i < n; i++)
{
    cout << "Nhap vao thong tin thi sinh thu " << i+1 << ": " << endl;
    string ma;
    string name;
    string ntns;
    float toan;
    float van;
    float anh;
    cout << "Ma: ";
    cin.ignore();
    cin >> ma;
    cout << "Ten: ";
    cin.ignore();
    getline(cin, name);
    cout << "Ngay sinh: "; cin.ignore();
    getline(cin, ntns);
    cout << "Diem toan: ";
    cin >> toan;
    cout << "Diem van: ";
    cin >> van;
    cout << "Diem anh: ";
    cin >> anh;

```

```

Candidate thi_sinh(ma, name, ntns, toan, van, anh);
danh_sach.push_back(thi_sinh);

```

```

}
return danh_sach;

```

Ở đây chúng ta đẩy thông tin của thí sinh thứ  $i$  vào hàm khởi tạo có tên `thi_sinh` đã được khởi tạo ở lớp `Candidate`.

Quay lại thí sinh thứ  $i + 1$  cho đến khi  $i$  không còn thỏa điều kiện của vòng lặp `for`

(3)

```
static void display(vector<Candidate> danh_sach)
{
    cout <<endl <<"Danh sach thi sinh co tong diem hon 15: " <<endl;
    for (Candidate thi_sinh : danh_sach) {
        if (thi_sinh.sum() > 15) {
            thi_sinh.Xuat();
        }
    }
}
```

- Ta tạo một hàm tĩnh tên display để in thông tin của các thí sinh thỏa yêu cầu của đề bài là có tổng điểm 3 môn hơn 15
- Với vòng lặp for ta duyệt qua các phần tử của lớp danh\_sach đã được ta nhập vào trước để đưa qua đối tượng thi\_sinh dưới dạng Candidate rồi kiểm tra lại với hàm sum thỏa điều kiện của đề sau đó được ta xuất ra.

---

HẾT