**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**



**MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**BÀI TẬP THỰC HÀNH 1**

GVHD: Nguyễn Ngọc Quí

Sinh viên thực hiện: Bùi Ngọc Thái

🙡🙢 Tp. Hồ Chí Minh, 02/2024 🙠🙣

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*……., ngày……...tháng……năm 2024*

**Người nhận xét**

*(Ký tên và ghi rõ họ tên****)***

**MỤC LỤC**

[Bài tập 1: Xây dựng lớp ngày tháng năm. 8](#_Toc162388258)

[**1. Thực hiện xây dựng lớp. 8**](#_Toc162388259)

[**2. Class diagram của lớp NgayThangNam. 8**](#_Toc162388260)

[**3. Thuộc tính của lớp NgayThangNam. 9**](#_Toc162388261)

[**4. Phương thức của class NgayThangNam. 9**](#_Toc162388262)

[**5. Xác định Input, Output. 10**](#_Toc162388263)

[**6. Hàm main của chương trình. 11**](#_Toc162388264)

[**7. Kết quả chạy của chương trình. 11**](#_Toc162388265)

[Bài tập 2: Xây dựng lớp phân số. 12](#_Toc162388266)

[**1. Thực hiện xây dựng lớp. 12**](#_Toc162388267)

[**2. Class diagram của lớp PhanSo. 12**](#_Toc162388268)

[**3. Thuộc tính của lớp PhanSo. 13**](#_Toc162388269)

[**4. Phương thức của lớp PhanSo. 13**](#_Toc162388270)

[**5. Xác định Input, Output. 17**](#_Toc162388271)

[**6. Hàm main của chương trình bài tập 2. 18**](#_Toc162388272)

[**7. Kết quả chạy của chương trình. 19**](#_Toc162388273)

[Bài tập 3: Xây dựng lớp số phức bao gồm. 20](#_Toc162388274)

[**1. Thực hiện xây dựng lớp. 20**](#_Toc162388275)

[**2. Class diagram của lớp SoPhuc. 20**](#_Toc162388276)

[**3. Thuộc tính của lớp SoPhuc. 21**](#_Toc162388277)

[**4. Phương thức của lớp SoPhuc: 21**](#_Toc162388278)

[**5. Dưới đây là bảng công thức của tổng, hiệu, tích, giữa hai số phức. 21**](#_Toc162388279)

[**6. Xác định Input, Output. 23**](#_Toc162388280)

[**7. Hàm main của chương trình bài tập 3. 24**](#_Toc162388281)

[**8. Kết quả chạy của chương trình. 24**](#_Toc162388282)

[Bài tập 4: Xây dựng lớp giờ phút giây. 24](#_Toc162388283)

[**1. Thực hiện xây dựng lớp. 25**](#_Toc162388284)

[**2. Class diagram của lớp GioPhutGiay. 25**](#_Toc162388285)

[**3. Thuộc tính của lớp GioPhutGiay. 25**](#_Toc162388286)

[**4. Phương thức của lớp GioPhutGiay. 26**](#_Toc162388287)

[**5. Xác định Input, Output: 27**](#_Toc162388288)

[**6. Hàm main của chương trình. 27**](#_Toc162388289)

[**7. Kết quả chạy của chương trình. 28**](#_Toc162388290)

[Mã nguồn: 28](#_Toc162388291)

**DANH MỤC BẢNG**

[**Bảng 1: Bảng công thức tính tổng, hiệu, tích, thương giữa hai số phức 22**](#_Toc161944502)

**DANH MỤC HÌNH ẢNH**

[**Hình 1.1: Thực hiện xây dựng lớp NgayThangNam 8**](#_Toc161945679)

[**Hình 1.2: Class diagram của lớp NgayThangNam 8**](#_Toc161945680)

[**Hình 1.3: Phương thức Nhap() của lớp NgayThangNam 9**](#_Toc161945681)

[**Hình 1.4: Phương thức Xuat() của lớp NgayThangNam 9**](#_Toc161945682)

[**Hình 1.5: Phương thức NgayThangNamTiepTheo() của lớp NgayThangNam (1) 9**](#_Toc161945683)

[**Hình 1.6: Phương thức NgayThangNamTiepTheo() của lớp NgayThangNam (2) 10**](#_Toc161945684)

[**Hình 1.7: Hàm main của chương trình bài tập 1 11**](#_Toc161945685)

[**Hình 1.8: Kết quả của chương trình với trường hợp cuối tháng 12 của năm 11**](#_Toc161945686)

[**Hình 1.9: Kết quả của chương trình với trường hợp cuối tháng 2 của năm nhuận 11**](#_Toc161945687)

[**Hình 1.10: Kết quả của chương trình ứng với trường hợp các ngày bình thường 11**](#_Toc161945688)

[**Hình 2.1: Thực hiện xây dựng lớp PhanSo 12**](#_Toc161945689)

[**Hình 2.2: Class diagram của lớp PhanSo 12**](#_Toc161945690)

[**Hình 2.3: Phương thức Nhap() của lớp PhanSo 13**](#_Toc161945691)

[**Hình 2.4: Phương thức Nhap() của lớp PhanSo 14**](#_Toc161945692)

[**Hình 2.5: Phương thức RutGon() của lớp PhanSo kèm hàm UCLN 14**](#_Toc161945693)

[**Hình 2.6: Phương thức Tong(PhanSo ps) của lớp PhanSo 15**](#_Toc161945694)

[**Hình 2.7: Phương thức Hieu(PhanSo ps) của lớp PhanSo 15**](#_Toc161945695)

[**Hình 2.8: Phương thức Tich(PhanSo ps) của lớp PhanSo 16**](#_Toc161945696)

[**Hình 2.9: Phương thức Thuong(PhanSo ps) của lớp PhanSo 16**](#_Toc161945697)

[**Hình 2.10: Phương thức SoSanh(PhanSo ps) của lớp PhanSo 17**](#_Toc161945698)

[**Hình 2.11: Hàm main của chương trình bài tập 2 18**](#_Toc161945699)

[**Hình 2.12: Kết quả của chương trình ứng với phân số thứ nhất nhỏ hơn phân số thứ hai 19**](#_Toc161945700)

[**Hình 2.13: Kết quả của chương trình ứng với phân số thứ nhất lớn hơn phân số thứ hai 19**](#_Toc161945701)

[**Hình 2.14: Kết quả của chương trình ứng với hai phân số bằng nhau 19**](#_Toc161945702)

[**Hình 3.1: Thực hiện xây dựng lớp SoPhuc 20**](#_Toc161944809)

[**Hình 3.2: Class diagram của lớp SoPhuc 20**](#_Toc161944810)

[**Hình 3.3: Phương thức Nhap() của lớp SoPhuc 21**](#_Toc161944811)

[**Hình 3.4: Phương thức Xuat() của lớp SoPhuc 21**](#_Toc161944812)

[**Hình 3.5: Phương thức Tong(SoPhuc b) của lớp SoPhuc 22**](#_Toc161944813)

[**Hình 3.6: Phương thức Hieu(SoPhuc b) của lớp SoPhuc 22**](#_Toc161944814)

[**Hình 3.7: Phương thức Tich(SoPhuc b) của lớp SoPhuc 23**](#_Toc161944815)

[**Hình 3.8: Phương thức Thuong(SoPhuc b) của lớp SoPhuc 23**](#_Toc161944816)

[**Hình 3.9: Hàm main của chương trình bài tập 3 24**](#_Toc161944817)

[**Hình 3.10: Kết quả chạy của chương trình bài tập 3 24**](#_Toc161944818)

[**Hình 4.1: Thực hiện xây dựng lớp GioPhutGiay 25**](#_Toc161944819)

[**Hình 4.2: Class diagram của lớp GioPhutGiay 25**](#_Toc161944820)

[**Hình 4.3: Phương thức Nhap() của lớp GioPhutGiay 26**](#_Toc161944821)

[**Hình 4.4: Phương thức Xuat() của class GioPhutGiay 26**](#_Toc161944822)

[**Hình 4.5: Phương thức TinhCongThemMotGiay() của lớp GioPhutGiay 27**](#_Toc161944823)

[**Hình 4.6: Hàm main của chương trình bài tập 4 27**](#_Toc161944824)

[**Hình 4.7: Kết quả của chương trình với trường hợp bình thường 28**](#_Toc161944825)

[**Hình 4.8: Kết quả của chương trình ứng với trường hợp 23:59:59 28**](#_Toc161944826)

**NỘI DUNG BÀI LÀM**

#### Bài tập 1: Xây dựng lớp ngày tháng năm.

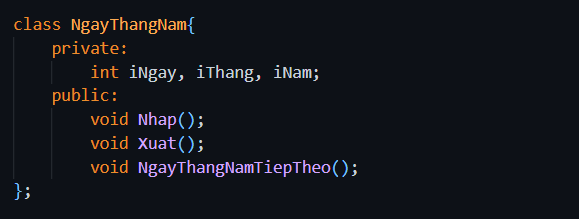
* Thuộc tính: iNgay, iThang, iNam
* Phương thức: Nhap(), Xuat(), NgayThangNamTiepTheo()

**Yêu cầu**: Thực hiện xây dựng lớp, vẽ class diagram và khai báo các thuộc tính,

phương thức. Viết nội dung vào các phương thức đã khai báo. Gọi các phương thức

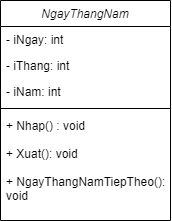
trong hàm main().

##### Thực hiện xây dựng lớp.



Hình 1.1: Thực hiện xây dựng lớp NgayThangNam

##### Class diagram của lớp NgayThangNam.



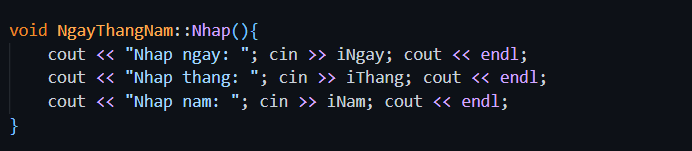
Hình 1.2: Class diagram của lớp NgayThangNam

##### Thuộc tính của lớp NgayThangNam.

* iNgay: Lưu trữ thông tin về ngày.
* iThang: Lưu trữ thông tin về tháng.
* iNam: Lưu trữ thông tin về năm.

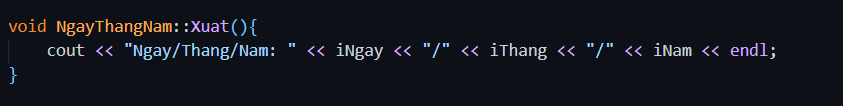
##### Phương thức của class NgayThangNam.

* Phương thức Nhap(): Nhập thông tin về ngày, tháng và năm hợp lệ từ người dùng.



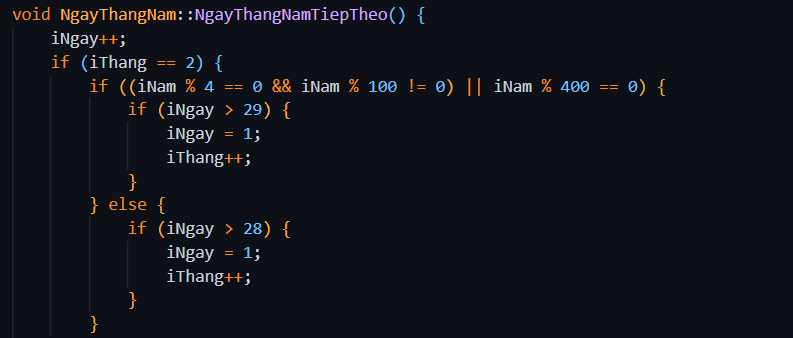
Hình 1.3: Phương thức Nhap() của lớp NgayThangNam

* Phương thức Xuat(): Hiển thị thông tin về ngày, tháng và năm.

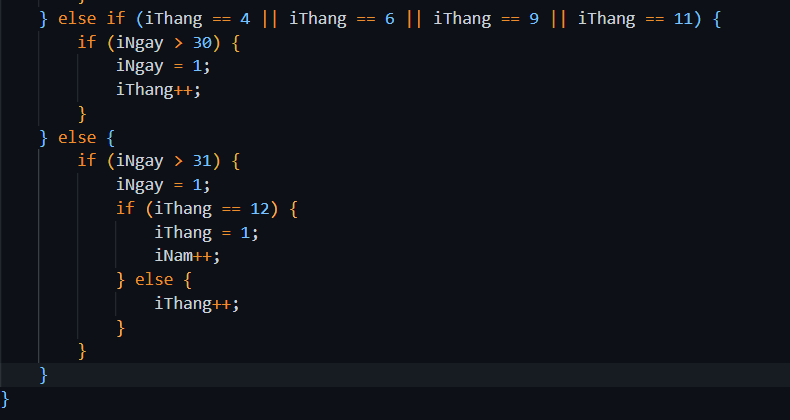


Hình 1.4: Phương thức Xuat() của lớp NgayThangNam

* Phương thức NgayThangNamTiepTheo(): Tính toán và hiển thị ngày, tháng và năm tiếp theo.



Hình 1.5: Phương thức NgayThangNamTiepTheo() của lớp NgayThangNam (1)



Hình 1.6: Phương thức NgayThangNamTiepTheo() của lớp NgayThangNam (2)

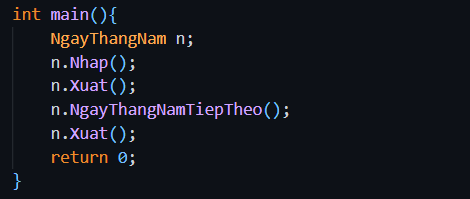
Hướng giải quyết: trong phương thức NgayThangNamTiepTheo(), tăng ngày lên 1 đơn vị, khi đó ta xét các trường hợp như sau:

* Nếu là tháng hai, hay iThang == 2, ta cần kiểm tra xem là năm thuần hay năm nhuận. Để kiểm tra xem năm nhuần hay năm nhuận thì ta có năm nhuận là năm chia hết cho 400 hoặc chia hết cho 4 nhưng không chia hết cho 100. Có 2 trường hợp:
* Năm thuần: Nếu iNgay > 28, đặt lại iNgay = 1, iThang tăng thêm 1 đơn vị.
* Năm thuận: Nếu iNgay > 29, đặt lại iNgay = 1, iThang tăng thêm 1 đơn vị.
* Trong các tháng 4, 6, 9, 11 là các tháng có 30 ngày hay iThang == 4 || iThang == 6 || iThang == 9 || iThang == 11. Nếu iNgay > 30, đặt iNgay = 1, iThang tăng thêm 1 đơn vị.
* Các tháng còn lại có 31 ngày. Nếu iNgay > 31, đặt iNgay = 1, nếu iThang == 12 hay là tháng cuối năm, đặt iThang = 1, iNam tăng thêm 1 đơn vị, ngược lại nếu iThang khác 12 thì iThang tăng thêm 1 đơn vị.

##### Xác định Input, Output.

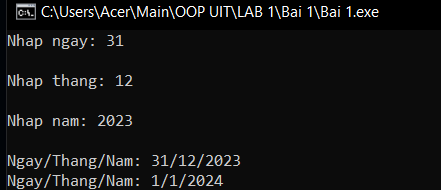
* Input (đầu vào): Ngày tháng năm được nhập từ bàn phím/ người dùng.
* Output (đầu ra): Ngày tháng năm đã được nhập và ngày tháng năm sau khi được cộng thêm một ngày.

##### Hàm main của chương trình.

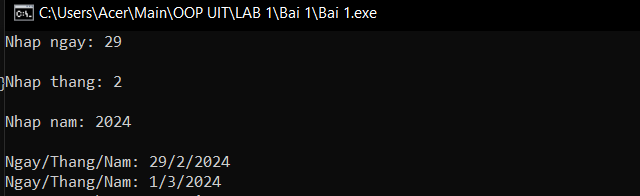


Hình 1.7: Hàm main của chương trình bài tập 1

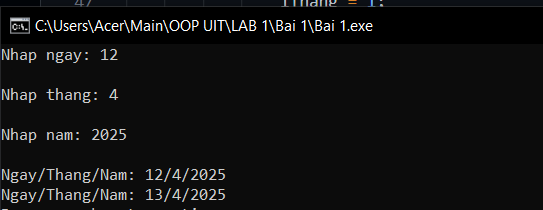
##### Kết quả chạy của chương trình.



Hình 1.8: Kết quả của chương trình với trường hợp cuối tháng 12 của năm



Hình 1.9: Kết quả của chương trình với trường hợp cuối tháng 2 của năm nhuận



Hình 1.10: Kết quả của chương trình ứng với trường hợp các ngày bình thường

#### Bài tập 2: Xây dựng lớp phân số.

* Thuộc tính: iTu, iMau
* Phương thức: Nhap(), Xuat(), RutGon(), Tong(), Hieu(), Tich(), Thuong(),

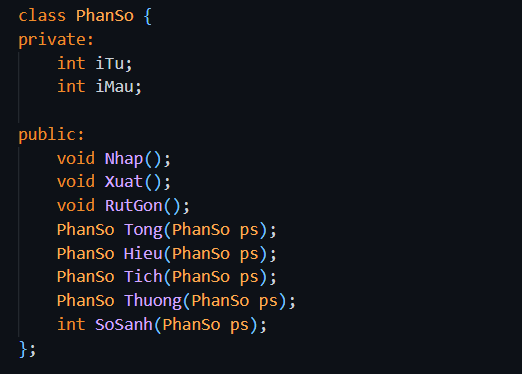
SoSanh()

**Yêu cầu:** Thực hiện xây dựng lớp, vẽ class diagram và khai báo các thuộc tính,

phương thức. Viết nội dung vào các phương thức đã khai báo. Gọi các phương thức

trong hàm main().

###### Thực hiện xây dựng lớp.



Hình 2.1: Thực hiện xây dựng lớp PhanSo

###### Class diagram của lớp PhanSo.



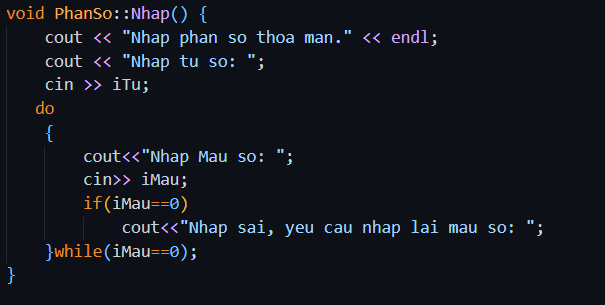
Hình 2.2: Class diagram của lớp PhanSo

###### Thuộc tính của lớp PhanSo.

* iTu: Một số nguyên đại diện cho tử số của phân số.
* iMau: Một số nguyên đại diện cho mẫu số của phân số.

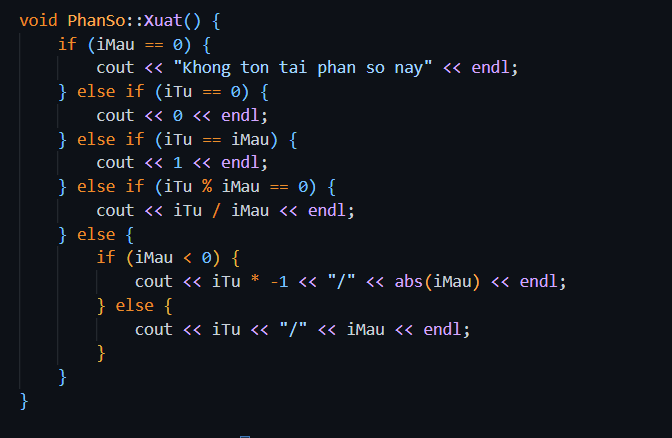
###### Phương thức của lớp PhanSo.

* Nhap(): Nhận dữ liệu từ người dùng, yêu cầu nhập tử số và mẫu số của phân số.



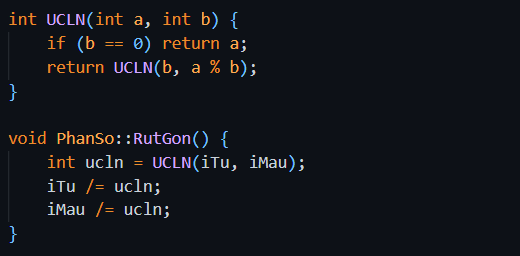
Hình 2.3: Phương thức Nhap() của lớp PhanSo

* Xuat(): Hiển thị phân số dưới dạng chuẩn (rút gọn nếu có thể).
* Hướng giải quyết:
* Nếu iMau (mẫu số) bằng 0, hàm in ra thông báo “Khong ton tai phan so nay”
* Nếu iTu (tử số) bằng 0, hàm in ra giá trị 0.
* Nếu iTu bằng iMau, hàm in ra giá trị 1.
* Nếu iTu chia hết cho iMau, hàm in ra kết quả của phép chia iTu / iMau.
* Trong trường hợp còn lại, hàm xử lý phân số bằng cách kiểm tra dấu của iMau: Nếu iMau âm, hàm in ra phân số dạng (-iTu)/(|iMau|), Ngược lại, hàm in ra phân số dạng iTu/iMau.



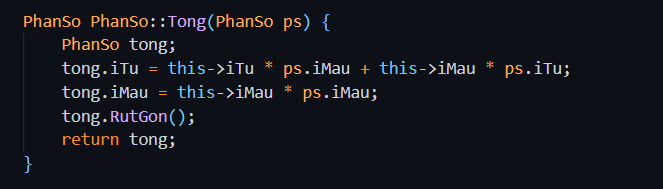
Hình 2.4: Phương thức Nhap() của lớp PhanSo

* RutGon(): Rút gọn phân số bằng cách tìm ước chung lớn nhất của tử số và mẫu số.



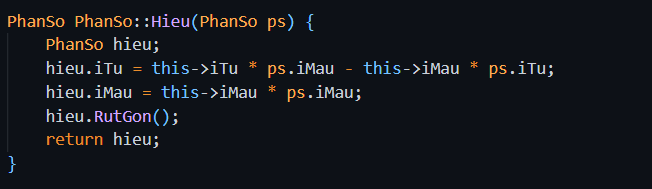
Hình 2.5: Phương thức RutGon() của lớp PhanSo kèm hàm UCLN

* Hướng giải quyết:
* Phương thức này sử dụng thuật toán tìm ước chung lớn nhất (UCLN) của tử số và mẫu số để rút gọn phân số.
* Đầu tiên, tính UCLN của iTu và iMau.
* Sau đó, chia cả iTu và iMau cho UCLN để đạt được phân số rút gọn.
* Tong(PhanSo ps): Tính tổng với một phân số ps khác và trả về kết quả rút gọn.



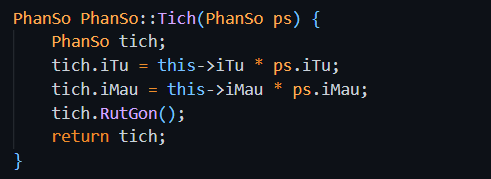
Hình 2.6: Phương thức Tong(PhanSo ps) của lớp PhanSo

* Hướng giải quyết:
* Phương thức này tính tổng của phân số hiện tại với một phân số ps khác.
* Sử dụng công thức: tong.iTu = this->iTu \* ps.iMau + this->iMau \* ps.iTu và tong.iMau = this->iMau \* ps.iMau.
* Sau đó, gọi phương thức RutGon() để rút gọn kết quả.
* Hieu(PhanSo ps): Tính hiệu với một phân số ps khác và trả về kết quả rút gọn.



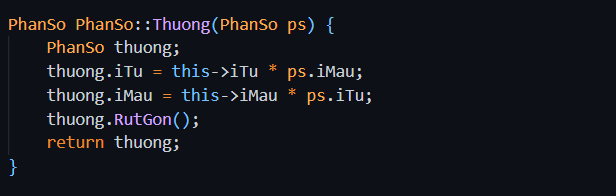
Hình 2.7: Phương thức Hieu(PhanSo ps) của lớp PhanSo

* Hướng giải quyết:
* Phương thức này tính hiệu của phân số hiện tại với một phân số ps khác.
* Sử dụng công thức: hieu.iTu = this->iTu \* ps.iMau - this->iMau \* ps.iTu và hieu.iMau = this->iMau \* ps.iMau.
* Sau đó, gọi phương thức RutGon() để rút gọn kết quả.
* Tich(PhanSo ps): Tính tích với một phân số ps khác và trả về kết quả rút gọn.



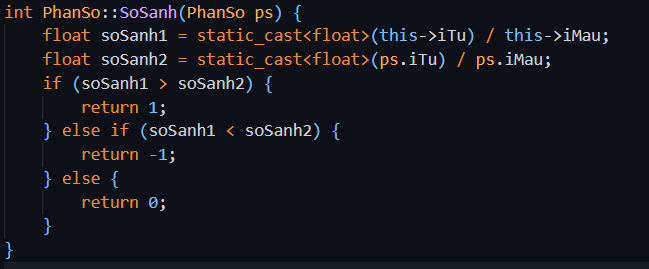
Hình 2.8: Phương thức Tich(PhanSo ps) của lớp PhanSo

* Hướng giải quyết:
* Phương thức này tính tích của phân số hiện tại với một phân số ps khác.
* Sử dụng công thức: tich.iTu = this->iTu \* ps.iTu và tich.iMau = this->iMau \* ps.iMau.
* Sau đó, gọi phương thức RutGon() để rút gọn kết quả.
* Thuong(PhanSo ps): Tính thương với một phân số ps khác và trả về kết quả rút gọn.



Hình 2.9: Phương thức Thuong(PhanSo ps) của lớp PhanSo

* Hướng giải quyết:
* Phương thức này tính thương của phân số hiện tại với một phân số ps khác.
* Sử dụng công thức: thuong.iTu = this->iTu \* ps.iMau và thuong.iMau = this->iMau \* ps.iTu.
* Sau đó, gọi phương thức RutGon() để rút gọn kết quả.
* SoSanh(PhanSo ps): So sánh phân số hiện tại với một phân số ps khác và trả về 1 nếu lớn hơn, -1 nếu nhỏ hơn, và 0 nếu bằng nhau.



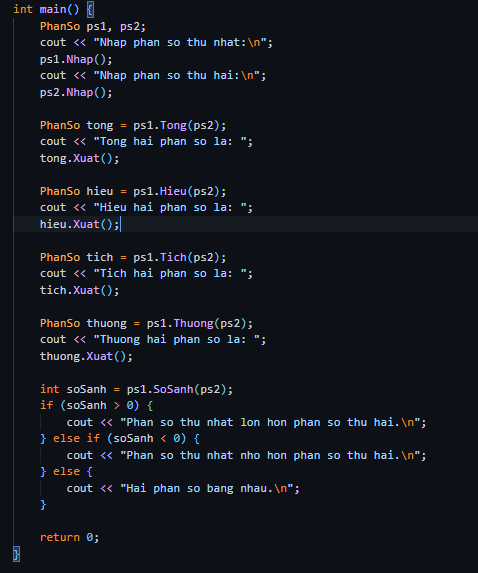
Hình 2.10: Phương thức SoSanh(PhanSo ps) của lớp PhanSo

* Hướng giải quyết:
* float soSanh1 = static\_cast<float>(this->iTu) / this->iMau;
* Đây là bước tính giá trị của phân số hiện tại (this) dưới dạng số thực (float). Chúng ta chuyển tử số (iTu) sang kiểu float và chia cho mẫu số (iMau).
* Kết quả được lưu trong biến soSanh1.
* float soSanh2 = static\_cast<float>(ps.iTu) / ps.iMau;
* Tương tự như trên, đây là bước tính giá trị của phân số ps (đối tượng truyền vào hàm) dưới dạng số thực.
* Kết quả được lưu trong biến soSanh2.
* if (soSanh1 > soSanh2) { return 1; } else if (soSanh1 < soSanh2) { return -1; } else { return 0; }
* Ở đây, chúng ta so sánh giá trị của soSanh1 và soSanh2.
* Nếu soSanh1 lớn hơn soSanh2, hàm trả về 1 (để chỉ ra phân số hiện tại lớn hơn phân số ps).
* Nếu soSanh1 nhỏ hơn soSanh2, hàm trả về -1 (để chỉ ra phân số hiện tại nhỏ hơn phân số ps).
* Nếu soSanh1 bằng soSanh2, hàm trả về 0 (để chỉ ra phân số hiện tại bằng phân số ps).

###### Xác định Input, Output.

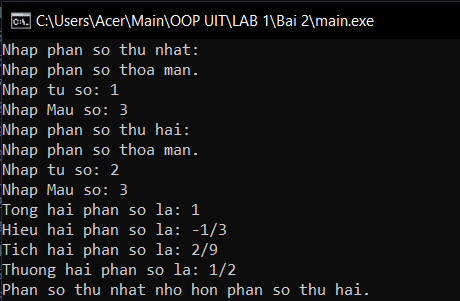
* Input (đầu vào): Tử số và mẫu số của hai phân số được nhập từ bàn phím/ người dùng.
* Output (đầu ra): Phân số dưới dạng Tử số / Mẫu số của tổng, hiệu, tích, thương giữa hai phân số, và kết quả so sánh giữa hai phân số, nếu phân số thứ nhất lớn hơn phân số thứ hai thì xuất “Phan so thu nhat lon hon phan so thu hai.”, nếu phân số thứ nhất nhỏ hơn phân số thứ hai thì xuất "Phan so thu nhat nho hon phan so thu hai.", nếu hai phân số bằng nhau thì xuất "Hai phan so bang nhau.".

###### Hàm main của chương trình bài tập 2.

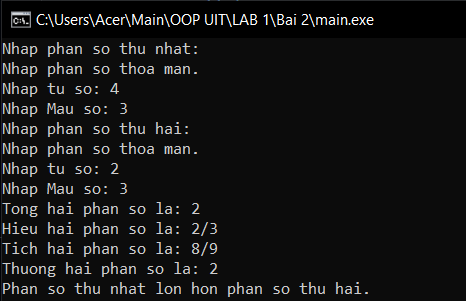


Hình 2.11: Hàm main của chương trình bài tập 2

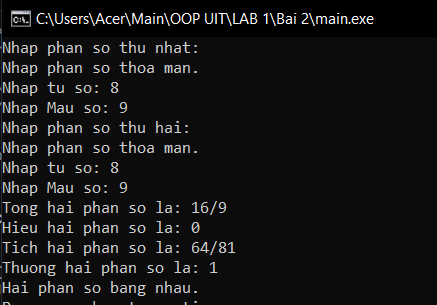
###### Kết quả chạy của chương trình.



Hình 2.12: Kết quả của chương trình ứng với phân số thứ nhất nhỏ hơn phân số thứ hai



Hình 2.13: Kết quả của chương trình ứng với phân số thứ nhất lớn hơn phân số thứ hai



Hình 2.14: Kết quả của chương trình ứng với hai phân số bằng nhau

#### Bài tập 3: Xây dựng lớp số phức bao gồm.

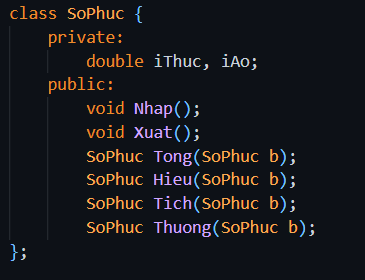
* Thuộc tính: iThuc, iAo
* Phương thức: Nhap(), Xuat(), Tong(), Hieu(), Tich(), Thuong()

**Yêu cầu:** Thực hiện xây dựng lớp, vẽ class diagram và khai báo các thuộc tính,

phương thức. Viết nội dung vào các phương thức đã khai báo. Gọi các phương thức

trong hàm main().

1. Thực hiện xây dựng lớp.



Hình 3.1: Thực hiện xây dựng lớp SoPhuc

1. Class diagram của lớp SoPhuc.



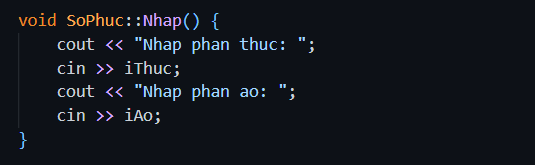
Hình 3.2: Class diagram của lớp SoPhuc

1. Thuộc tính của lớp SoPhuc.

* iThuc: Đây là thuộc tính kiểu double biểu diễn phần thực của số phức.
* iAo: Đây là thuộc tính kiểu double biểu diễn phần ảo của số phức.

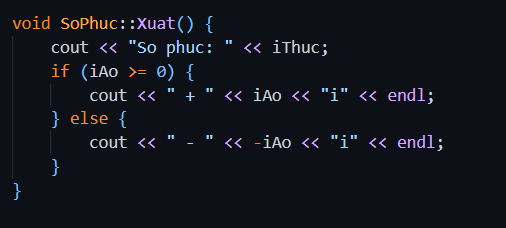
1. Phương thức của lớp SoPhuc:

* Nhap(): Phương thức này không có tham số đầu vào và không trả về giá trị. Nó được sử dụng để nhập giá trị cho phần thực và phần ảo của số phức từ bàn phím.



Hình 3.3: Phương thức Nhap() của lớp SoPhuc

* Xuat(): Phương thức này không có tham số đầu vào và không trả về giá trị. Nó được sử dụng để xuất số phức, nếu phần ảo (iAo) là số dương, sẽ hiển thị dấu cộng; nếu phần ảo là số âm, sẽ hiển thị dấu trừ và giá trị tuyệt đối của phần ảo



Hình 3.4: Phương thức Xuat() của lớp SoPhuc

1. Dưới đây là bảng công thức của tổng, hiệu, tích, giữa hai số phức.

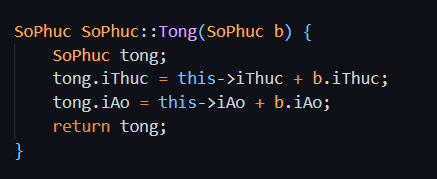
* Có hai số phức:
* a = x + yi.
* b = u + vi.
* Với điều kiện thương giữa 2 số phức: u2 + v2 ≠ 0.

|  |  |
| --- | --- |
| a + b |  |
| a – b |  |
| ab |  |
| a/b |  |

Bảng 1: Bảng công thức tính tổng, hiệu, tích, thương giữa hai số phức

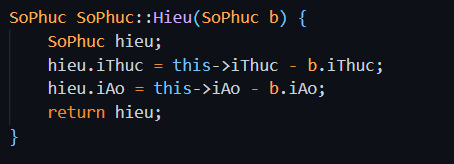
Từ công thức ở bảng 1, ta đã giải quyết vấn đề tính toán tổng, hiệu, thương giữa hai số phức, từ đó xây dựng các phương thức như sau:

* Tong(SoPhuc b): Phương thức này nhận vào một số phức b và trả về tổng của số phức hiện tại và số phức b.



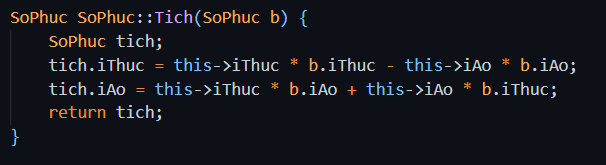
Hình 3.5: Phương thức Tong(SoPhuc b) của lớp SoPhuc

* Hieu(SoPhuc b): Phương thức này nhận vào một số phức b và trả về hiệu của số phức hiện tại và số phức b.



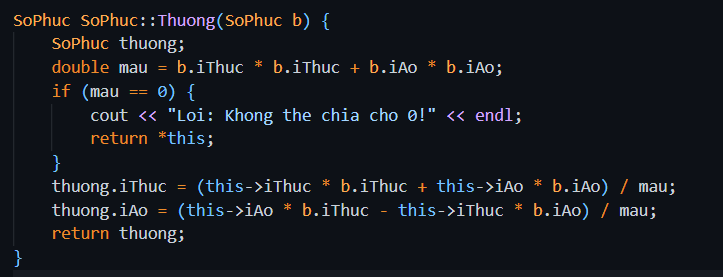
Hình 3.6: Phương thức Hieu(SoPhuc b) của lớp SoPhuc

* Tich(SoPhuc b): Phương thức này nhận vào một số phức b và trả về tích của số phức hiện tại và số phức b.



Hình 3.7: Phương thức Tich(SoPhuc b) của lớp SoPhuc

* Thuong(SoPhuc b): Phương thức này nhận vào một số phức b và trả về thương của số phức hiện tại và số phức b. Nếu số phức b bằng 0, phương thức sẽ in ra thông báo lỗi và trả về số phức hiện tại.

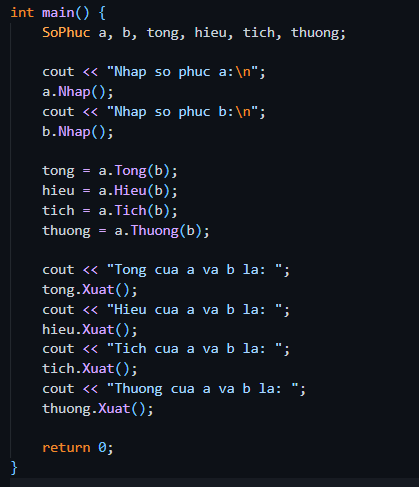


Hình 3.8: Phương thức Thuong(SoPhuc b) của lớp SoPhuc

1. Xác định Input, Output.

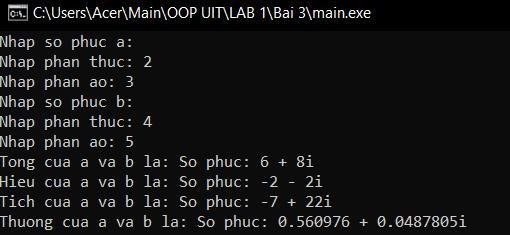
* Input ứng với phương thức Nhap(): nhập giá trị cho phần thực và phần ảo của số phức từ bàn phím.
* Output ứng với phương thức Xuat(): Nếu phần ảo (iAo) là số dương, sẽ hiển thị dấu cộng; nếu phần ảo là số âm, sẽ hiển thị dấu trừ và giá trị tuyệt đối của phần ảo

1. Hàm main của chương trình bài tập 3.



Hình 3.9: Hàm main của chương trình bài tập 3

1. Kết quả chạy của chương trình.



Hình 3.10: Kết quả chạy của chương trình bài tập 3

#### Bài tập 4: Xây dựng lớp giờ phút giây.

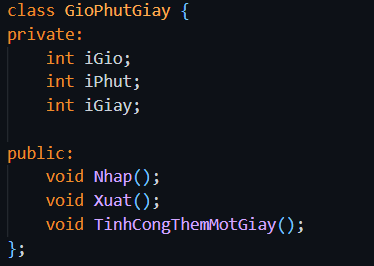
* Thuộc tính: iGio, iPhut, iGiay
* Phương thức: Nhap(), Xuat(), TinhCongThemMotGiay()

**Yêu cầu:** Thực hiện xây dựng lớp, vẽ class diagram và khai báo các thuộc tính,

phương thức. Viết nội dung vào các phương thức đã khai báo. Gọi các phương thức

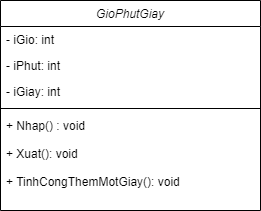
trong hàm main().

1. Thực hiện xây dựng lớp.



Hình 4.1: Thực hiện xây dựng lớp GioPhutGiay

1. Class diagram của lớp GioPhutGiay.



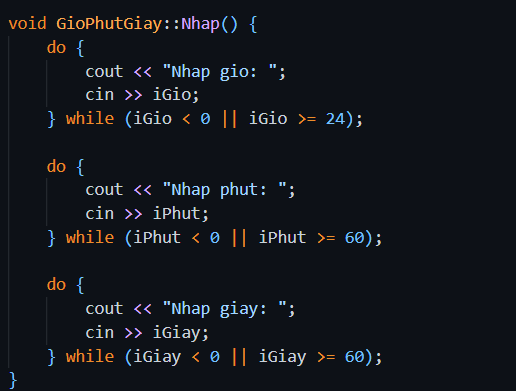
Hình 4.2: Class diagram của lớp GioPhutGiay

1. Thuộc tính của lớp GioPhutGiay.

* iGio: Một số nguyên đại diện cho giờ.
* iPhut: Một số nguyên đại diện cho phút.
* iGiay: Một số nguyên đại diện cho giây.

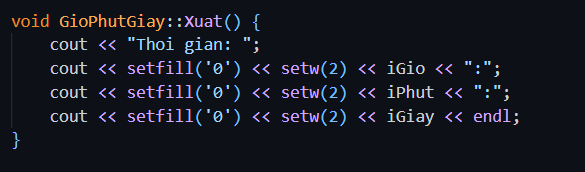
1. Phương thức của lớp GioPhutGiay.

* Nhap(): Phương thức này yêu cầu người dùng nhập giá trị cho iGio, iPhut và iGiay sao cho hợp lệ, nếu nhập không hợp lệ thì nhập lại sao cho hợp lệ.



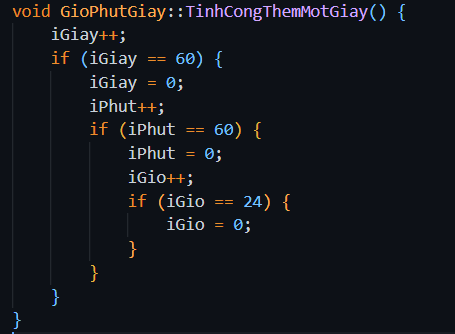
Hình 4.3: Phương thức Nhap() của lớp GioPhutGiay

* Xuat(): Hiển thị thời gian theo định dạng hh:mm:ss.



Hình 4.4: Phương thức Xuat() của class GioPhutGiay

* Giải thích:
* setfill('0') đảm bảo rằng các giá trị giờ, phút và giây được điền vào bằng số 0 ở phía trước nếu chúng có ít hơn 2 chữ số.
* setw(2) đặt chiều rộng của mỗi giá trị là 2 ký tự.
* TinhCongThemMotGiay(): Tăng thời gian lên một giây. Nếu giây đạt 60, nó quay trở lại 0 và tăng phút. Tương tự, nếu phút đạt 60, chúng quay trở lại 0 và tăng giờ. Nếu giờ đạt 24, chúng quay trở lại 0.



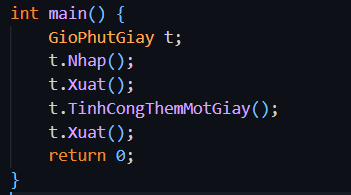
Hình 4.5: Phương thức TinhCongThemMotGiay() của lớp GioPhutGiay

* Hướng giải quyết:
* Xử lý trường hợp giá trị vượt quá ngưỡng:
* Nếu giây vượt quá 59, hãy đặt lại giây về 0 và tăng phút lên 1.
* Tương tự, nếu phút vượt quá 59, đặt lại phút về 0 và tăng giờ lên 1.
* Xử lý trường hợp giá trị giờ vượt quá 23: Nếu giờ vượt quá 23, đặt lại giờ về 0.

1. Xác định Input, Output:

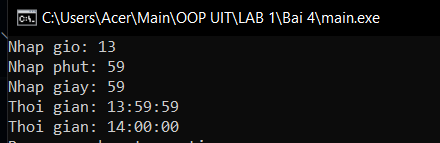
* Input: Người dùng cung cấp giá trị cho iGio, iPhut và iGiay sao cho hợp lệ.
* Output: Lớp hiển thị thời gian theo định dạng hh:mm:ss của người dùng cung cấp và hiển thị thời gian theo định dạng hh:mm:ss sau khi cộng một giây.

1. Hàm main của chương trình.

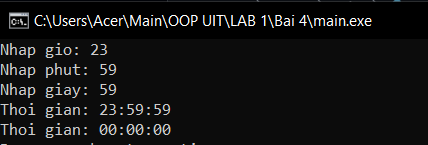


Hình 4.6: Hàm main của chương trình bài tập 4

1. Kết quả chạy của chương trình.



Hình 4.7: Kết quả của chương trình với trường hợp bình thường



Hình 4.8: Kết quả của chương trình ứng với trường hợp 23:59:59

#### Mã nguồn:

<https://drive.google.com/drive/folders/11T5LdvAcYJeP9pK6QJzZQh8drl8RnxR4?usp=drive_link>