

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
LỚP IT002.P11.CTTN.1

NGUYỄN TRỌNG TẤT THÀNH

MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
BÀI TẬP THỰC HÀNH

**OBJECT-ORIENTED PROGRAMMING
PRACTICE EXERCISES**

CỬ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

TP. HỒ CHÍ MINH, NĂM 2024

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
LỚP IT002.P11.CTTN.1

NGUYỄN TRỌNG TẤT THÀNH - 23521455

MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
BÀI TẬP THỰC HÀNH

**OBJECT-ORIENTED PROGRAMMING
PRACTICE EXERCISES**

CỬ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

GIẢNG VIÊN HƯỚNG DẪN
CN. NGUYỄN NGỌC QUÍ

TP. HỒ CHÍ MINH, NĂM 2024

Mục lục

Mục lục	1
Danh sách hình vẽ	2
1 Bài tập 1: Xây dựng lớp Phân số	1
1.1 Đề bài	1
1.2 Class Diagram	1
1.3 Giải thuật	2
1.4 Mã nguồn	3
1.5 Kết quả chạy thử	3
2 Bài tập 2: Xây dựng lớp Số Phức	5
2.1 Đề bài	5
2.2 Class Diagram	5
2.3 Giải thuật	6
2.4 Mã nguồn	6
2.5 Kết quả chạy thử	7
3 Bài tập 3: Xây dựng lớp Thời Gian	8
3.1 Đề bài	8
3.2 Class Diagram	8
3.3 Giải thuật	9
3.4 Mã nguồn	10
3.5 Kết quả chạy thử	10
4 Bài tập 4: Xây dựng lớp Ngày Tháng Năm	11
4.1 Đề bài	11
4.2 Class Diagram	11
4.3 Giải thuật	12
4.4 Mã nguồn	13
4.5 Kết quả chạy thử	13

Danh sách hình vẽ

0.1	Class Diagram của lớp Phân Số	2
0.2	Kết quả chạy thử chương trình cho Bài tập 1	4
0.3	Class Diagram của lớp Số Phức	6
0.4	Kết quả chạy thử chương trình cho Bài tập 2	7
0.5	Class Diagram của lớp Thời Gian	9
0.6	Kết quả chạy thử chương trình cho Bài tập 3	10
0.7	Class Diagram của lớp Ngày Tháng Năm	12
0.8	Kết quả chạy thử chương trình cho Bài tập 4	13

1 Bài tập 1: Xây dựng lớp Phân số

1.1 Đề bài

Xây dựng lớp phân số với các thuộc tính và phương thức sau:

- **Thuộc tính:**

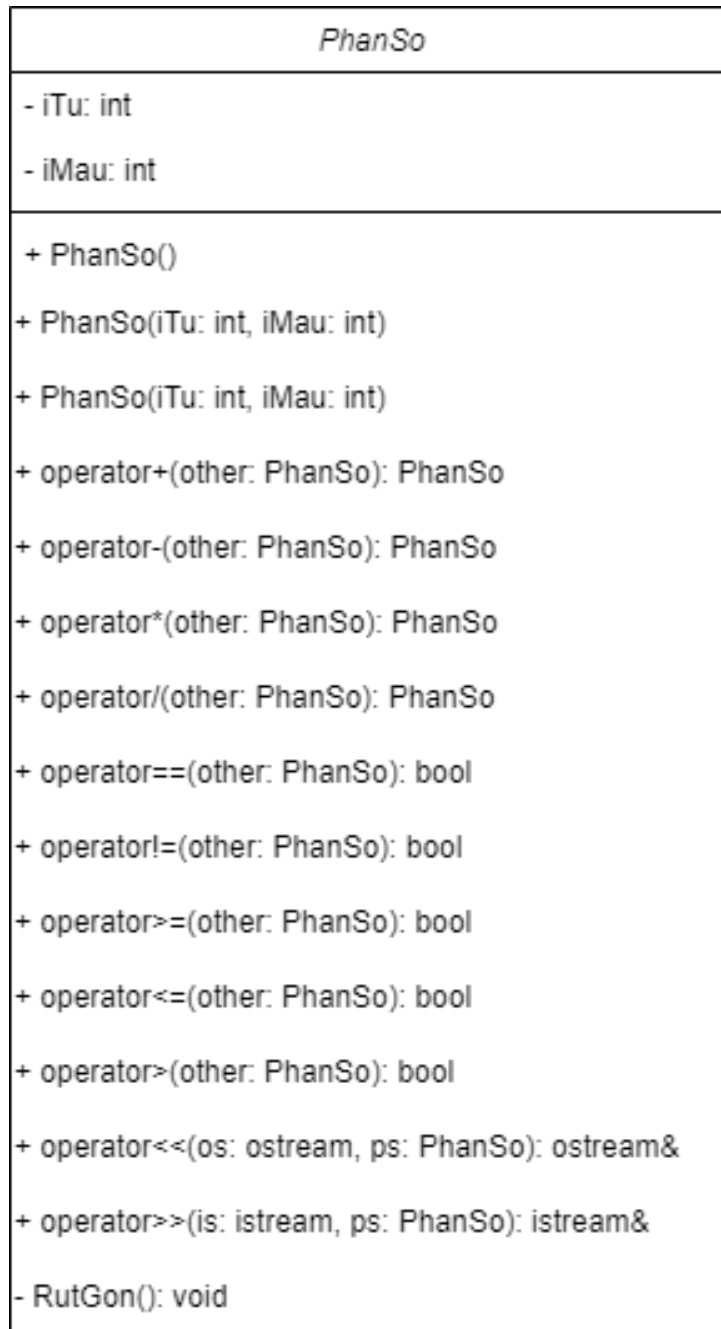
- iTu: Tử số.
- iMau: Mẫu số.

- **Phương thức:**

- PhanSo(): Constructor mặc định.
- PhanSo(int tu, int mau): Constructor có tham số.
- operator+, operator-, operator*, operator/: Các toán tử cộng, trừ, nhân, chia.
- operator==, operator!=, operator>, operator<, operator>=, operator<=: Các toán tử so sánh.
- operator», operator«: Toán tử nhập và xuất.

1.2 Class Diagram

Hình dưới đây là Class Diagram mô tả cấu trúc và các thành phần của lớp Phân Số:



Hình 0.1. Class Diagram của lớp Phân Số

1.3 Giải thuật

Để thực hiện các phép toán trên phân số, chương trình thực hiện các bước sau:

- Định nghĩa các thuộc tính cho tử số (iTU) và mẫu số (iMau).
- Xây dựng các phương thức khởi tạo và các toán tử để thực hiện các phép toán phân số.

-
- Xây dựng hàm `RutGon()` để rút gọn phân số về dạng tối giản.
 - Định nghĩa toán tử `operator«` và `operator»` để hỗ trợ nhập và xuất phân số.

1.4 Mã nguồn

Mã nguồn của lớp Phân Số và các phương thức đã được tải lên và có thể truy cập tại link sau:

- [Mã nguồn trên Google Drive](#)
- [Mã nguồn trên GitHub](#)

1.5 Kết quả chạy thử

Kết quả chạy thử thực tế được thể hiện trong hình dưới đây:

```
Phân số 1: 3/4
Phân số 2: 2/5
Tổng: 23/20
Hiệu: 7/20
Tích: 3/10
Thương: 15/8
Phân số 1 == Phân số 2: 0
Phân số 1 != Phân số 2? Đúng
Phân số 1 > Phân số 2? Đúng
Phân số 1 < Phân số 2? Sai
Phân số 1 >= Phân số 2? Đúng
Phân số 1 <= Phân số 2? Sai
Nhập phân số thứ 3:
Nhập tu so: 10
Nhập mau so: 0
Mau so phải khác 0
Nhập tu so: 100 10
Nhập mau so: Phân số vừa nhập là: 10/1
```

Hình 0.2. Kết quả chạy thử chương trình cho Bài tập 1

2 Bài tập 2: Xây dựng lớp Số Phức

2.1 Đề bài

Xây dựng lớp số phức với các thuộc tính và phương thức sau:

- **Thuộc tính:**

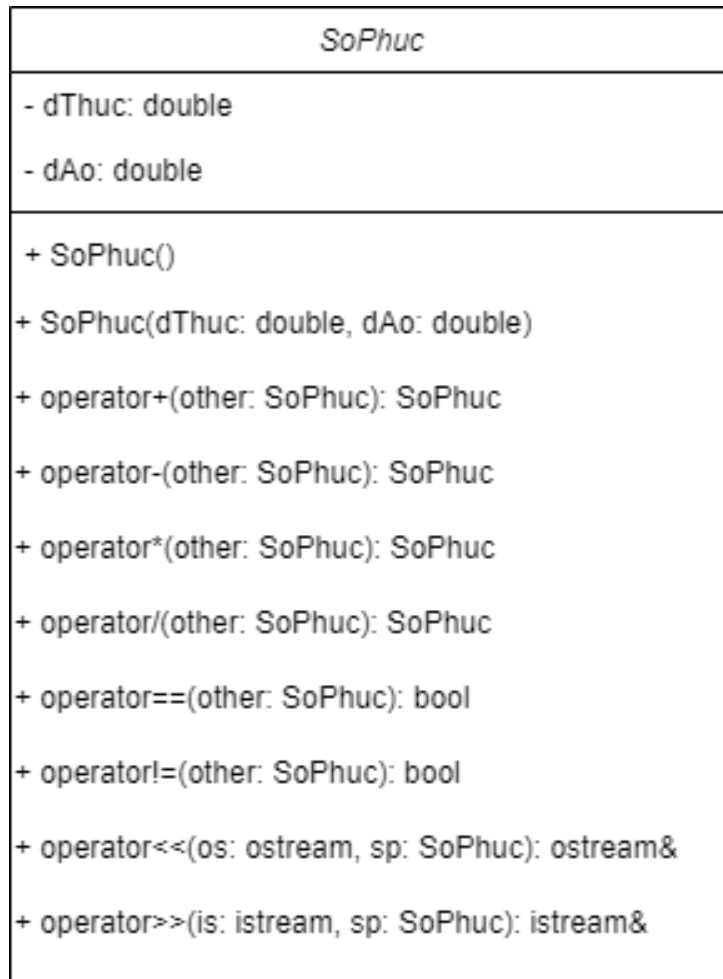
- `dThuc`: Phần thực của số phức.
- `dAo`: Phần ảo của số phức.

- **Phương thức:**

- `SoPhuc()`: Constructor mặc định.
- `SoPhuc(double thuc, double ao)`: Constructor có tham số.
- `operator+`, `operator-`, `operator*`, `operator/`: Các toán tử cộng, trừ, nhân, chia số phức.
- `operator==`, `operator!=`: Các toán tử so sánh bằng và khác.
- `operator»`, `operator«`: Toán tử nhập và xuất.

2.2 Class Diagram

Hình dưới đây là Class Diagram mô tả cấu trúc và các thành phần của lớp Số Phức:



Hình 0.3. Class Diagram của lớp Số Phức

2.3 Giải thuật

Để thực hiện các phép toán trên số phức, chương trình thực hiện các bước sau:

- Định nghĩa các thuộc tính cho phần thực (dThuc) và phần ảo (dAo).
- Xây dựng các phương thức khởi tạo và các toán tử để thực hiện các phép toán số phức.
- Định nghĩa toán tử operator<< và operator>> để hỗ trợ nhập và xuất số phức.

2.4 Mã nguồn

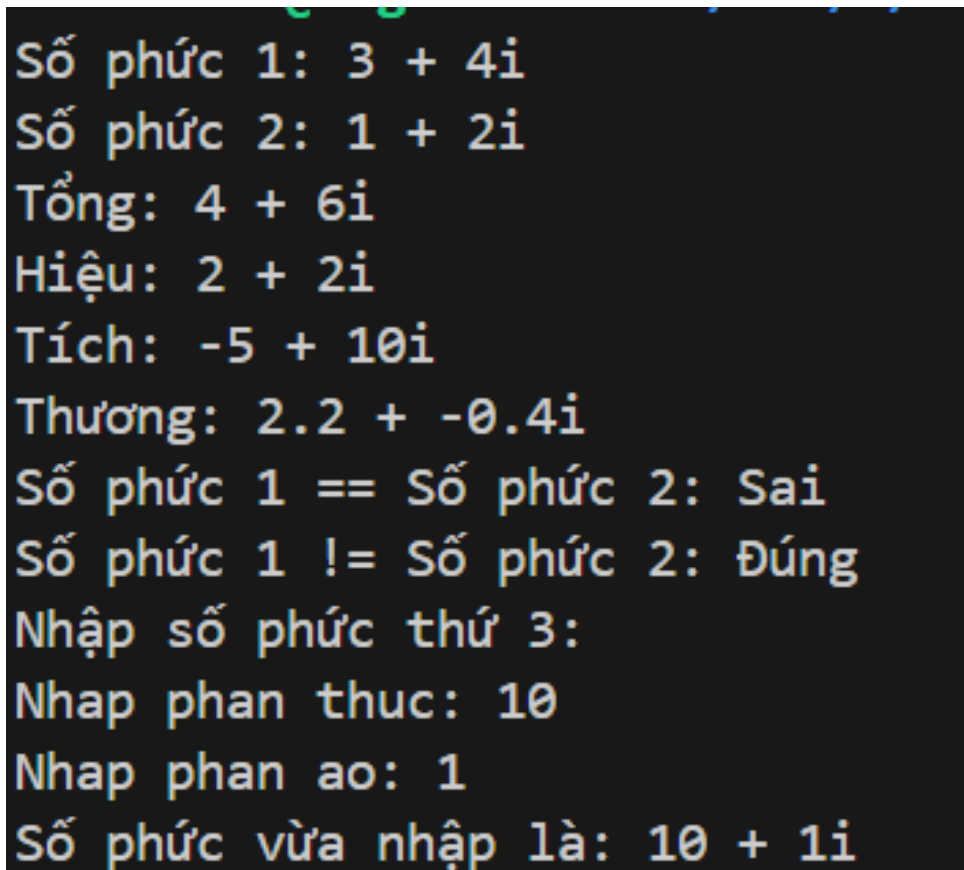
Mã nguồn của lớp Số Phức và các phương thức đã được tải lên và có thể truy cập tại link sau:

- [Mã nguồn trên Google Drive](#)

-
- [Mã nguồn trên GitHub](#)

2.5 Kết quả chạy thử

Kết quả chạy thử thực tế được thể hiện trong hình dưới đây:



```
Số phức 1: 3 + 4i
Số phức 2: 1 + 2i
Tổng: 4 + 6i
Hiệu: 2 + 2i
Tích: -5 + 10i
Thương: 2.2 + -0.4i
Số phức 1 == Số phức 2: Sai
Số phức 1 != Số phức 2: Đúng
Nhập số phức thứ 3:
Nhập phần thực: 10
Nhập phần ảo: 1
Số phức vừa nhập là: 10 + 1i
```

Hình 0.4. Kết quả chạy thử chương trình cho Bài tập 2

3 Bài tập 3: Xây dựng lớp Thời Gian

3.1 Đề bài

Xây dựng lớp thời gian với các thuộc tính và phương thức sau:

- **Thuộc tính:**

- `iGio`: Giờ.
- `iPhut`: Phút.
- `iGiay`: Giây.

- **Phương thức:**

- `ThoiGian()`: Constructor mặc định.
- `ThoiGian(int Gio, int Phut, int Giay)`: Constructor có tham số.
- `TinhGiay()`: Phương thức chuyển thời gian sang giây.
- `TinhLaiGio(int Giay)`: Phương thức thiết lập lại giờ từ tổng số giây.
- `operator+`, `operator-`, `operator++`, `operator--`: Các toán tử cộng, trừ, tăng, giảm thời gian.
- `operator==`, `operator!=`, `operator>`, `operator<`, `operator>=`, `operator<=`: Các toán tử so sánh.
- `operator»`, `operator«`: Toán tử nhập và xuất.

3.2 Class Diagram

Hình dưới đây là Class Diagram mô tả cấu trúc và các thành phần của lớp Thời Gian:

<i>ThoiGian</i>
- iGio: int - iPhut: int - iGiay: int
+ ThoiGian() + ThoiGian(Gio: int, Phut: int, Giay: int) + TinhGiay(): int + TinhLaiGio(Giay: int): void + operator+(Giay: int): ThoiGian + operator-(Giay: int): ThoiGian + operator+(a: ThoiGian): ThoiGian + operator-(a: ThoiGian): ThoiGian + operator++(): ThoiGian + operator--(): ThoiGian + operator==(a: ThoiGian): bool + operator!=(a: ThoiGian): bool + operator>=(a: ThoiGian): bool + operator<=(a: ThoiGian): bool + operator>(a: ThoiGian): bool + operator<(a: ThoiGian): bool

Hình 0.5. Class Diagram của lớp Thời Gian

3.3 Giải thuật

Để thực hiện các phép toán và thao tác trên thời gian, chương trình thực hiện các bước sau:

-
- Định nghĩa các thuộc tính cho giờ (iGio), phút (iPhut), và giây (iGiay).
 - Xây dựng các phương thức khởi tạo và các toán tử để thực hiện các phép toán thời gian.
 - Sử dụng hàm TinhGiay() để chuyển đổi thời gian sang giây và TinhLaiGio(int) để thiết lập lại giờ từ giây.
 - Định nghĩa toán tử operator« và operator» để hỗ trợ nhập và xuất thời gian.

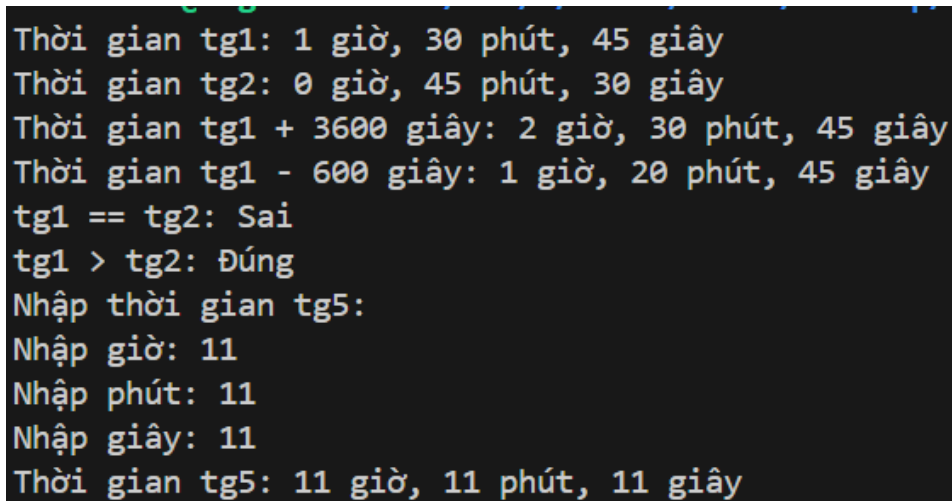
3.4 Mã nguồn

Mã nguồn của lớp Thời Gian và các phương thức đã được tải lên và có thể truy cập tại link sau:

- [Mã nguồn trên Google Drive](#)
- [Mã nguồn trên GitHub](#)

3.5 Kết quả chạy thử

Kết quả chạy thử thực tế được thể hiện trong hình dưới đây:



```
Thời gian tg1: 1 giờ, 30 phút, 45 giây
Thời gian tg2: 0 giờ, 45 phút, 30 giây
Thời gian tg1 + 3600 giây: 2 giờ, 30 phút, 45 giây
Thời gian tg1 - 600 giây: 1 giờ, 20 phút, 45 giây
tg1 == tg2: Sai
tg1 > tg2: Đúng
Nhập thời gian tg5:
Nhập giờ: 11
Nhập phút: 11
Nhập giây: 11
Thời gian tg5: 11 giờ, 11 phút, 11 giây
```

Hình 0.6. Kết quả chạy thử chương trình cho Bài tập 3

4 Bài tập 4: Xây dựng lớp Ngày Tháng Năm

4.1 Đề bài

Xây dựng lớp Ngày Tháng Năm với các thuộc tính và phương thức sau:

- **Thuộc tính:**

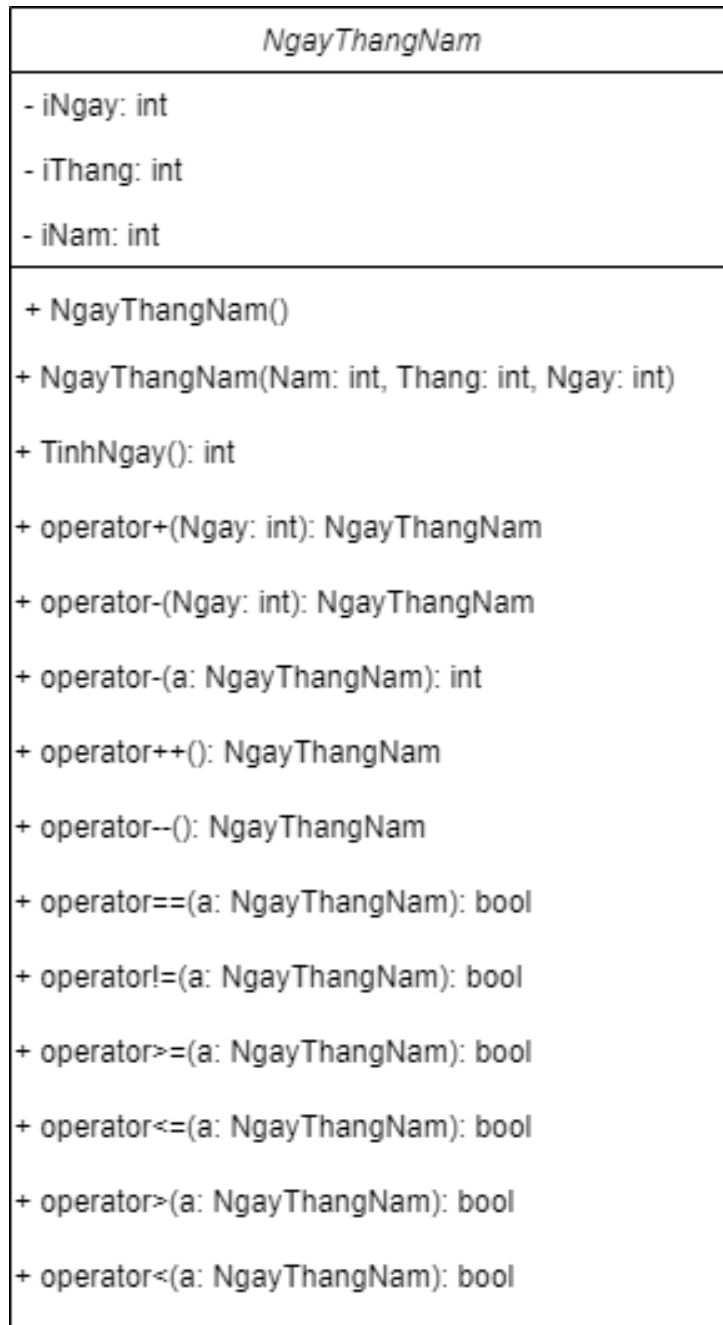
- `iNgay`: Ngày.
- `iThang`: Tháng.
- `iNam`: Năm.

- **Phương thức:**

- `NgayThangNam()`: Constructor mặc định.
- `NgayThangNam(int Nam, int Thang, int Ngay)`: Constructor có tham số.
- `TinhNgay()`: Phương thức tính toán số ngày từ ngày đầu tiên của năm đến ngày hiện tại.
- `operator+`, `operator-`, `operator++`, `operator--`: Các toán tử cộng, trừ, tăng, giảm ngày.
- `operator==`, `operator!=`, `operator>`, `operator<`, `operator>=`, `operator<=`: Các toán tử so sánh.
- `operator»`, `operator«`: Toán tử nhập và xuất.

4.2 Class Diagram

Hình dưới đây là Class Diagram mô tả cấu trúc và các thành phần của lớp Ngày Tháng Năm:



Hình 0.7. Class Diagram của lớp Ngày Tháng Năm

4.3 Giải thuật

Để thực hiện các phép toán và thao tác trên ngày tháng năm, chương trình thực hiện các bước sau:

- Định nghĩa các thuộc tính cho ngày (iNgày), tháng (iThang), và năm (iNam).

-
- Xây dựng các phương thức khởi tạo và các toán tử để thực hiện các phép toán với ngày tháng năm.
 - Sử dụng hàm `TinhNgay()` để tính số ngày từ đầu năm đến ngày hiện tại.
 - Định nghĩa toán tử `operator«` và `operator»` để hỗ trợ nhập và xuất ngày tháng năm.

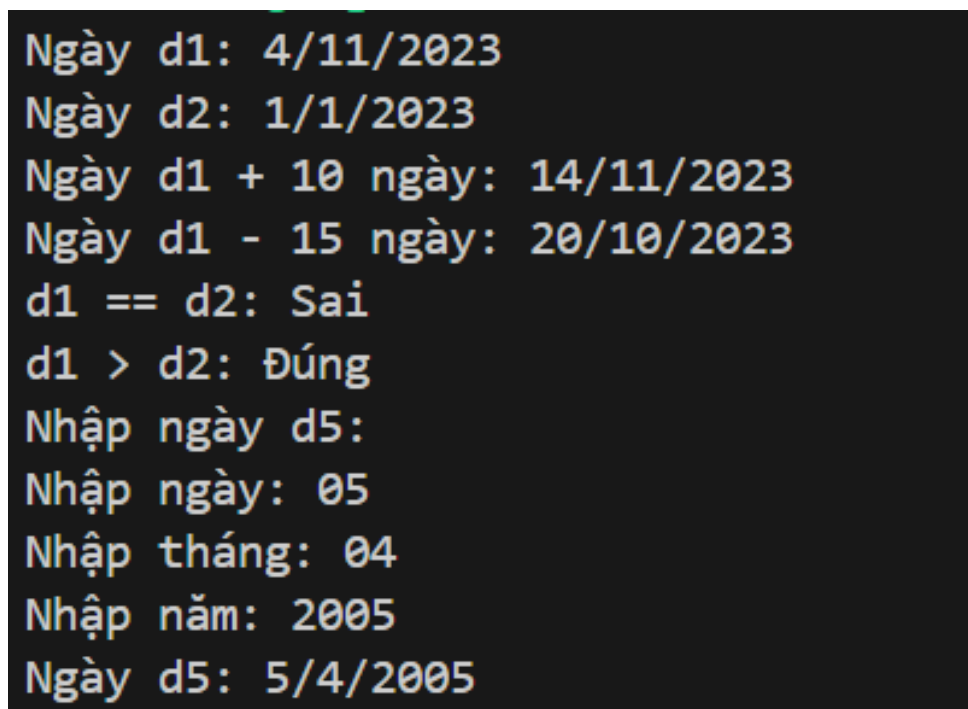
4.4 Mã nguồn

Mã nguồn của lớp Ngày Tháng Năm và các phương thức đã được tải lên và có thể truy cập tại link sau:

- [Mã nguồn trên Google Drive](#)
- [Mã nguồn trên GitHub](#)

4.5 Kết quả chạy thử

Kết quả chạy thử thực tế được thể hiện trong hình dưới đây:



```
Ngày d1: 4/11/2023
Ngày d2: 1/1/2023
Ngày d1 + 10 ngày: 14/11/2023
Ngày d1 - 15 ngày: 20/10/2023
d1 == d2: Sai
d1 > d2: Đúng
Nhập ngày d5:
Nhập ngày: 05
Nhập tháng: 04
Nhập năm: 2005
Ngày d5: 5/4/2005
```

Hình 0.8. Kết quả chạy thử chương trình cho Bài tập 4