

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA CÔNG NGHỆ PHẦN MỀM

BÙI NGỌC THIÊN THANH - 23521436

MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG  
BÀI TẬP THỰC HÀNH 3

**OBJECT-ORIENTED PROGRAMMING  
CLASSWORK 3**

CỬ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

GIẢNG VIÊN HƯỚNG DẪN  
CN. NGUYỄN NGỌC QUÍ

TP. HỒ CHÍ MINH, THÁNG 11 NĂM 2024



# Mục lục

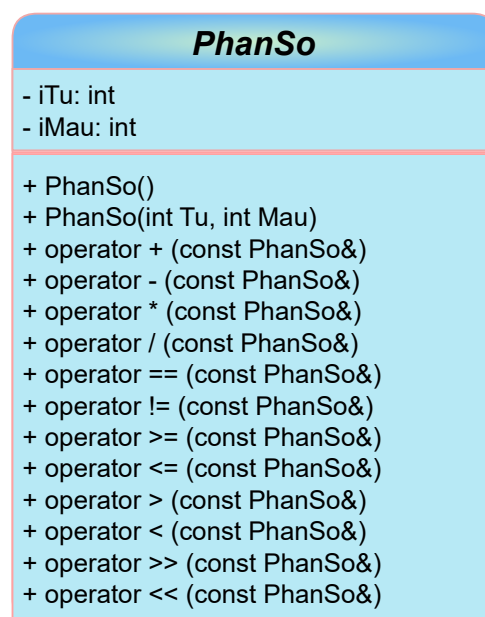
<b>Mục lục</b>	<b>i</b>
<b>1. Bài tập 1</b>	<b>1</b>
1.1 Biểu đồ lớp . . . . .	1
1.2 Mô tả đầu vào và đầu ra . . . . .	1
1.3 Ý tưởng . . . . .	2
1.4 Thực thi code . . . . .	2
1.5 Kiểm thử . . . . .	5
<b>2. Bài tập 2</b>	<b>7</b>
2.1 Biểu đồ lớp . . . . .	7
2.2 Mô tả đầu vào và đầu ra . . . . .	7
2.3 Ý tưởng . . . . .	8
2.4 Thực thi code . . . . .	8
2.5 Kiểm thử . . . . .	10
<b>3. Bài tập 3</b>	<b>13</b>
3.1 Biểu đồ lớp . . . . .	13
3.2 Mô tả đầu vào và đầu ra . . . . .	14
3.3 Ý tưởng . . . . .	14
3.4 Thực thi code . . . . .	14
3.5 Kiểm thử . . . . .	19
<b>4. Bài tập 4</b>	<b>21</b>
4.1 Biểu đồ lớp . . . . .	21
4.2 Mô tả đầu vào và đầu ra . . . . .	21
4.3 Ý tưởng . . . . .	22
4.4 Thực thi code . . . . .	22
4.5 Kiểm thử . . . . .	27

# 1. Bài tập 1

Xây dựng lớp phân số:

- **Thuộc tính:** iTu, iMau
- **Phương thức:** PhanSo(), PhanSo(int Tu, int Mau)
- **Thực hiện các phương thức operator:** +, -, \*, /, ==, !=, >=, <=, >, <, >>, <<.

## 1.1 Biểu đồ lớp



Hình 1.1: Sơ đồ khởi bài 1

## 1.2 Mô tả đầu vào và đầu ra

- **Đầu vào:**
  - Một số nguyên n đại diện cho số lượng phân số cần nhập.



- Sau đó, nhập n phân số với tử số và mẫu số. Mẫu số phải khác 0, nếu không, yêu cầu người dùng nhập lại mẫu số hợp lệ.

- **Đầu ra:**

- Phân số có giá trị nhỏ nhất và phân số có giá trị lớn nhất (không cần rút gọn), in dưới dạng tử số / mẫu số.

## 1.3 Ý tưởng

- Xây dựng lớp PhanSo với các thuộc tính tử số và mẫu số.
- Viết hàm khởi tạo để khởi tạo giá trị ban đầu cho phân số, đồng thời rút gọn phân số bằng cách sử dụng ước chung lớn nhất (UCLN).
- Định nghĩa các toán tử số học để thực hiện các phép tính cộng, trừ, nhân, chia phân số.
- Định nghĩa các toán tử so sánh bằng cách quy về so sánh nhân chéo của tử và mẫu.
- Định nghĩa toán tử nhập và xuất để nhận phân số từ bàn phím và in ra màn hình.

## 1.4 Thực thi code

### Bài 1

- **PhanSo.h**

```
1  #ifndef PHANSO_H
2  #define PHANSO_H
3
4  #include <iostream>
5
6  class PhanSo {
7  private:
8      int iTu;
9      int iMau;
10
11      void simplify();
12
13 public:
```



```

14     PhanSo();
15     PhanSo(int Tu, int Mau);
16
17     PhanSo operator+(const PhanSo& other);
18     PhanSo operator-(const PhanSo& other);
19     PhanSo operator*(const PhanSo& other);
20     PhanSo operator/(const PhanSo& other);
21
22     bool operator==(const PhanSo& other) const;
23     bool operator!=(const PhanSo& other) const;
24     bool operator>=(const PhanSo& other) const;
25     bool operator<=(const PhanSo& other) const;
26     bool operator>(const PhanSo& other) const;
27     bool operator<(const PhanSo& other) const;
28
29     friend std::istream& operator>>(std::istream& in, PhanSo& ps);
30     friend std::ostream& operator<<(std::ostream& out, const PhanSo& ps);
31 };
32
33 #endif

```

### • PhanSo.cpp

```

1     #include "PhanSo.h"
2     #include <iostream>
3     #include <numeric>
4
5     PhanSo::PhanSo() : iTu(0), iMau(1) {}
6
7     PhanSo::PhanSo(int Tu, int Mau) : iTu(Tu), iMau(Mau) {
8         simplify();
9     }
10
11 void PhanSo::simplify() {
12     int gcd = std::gcd(iTu, iMau);
13     iTu /= gcd;
14     iMau /= gcd;
15     if (iMau < 0) { // Ensure denominator is positive

```



```
16     iTu = -iTu;
17     iMau = -iMau;
18 }
19 }
20
21 PhanSo PhanSo::operator + (const PhanSo& other) {
22     return PhanSo(iTu * other.iMau + other.iTu * iMau, iMau * other.iMau);
23 }
24
25 PhanSo PhanSo::operator - (const PhanSo& other) {
26     return PhanSo(iTu * other.iMau - other.iTu * iMau, iMau * other.iMau);
27 }
28
29 PhanSo PhanSo::operator * (const PhanSo& other) {
30     return PhanSo(iTu * other.iTu, iMau * other.iMau);
31 }
32
33 PhanSo PhanSo::operator / (const PhanSo& other) {
34     return PhanSo(iTu * other.iMau, iMau * other.iTu);
35 }
36
37 bool PhanSo::operator == (const PhanSo& other) const {
38     return iTu * other.iMau == iMau * other.iTu;
39 }
40
41 bool PhanSo::operator != (const PhanSo& other) const {
42     return !(*this == other);
43 }
44
45 bool PhanSo::operator >= (const PhanSo& other) const {
46     return iTu * other.iMau >= iMau * other.iTu;
47 }
48
49 bool PhanSo::operator <= (const PhanSo& other) const {
50     return iTu * other.iMau <= iMau * other.iTu;
51 }
52
53 bool PhanSo::operator > (const PhanSo& other) const {
```



```
54     return iTu * other.iMau > iMau * other.iTu;
55 }
56
57 bool PhanSo::operator < (const PhanSo& other) const {
58     return iTu * other.iMau < iMau * other.iTu;
59 }
60
61 std::istream& operator >> (std::istream& in, PhanSo& ps) {
62     in >> ps.iTu >> ps.iMau;
63     ps.simplify();
64     return in;
65 }
66
67 std::ostream& operator << (std::ostream& out, const PhanSo& ps) {
68     out << ps.iTu << "/" << ps.iMau;
69     return out;
70 }
```

## 1.5 Kiểm thử

- Testcase 1:

```
PS D:\THANH\HK3\OOP\LAB_3\Bai1> g++ main.cpp PhanSo.cpp -o run
PS D:\THANH\HK3\OOP\LAB_3\Bai1> ./run
Nhap phan so 1 (tu mau): 1 2
Nhap phan so 2 (tu mau): 3 4
Phan so 1: 1/2
Phan so 2: 3/4
Tong: 5/4
Hieu: -1/4
Tich: 3/8
Thuong: 2/3
So sanh ps1 == ps2: False
```

- Testcase 2:





```
PS D:\THANH\HK3\OOP\LAB_3\Bai1> g++ main.cpp PhanSo.cpp -o run
PS D:\THANH\HK3\OOP\LAB_3\Bai1> ./run
Nhap phan so 1 (tu mau): 2 3
Nhap phan so 2 (tu mau): -2 3
Phan so 1: 2/3
Phan so 2: -2/3
Tong: 0/1
Hieu: 4/3
Tich: -4/9
Thuong: -1/1
So sanh ps1 == ps2: False
```

- Testcase 3:

```
PS D:\THANH\HK3\OOP\LAB_3\Bai1> g++ main.cpp PhanSo.cpp -o run
PS D:\THANH\HK3\OOP\LAB_3\Bai1> ./run
Nhap phan so 1 (tu mau): 5 10
Nhap phan so 2 (tu mau): 1 2
Phan so 1: 1/2
Phan so 2: 1/2
Tong: 1/1
Hieu: 0/1
Tich: 1/4
Thuong: 1/1
So sanh ps1 == ps2: True
```

- Testcase 4:

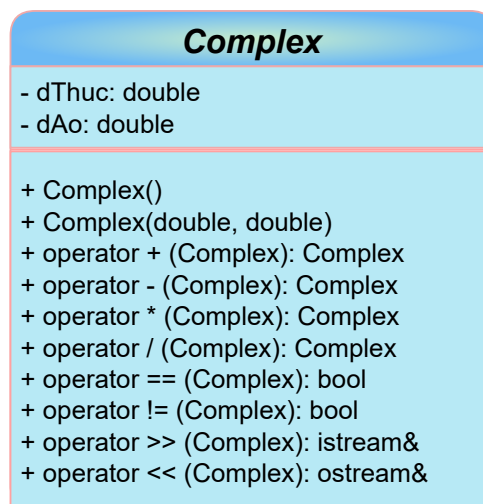
```
PS D:\THANH\HK3\OOP\LAB_3\Bai1> g++ main.cpp PhanSo.cpp -o run
PS D:\THANH\HK3\OOP\LAB_3\Bai1> ./run
Nhap phan so 1 (tu mau): 7 49
Nhap phan so 2 (tu mau): -1 7
Phan so 1: 1/7
Phan so 2: -1/7
Tong: 0/1
Hieu: 2/7
Tich: -1/49
Thuong: -1/1
So sanh ps1 == ps2: False
```

## 2. Bài tập 2

Xây dựng lớp số phức:

- **Thuộc tính:** dThuc, dAo
- **Phương thức:** SoPhuc(), SoPhuc (int thuc, int ao)
- **Thực hiện các phương thức operator:** +, -, \*, /, ==, !=, >, <.

### 2.1 Biểu đồ lớp



Hình 2.2: Sơ đồ khối bài 2

### 2.2 Mô tả đầu vào và đầu ra

- **Đầu vào:**
  - Người dùng nhập hai số phức c1 và c2, mỗi số phức gồm phần thực và phần ảo.
  - Dữ liệu nhập cho mỗi số phức sẽ được yêu cầu lần lượt với hai giá trị: phần thực và phần ảo.
- **Đầu ra:**
  - Kết quả của các phép toán cộng, trừ, nhân, chia giữa c1 và c2.



- Kiểm tra tính bằng nhau (==) và khác nhau (!=) giữa c1 và c2.
- Mỗi kết quả sẽ được xuất ra với định dạng của số phức (dạng  $a + bi$ ).

## 2.3 Ý tưởng

Lớp Complex được xây dựng để mô tả và thực hiện các phép toán số phức. Các phép toán số học và so sánh được triển khai dưới dạng nạp chồng toán tử để dễ dàng sử dụng, giúp chương trình có thể thực hiện các thao tác trên số phức một cách tự nhiên.

## 2.4 Thực thi code

Link code: [Bài 2](#)

### • Complex.h

```
1 // Complex.h
2 #ifndef COMPLEX_H
3 #define COMPLEX_H
4
5 #include <iostream>
6
7 class Complex {
8 private:
9     double dThuc; // Real part
10    double dAo; // Imaginary part
11
12 public:
13     // Constructors
14     Complex();
15     Complex(double thuc, double ao);
16
17     // Operator overloads
18     Complex operator+(const Complex& other) const;
19     Complex operator-(const Complex& other) const;
20     Complex operator*(const Complex& other) const;
21     Complex operator/(const Complex& other) const;
22     bool operator==(const Complex& other) const;
23     bool operator!=(const Complex& other) const;
```



```

24
25     // Input and output stream overloads
26     friend std::ostream& operator<<(std::ostream& out, const Complex& c);
27     friend std::istream& operator>>(std::istream& in, Complex& c);
28 };
29
30 #endif

```

### • Complex.cpp

```

1     // Complex.cpp
2     #include "Complex.h"
3     #include <iostream>
4     #include <cmath>
5
6     Complex::Complex() : dThuc(0), dAo(0) {}
7
8     Complex::Complex(double thuc, double ao) : dThuc(thuc), dAo(ao) {}
9
10    Complex Complex::operator+(const Complex& other) const {
11        return Complex(dThuc + other.dThuc, dAo + other.dAo);
12    }
13
14    Complex Complex::operator-(const Complex& other) const {
15        return Complex(dThuc - other.dThuc, dAo - other.dAo);
16    }
17
18    Complex Complex::operator*(const Complex& other) const {
19        double real = dThuc * other.dThuc - dAo * other.dAo;
20        double imaginary = dThuc * other.dAo + dAo * other.dThuc;
21        return Complex(real, imaginary);
22    }
23
24    Complex Complex::operator/(const Complex& other) const {
25        double denominator = other.dThuc * other.dThuc + other.dAo * other.dAo;
26        double real = (dThuc * other.dThuc + dAo * other.dAo) / denominator;
27        double imaginary = (dAo * other.dThuc - dThuc * other.dAo) /
            denominator;

```



```
28     return Complex(real, imaginary);
29 }
30
31 bool Complex::operator==(const Complex& other) const {
32     return dThuc == other.dThuc && dAo == other.dAo;
33 }
34
35 bool Complex::operator!=(const Complex& other) const {
36     return !(*this == other);
37 }
38
39 std::ostream& operator<<(std::ostream& out, const Complex& c) {
40     out << c.dThuc << " + " << c.dAo << "i";
41     return out;
42 }
43
44 std::istream& operator>>(std::istream& in, Complex& c) {
45     std::cout << "Enter real part: ";
46     in >> c.dThuc;
47     std::cout << "Enter imaginary part: ";
48     in >> c.dAo;
49     return in;
50 }
```

## 2.5 Kiểm thử

- Testcase 1:



```
PS D:\THANH\HK3\OOP\LAB_3\Bai2> g++ main.cpp Complex.cpp -o run
PS D:\THANH\HK3\OOP\LAB_3\Bai2> ./run
Nhap so phuc c1:
Enter real part: 3
Enter imaginary part: 4
Nhap so phuc c2:
Enter real part: 1
Enter imaginary part: -2
Tong: 4 + 2i
Hieu: 2 + 6i
Tich: 11 + -2i
Thuong: -1 + 2i
c1 == c2: False
c1 != c2: True
```

- Testcase 2:

```
PS D:\THANH\HK3\OOP\LAB_3\Bai2> g++ main.cpp Complex.cpp -o run
PS D:\THANH\HK3\OOP\LAB_3\Bai2> ./run
Nhap so phuc c1:
Enter real part: 5
Enter imaginary part: 3
Nhap so phuc c2:
Enter real part: 5
Enter imaginary part: 3
Tong: 10 + 6i
Hieu: 0 + 0i
Tich: 16 + 30i
Thuong: 1 + 0i
c1 == c2: True
c1 != c2: False
```

- Testcase 3:

```
PS D:\THANH\HK3\OOP\LAB_3\Bai2> g++ main.cpp Complex.cpp -o run
PS D:\THANH\HK3\OOP\LAB_3\Bai2> ./run
Nhap so phuc c1:
Enter real part: 2
Enter imaginary part: 3
Nhap so phuc c2:
Enter real part: 0
Enter imaginary part: 1
Tong: 2 + 4i
Hieu: 2 + 2i
Tich: -3 + 2i
Thuong: 3 + -2i
c1 == c2: False
c1 != c2: True
```



- Testcase 4:

```
PS D:\THANH\HK3\OOP\LAB_3\Bai2> g++ main.cpp Complex.cpp -o run
PS D:\THANH\HK3\OOP\LAB_3\Bai2> ./run
Nhap so phuc c1:
Enter real part: 8
Enter imaginary part: -7
Nhap so phuc c2:
Enter real part: 3
Enter imaginary part: -4
Tong: 11 + -11i
Hieu: 5 + -3i
Tich: -4 + -53i
Thuong: 2.08 + 0.44i
c1 == c2: False
c1 != c2: True
```

- Testcase 5:

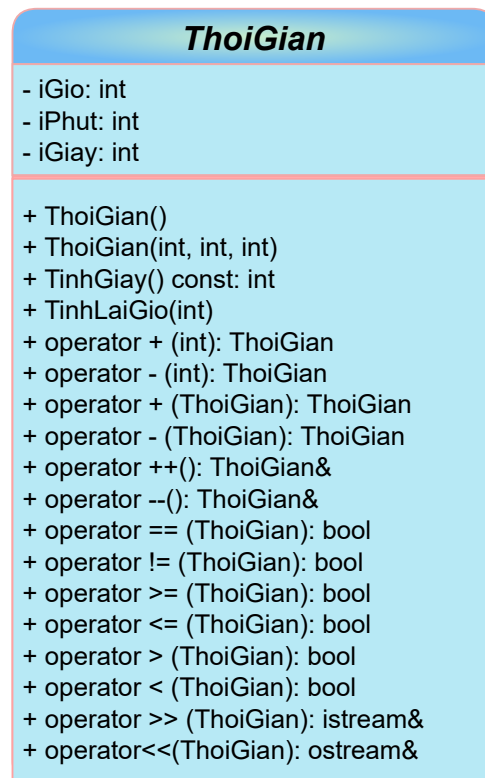
```
PS D:\THANH\HK3\OOP\LAB_3\Bai2> g++ main.cpp Complex.cpp -o run
PS D:\THANH\HK3\OOP\LAB_3\Bai2> ./run
Nhap so phuc c1:
Enter real part: 6
Enter imaginary part: 10
Nhap so phuc c2:
Enter real part: 6
Enter imaginary part: 10
Tong: 12 + 20i
Hieu: 0 + 0i
Tich: -64 + 120i
Thuong: 1 + 0i
c1 == c2: True
c1 != c2: False
```

## 3. Bài tập 3

Xây dựng lớp thời gian:

- **Thuộc tính:** iGio, iPhut, iGiay
- **Phương thức:** ThoiGian(), ThoiGian (int Gio, int Phut, int Giay), TinhGiay(), TinhLaiGio(int Giay)
- **Thực hiện các phương thức operator:** +(int Giay), -(int Giay), +(ThoiGian a), -(ThoiGian a), ++, --, ==, !=, >=, <=, >, <, >>, <<.

### 3.1 Biểu đồ lớp



Hình 3.3: Sơ đồ khối bài 3





## 3.2 Mô tả đầu vào và đầu ra

- **Đầu vào:**
  - Nhập vào ba giá trị: giờ, phút, giây
  - Nhập các giá trị giây ( hoặc ) hoặc một đối tượng thời gian để thực hiện phép toán giữa hai thời gian.
- **Đầu ra:**
  - Xuất Thời Gian: A giờ, B phút, C giây
  - Kết quả phép toán.

## 3.3 Ý tưởng

- **Khởi tạo:** Sử dụng constructor để khởi tạo các thuộc tính của lớp ThoiGian từ đầu vào.
- **Tính giây:** Phương thức TinhGiay() tính tổng số giây từ giờ, phút và giây.
- **Phép toán:** Các phép toán được thực hiện thông qua nạp chồng toán tử, cho phép thực hiện các phép cộng, trừ, so sánh giữa hai đối tượng thời gian hoặc với số giây.
- **Xử lý kết quả:** Kết quả được xuất ra với định dạng dễ hiểu, đồng thời thực hiện tự động hóa cho các trường hợp chuyển đổi (ví dụ: khi giây vượt quá 60, tự động chuyển sang phút và giờ).

## 3.4 Thực thi code

Link code: [Bài 3](#)

- **ThoiGian.h**

```
1 // ThoiGian.h
2 #ifndef THOIGIAN_H
3 #define THOIGIAN_H
4
5 #include <iostream>
6
7 class ThoiGian {
```



```

8     private:
9         int iGio;
10        int iPhut;
11        int iGiay;
12
13    public:
14        // Constructors
15        ThoiGian();
16        ThoiGian(int Gio, int Phut, int Giay);
17
18        // Method to calculate total seconds
19        int TinhGiay() const;
20
21        // Method to recalculate hours, minutes, seconds from seconds
22        void TinhLaiGio(int totalSeconds);
23
24        // Operator overloads
25        ThoiGian operator + (int Giay);
26        ThoiGian operator - (int Giay);
27        ThoiGian operator + (const ThoiGian& a);
28        ThoiGian operator - (const ThoiGian& a);
29
30        ThoiGian& operator++(); // Prefix increment
31        ThoiGian operator++(int); // Postfix increment
32        ThoiGian& operator--(); // Prefix decrement
33        ThoiGian operator--(int); // Postfix decrement
34
35        bool operator == (const ThoiGian& a) const;
36        bool operator != (const ThoiGian& a) const;
37        bool operator >= (const ThoiGian& a) const;
38        bool operator <= (const ThoiGian& a) const;
39        bool operator > (const ThoiGian& a) const;
40        bool operator < (const ThoiGian& a) const;
41
42        friend std::istream& operator>>(std::istream& in, ThoiGian& tg);
43        friend std::ostream& operator<<(std::ostream& out, const ThoiGian& tg);
44    };
45

```



46     #endif

### • ThoiGian.cpp

```
1     // ThoiGian.cpp
2     #include "ThoiGian.h"
3
4     // Constructors
5     ThoiGian::ThoiGian() : iGio(0), iPhut(0), iGiay(0) {}
6     ThoiGian::ThoiGian(int Gio, int Phut, int Giay) : iGio(Gio), iPhut(Phut),
7         iGiay(Giay) {}
8
9     // Calculate total seconds
10    int ThoiGian::TinhGiay() const {
11        return iGio * 3600 + iPhut * 60 + iGiay;
12    }
13
14    // Recalculate hours, minutes, seconds from total seconds
15    void ThoiGian::TinhLaiGio(int totalSeconds) {
16        iGio = totalSeconds / 3600;
17        iPhut = (totalSeconds % 3600) / 60;
18        iGiay = totalSeconds % 60;
19    }
20
21    // Operator overloads for +, -
22    ThoiGian ThoiGian::operator+(int Giay) {
23        int totalSeconds = TinhGiay() + Giay;
24        ThoiGian result;
25        result.TinhLaiGio(totalSeconds);
26        return result;
27    }
28
29    ThoiGian ThoiGian::operator-(int Giay) {
30        int totalSeconds = TinhGiay() - Giay;
31        ThoiGian result;
32        result.TinhLaiGio(totalSeconds);
33        return result;
34    }
```



```
34
35     ThoiGian ThoiGian::operator+(const ThoiGian& a) {
36         int totalSeconds = TinhGiay() + a.TinhGiay();
37         ThoiGian result;
38         result.TinhLaiGio(totalSeconds);
39         return result;
40     }
41
42     ThoiGian ThoiGian::operator-(const ThoiGian& a) {
43         int totalSeconds = TinhGiay() - a.TinhGiay();
44         ThoiGian result;
45         result.TinhLaiGio(totalSeconds);
46         return result;
47     }
48
49     // Increment and decrement operators
50     ThoiGian& ThoiGian::operator++() {
51         *this = *this + 1;
52         return *this;
53     }
54
55     ThoiGian ThoiGian::operator++(int) {
56         ThoiGian temp = *this;
57         ++(*this);
58         return temp;
59     }
60
61     ThoiGian& ThoiGian::operator--() {
62         *this = *this - 1;
63         return *this;
64     }
65
66     ThoiGian ThoiGian::operator--(int) {
67         ThoiGian temp = *this;
68         --(*this);
69         return temp;
70     }
71
```



```
72 // Comparison operators
73 bool ThoiGian::operator==(const ThoiGian& a) const {
74     return TinhGiay() == a.TinhGiay();
75 }
76
77 bool ThoiGian::operator!=(const ThoiGian& a) const {
78     return TinhGiay() != a.TinhGiay();
79 }
80
81 bool ThoiGian::operator>=(const ThoiGian& a) const {
82     return TinhGiay() >= a.TinhGiay();
83 }
84
85 bool ThoiGian::operator<=(const ThoiGian& a) const {
86     return TinhGiay() <= a.TinhGiay();
87 }
88
89 bool ThoiGian::operator>(const ThoiGian& a) const {
90     return TinhGiay() > a.TinhGiay();
91 }
92
93 bool ThoiGian::operator<(const ThoiGian& a) const {
94     return TinhGiay() < a.TinhGiay();
95 }
96
97 // Input and output operators
98 std::istream& operator>>(std::istream& in, ThoiGian& tg) {
99     std::cout << "Nhap gio: ";
100     in >> tg.iGio;
101     std::cout << "Nhap phut: ";
102     in >> tg.iPhut;
103     std::cout << "Nhap giay: ";
104     in >> tg.iGiay;
105     return in;
106 }
107
108 std::ostream& operator<<(std::ostream& out, const ThoiGian& tg) {
109     out << tg.iGio << " gio, " << tg.iPhut << " phut, " << tg.iGiay << "
```



```
110         giay";  
111     return out;  
    }
```

## 3.5 Kiểm thử

- Testcase 1:

```
PS D:\THANH\HK3\OOP\LAB_3\Bai3> g++ main.cpp ThoiGian.cpp -o run  
PS D:\THANH\HK3\OOP\LAB_3\Bai3> ./run  
Nhap thoi gian t1:  
Nhap gio: 1  
Nhap phut: 30  
Nhap giay: 45  
Nhap thoi gian t2:  
Nhap gio: 2  
Nhap phut: 15  
Nhap giay: 30  
Tong t1 + t2: 3 gio, 46 phut, 15 giay  
Hieu t1 - t2: 0 gio, -44 phut, -45 giay  
Nhap so giay them vao t1: 50  
t1 sau khi them giay: 1 gio, 31 phut, 35 giay  
Nhap so giay tru khoi t1: 20  
t1 sau khi tru giay: 1 gio, 30 phut, 25 giay  
t1 == t2: False  
t1 > t2: False
```

- Testcase 2:

```
PS D:\THANH\HK3\OOP\LAB_3\Bai3> g++ main.cpp ThoiGian.cpp -o run  
PS D:\THANH\HK3\OOP\LAB_3\Bai3> ./run  
Nhap thoi gian t1:  
Nhap gio: 0  
Nhap phut: 45  
Nhap giay: 15  
Nhap thoi gian t2:  
Nhap gio: 0  
Nhap phut: 45  
Nhap giay: 15  
Tong t1 + t2: 1 gio, 30 phut, 30 giay  
Hieu t1 - t2: 0 gio, 0 phut, 0 giay  
Nhap so giay them vao t1: 10  
t1 sau khi them giay: 0 gio, 45 phut, 25 giay  
Nhap so giay tru khoi t1: 5  
t1 sau khi tru giay: 0 gio, 45 phut, 10 giay  
t1 == t2: True  
t1 > t2: False
```



- Testcase 3:

```
PS D:\THANH\HK3\OOP\LAB_3\Bai3> g++ main.cpp ThoiGian.cpp -o run
PS D:\THANH\HK3\OOP\LAB_3\Bai3> ./run
Nhap thoi gian t1:
Nhap gio: 2
Nhap phut: 10
Nhap giay: 5
Nhap thoi gian t2:
Nhap gio: 1
Nhap phut: 30
Nhap giay: 10
Tong t1 + t2: 3 gio, 40 phut, 15 giay
Hieu t1 - t2: 0 gio, 39 phut, 55 giay
Nhap so giay them vao t1: 120
t1 sau khi them giay: 2 gio, 12 phut, 5 giay
Nhap so giay tru khoi t1: 30
t1 sau khi tru giay: 2 gio, 9 phut, 35 giay
t1 == t2: False
t1 > t2: True
```

- Testcase 4:

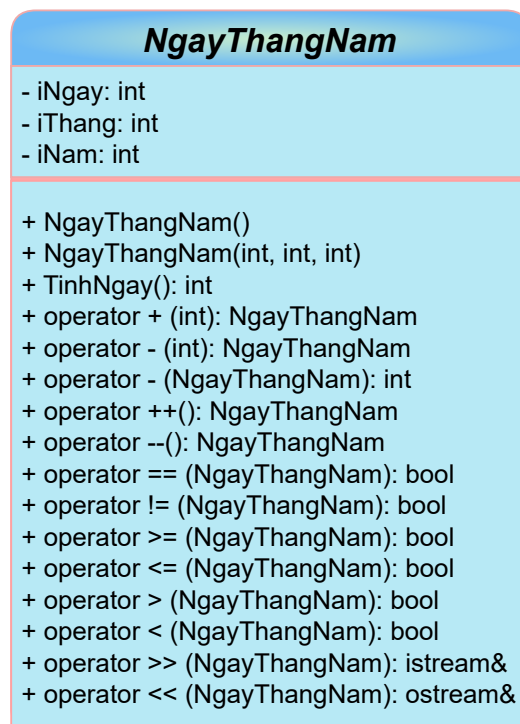
```
PS D:\THANH\HK3\OOP\LAB_3\Bai3> g++ main.cpp ThoiGian.cpp -o run
PS D:\THANH\HK3\OOP\LAB_3\Bai3> ./run
Nhap thoi gian t1:
Nhap gio: 11
Nhap phut: 11
Nhap giay: 11
Nhap thoi gian t2:
Nhap gio: 12
Nhap phut: 12
Nhap giay: 12
Tong t1 + t2: 23 gio, 23 phut, 23 giay
Hieu t1 - t2: -1 gio, -1 phut, -1 giay
Nhap so giay them vao t1: 10
t1 sau khi them giay: 11 gio, 11 phut, 21 giay
Nhap so giay tru khoi t1: 03
t1 sau khi tru giay: 11 gio, 11 phut, 8 giay
t1 == t2: False
t1 > t2: False
```

## 4. Bài tập 4

Xây dựng lớp ngày tháng năm:

- **Thuộc tính:** iNgày, iThang, iNam
- **Phương thức:** NgayThangNam(), NgayThangNam (int Nam, int Thang, intNgày), TinhNgay()
- **Thực hiện các phương thức operator:** +(int ngay), -(int ngay), -(NgayThangNam), ++, --, ==, !=, >=, <=, >, <, >>, <<.

### 4.1 Biểu đồ lớp



Hình 4.4: Sơ đồ khối bài 4

### 4.2 Mô tả đầu vào và đầu ra

- **Đầu vào:** Nhập ngày, tháng, năm cho hai đối tượng date1 và date2 dưới dạng: ngày/tháng/năm.





- **Đầu ra:**

- Xuất ra các thông tin về date1 và date2.
- Kết quả phép toán cộng, trừ ngày.
- Số ngày giữa hai ngày.

## 4.3 Ý tưởng

- **Khởi tạo:** Sử dụng các constructor để khởi tạo đối tượng ngày tháng năm.
- **Tính ngày:** Sử dụng phương thức TinhNgay() để tính tổng số ngày từ đầu năm đến ngày đã cho.
- **Phép toán:** Nạp chồng toán tử cho phép thực hiện các phép toán cộng, trừ, so sánh giữa các đối tượng NgayThangNam.
- **Nhập/ Xuất** Sử dụng các phương thức nhập/xuất để tương tác với người dùng.

## 4.4 Thực thi code

Link code: [Bài 4](#)

- **NgayThangNam.h**

```
1  #ifndef NGAYTHANGNAM_H
2  #define NGAYTHANGNAM_H
3
4  #include <bits/stdc++.h>
5
6  class NgayThangNam {
7  private:
8      int iNgay;
9      int iThang;
10     int iNam;
11
12
13     bool isLeapYear(int year) const;
14
15 }
```



```
16     int getDaysInMonth(int month, int year) const;
17
18
19     bool isValidDate(int ngay, int thang, int nam) const;
20
21 public:
22
23     NgayThangNam();
24
25     NgayThangNam(int Nam, int Thang, int Ngay);
26
27     int TinhNgay() const;
28
29     void TinhLaiNgay(int soNgay);
30
31     NgayThangNam operator+(int ngay) const;
32
33     NgayThangNam operator-(int ngay) const;
34
35     int operator-(const NgayThangNam& other) const;
36
37     NgayThangNam& operator++();
38     NgayThangNam operator++(int);
39
40
41     NgayThangNam& operator--();
42     NgayThangNam operator--(int);
43
44
45     bool operator==(const NgayThangNam& other) const;
46
47     bool operator!=(const NgayThangNam& other) const;
48
49     bool operator>=(const NgayThangNam& other) const;
50
51     bool operator<=(const NgayThangNam& other) const;
52
53     bool operator>(const NgayThangNam& other) const;
```



```

54
55     bool operator<(const NgayThangNam& other) const;
56
57     friend std::istream& operator>>(std::istream& is, NgayThangNam& date);
58
59     friend std::ostream& operator<<(std::ostream& os, const NgayThangNam&
60         date);
61 };
62 #endif

```

### • ThoiGian.cpp

```

1     #include "NgayThangNam.h"
2
3
4     NgayThangNam::NgayThangNam() : iNgay(1), iThang(1), iNam(0) {}
5
6
7     NgayThangNam::NgayThangNam(int Nam, int Thang, int Ngay) : iNgay(Ngay),
8         iThang(Thang), iNam(Nam) {}
9
10    bool NgayThangNam::isLeapYear(int year) const {
11        return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
12    }
13
14
15    int NgayThangNam::getDaysInMonth(int month, int year) const {
16        switch (month) {
17            case 4: case 6: case 9: case 11: return 30;
18            case 2: return isLeapYear(year) ? 29 : 28;
19            default: return 31;
20        }
21    }
22
23    int NgayThangNam::TinhNgay() const {
24        int totalDays = iNgay;

```



```

25     for (int y = 0; y < iNam; ++y) totalDays += isLeapYear(y) ? 366 : 365;
26     for (int m = 1; m < iThang; ++m) totalDays += getDaysInMonth(m, iNam);
27     return totalDays;
28 }
29
30 void NgayThangNam::TinhLaiNgay(int soNgay) {
31     iNam = 0;
32     while (soNgay >= (isLeapYear(iNam) ? 366 : 365)) soNgay -=
33         isLeapYear(iNam++) ? 366 : 365;
34     iThang = 1;
35     while (soNgay >= getDaysInMonth(iThang, iNam)) soNgay -=
36         getDaysInMonth(iThang++, iNam);
37     iNgay = soNgay + 1;
38 }
39
40 NgayThangNam NgayThangNam::operator+(int ngay) const {
41     NgayThangNam result = *this;
42     result.TinhLaiNgay(TinhNgay() + ngay);
43     return result;
44 }
45
46 NgayThangNam NgayThangNam::operator-(int ngay) const {
47     NgayThangNam result = *this;
48     result.TinhLaiNgay(TinhNgay() - ngay);
49     return result;
50 }
51
52
53 int NgayThangNam::operator-(const NgayThangNam& other) const {
54     return TinhNgay() - other.TinhNgay();
55 }
56
57
58 NgayThangNam& NgayThangNam::operator++() {
59     *this = *this + 1;
60     return *this;

```



```
61     }
62
63     NgayThangNam NgayThangNam::operator++(int) {
64         NgayThangNam temp = *this;
65         ++(*this);
66         return temp;
67     }
68
69
70     NgayThangNam& NgayThangNam::operator--() {
71         *this = *this - 1;
72         return *this;
73     }
74
75
76     NgayThangNam NgayThangNam::operator--(int) {
77         NgayThangNam temp = *this;
78         --(*this);
79         return temp;
80     }
81
82
83     bool NgayThangNam::operator==(const NgayThangNam& other) const {
84         return iNgay == other.iNgay && iThang == other.iThang && iNam ==
            other.iNam;
85     }
86
87     bool NgayThangNam::operator!=(const NgayThangNam& other) const {
88         return !(*this == other);
89     }
90
91
92     bool NgayThangNam::operator>=(const NgayThangNam& other) const { return
        TinhNgay() >= other.TinhNgay(); }
93
94     bool NgayThangNam::operator<=(const NgayThangNam& other) const { return
        TinhNgay() <= other.TinhNgay(); }
95
```



```

96     bool NgayThangNam::operator>(const NgayThangNam& other) const { return
        TinhNgay() > other.TinhNgay(); }
97
98     bool NgayThangNam::operator<(const NgayThangNam& other) const { return
        TinhNgay() < other.TinhNgay(); }
99
100
101     std::istream& operator>>(std::istream& is, NgayThangNam& date) {
102         std::cout << "Nhap ngay: ";
103         is >> date.iNgay;
104         std::cout << "Nhap thang: ";
105         is >> date.iThang;
106         std::cout << "Nhap nam: ";
107         is >> date.iNam;
108         return is;
109     }
110
111     std::ostream& operator<<(std::ostream& os, const NgayThangNam& date) {
112         os << date.iNgay << "/" << date.iThang << "/" << date.iNam;
113         return os;
114     }

```

## 4.5 Kiểm thử

- Testcase 1:

```

PS D:\THANH\HK3\OOP\LAB_3\Bai4> g++ main.cpp NgayThangNam.cpp -o run
PS D:\THANH\HK3\OOP\LAB_3\Bai4> ./run
Nhap ngay thang nam date1:
Nhap ngay: 25
Nhap thang: 2
Nhap nam: 2024
Nhap ngay thang nam date2:
Nhap ngay: 25
Nhap thang: 3
Nhap nam: 2024
Ngày thu nhất: 25/2/2024
Ngày thu hai: 25/3/2024
Ngày thu nhất sau khi cộng 15 ngày: 12/3/2024
Ngày thu hai sau khi trừ 20 ngày: 6/3/2024
Khoảng cách giữa hai ngày là: 29 ngày

```



- Testcase 2:

```
PS D:\THANH\HK3\OOP\LAB_3\Bai4> ./run
Nhap ngay thang nam date1:
Nhap ngay: 17
Nhap thang: 9
Nhap nam: 2005
Nhap ngay thang nam date2:
Nhap ngay: 11
Nhap thang: 9
Nhap nam: 2009
Ngày thu nhất: 17/9/2005
Ngày thu hai: 11/9/2009
Ngày thu nhất sau khi cộng 15 ngày: 3/10/2005
Ngày thu hai sau khi trừ 20 ngày: 23/8/2009
Khoảng cách giữa hai ngày là: 1455 ngày
```

- Testcase 3:

```
PS D:\THANH\HK3\OOP\LAB_3\Bai4> ./run
Nhap ngay thang nam date1:
Nhap ngay: 12
Nhap thang: 9
Nhap nam: 1973
Nhap ngay thang nam date2:
Nhap ngay: 12
Nhap thang: 9
Nhap nam: 2024
Ngày thu nhất: 12/9/1973
Ngày thu hai: 12/9/2024
Ngày thu nhất sau khi cộng 15 ngày: 28/9/1973
Ngày thu hai sau khi trừ 20 ngày: 24/8/2024
Khoảng cách giữa hai ngày là: 18628 ngày
```