

Chương 8

TOÁN TỬ SO SÁNH

- Nguyễn Hữu Lợi
- Đoàn Chánh Thống
- ThS. Nguyễn Thành Hiệp
- ThS. Trương Quốc Dũng
- ThS. Võ Duy Nguyên
- ThS. Nguyễn Văn Toàn
- TS. Nguyễn Duy Khánh
- TS. Nguyễn Tấn Trần Minh Khang

1. MỤC TIÊU

Relational
Operators
in C++



1. Mục tiêu

- Hiểu được phương thức toán tử so sánh (relational operators, comparison operators) là gì?
- Cài đặt toán tử so sánh cho các lớp đối tượng.



2. KHÁI NIỆM

Relational
Operators
in C++



1. Mục tiêu

- Toán tử so sánh (relational and comparison operators) trong ngôn ngữ lập trình C được sử dụng để so sánh giá trị của biến (variable) này với giá trị của biến khác.
- Mở rộng cho phương pháp lập trình hướng đối tượng (object-oriented programming) với C++ ta có thể nói toán tử so sánh được sử dụng để so sánh đối tượng (object) này với đối tượng khác.
- Hiển nhiên việc so sánh hai đối tượng (object) phải được thực hiện theo một tiêu chí nào đó.



3. CÁC TOÁN TỬ SO SÁNH CƠ BẢN

3. Các toán tử so sánh cơ bản

- Toán tử so sánh lớn hơn – **greater than** (operator $>$)
- Toán tử so sánh nhỏ hơn – **less than** (operator $<$)
- Toán tử so sánh bằng – **equal to** (operator $==$)
- Toán tử so sánh khác – **not equal to** (operator $!=$)
- Toán tử so sánh lớn hơn bằng – **greater than or equal to** (operator $>=$)
- Toán tử so sánh nhỏ hơn bằng – **less than or equal to** (operator $<=$)

4. ĐẶT VẤN ĐỀ

Relational
Operators
in C++



4. Đặt vấn đề

— Hãy định nghĩa các phương thức cần thiết cho lớp đối tượng `CHocSinh` để các câu lệnh sau có thể thực thi được.

— Đoạn chương trình

1. `CHocSinh a,b;`

2. `cin >> a >> b;`

3. `if(a > b)`

4. `cout << "DiemTrungBinh a lon hon b";`

5. `else`

6. `cout << "DiemTrungBinh a ko lon hon b";`

5. GIẢI QUYẾT VẤN ĐỀ

Relational
Operators
in C++



5. Giải quyết vấn đề

— Đoạn chương trình

1. CHocSinh a,b;

2. cin >> a >> b;

3. if(a > b)

4. cout << "DTB a lon hon DTB b";

5. else

6. cout << "DiemTrungBinh a ko lon hon b";

— Để giải quyết vấn đề trên ta phải khai báo và định nghĩa phương thức toán tử so sánh lớn hơn cho lớp đối tượng CHocSinh.

5. Giải quyết vấn đề

```
11.class CHocSinh
12.{
13.    private:
14.        string HoTen;
15.        int Toan;
16.        int Van;
17.        float DiemTrungBinh;
18.    public:
19.        int operator>(const CHocSinh&);
20.        ...
21.};
```

5. Giải quyết vấn đề

```
11.class CHocSinh
12.{
13.    private:
14.        string HoTen;
15.        int Toan;
16.        int Van;
17.        float DiemTrungBinh;
18.    public:
19.        int operator>(const CHocSinh&);
20.        int operator<(const CHocSinh&);
21.        int operator>=(const CHocSinh&);
22.        int operator<=(const CHocSinh&);
23.        int operator==(const CHocSinh&);
24.        int operator!=(const CHocSinh&);
25.};
```

Relational
Operators
in C++



5. Giải quyết vấn đề

— Định nghĩa các phương thức

```
11.int CHocSinh::operator > (const CHocSinh &x)
12.{
13.    if(DiemTrungBinh > x.DiemTrungBinh)
14.        return 1;
15.    return 0;
16.}
```



5. Giải quyết vấn đề

— Định nghĩa các phương thức

```
11.int CHocSinh::operator < (const CHocSinh &x)
12.{
13.    if(DiemTrungBinh < x.DiemTrungBinh)
14.        return 1;
15.    return 0;
16.}
```



5. Giải quyết vấn đề

— Định nghĩa các phương thức

```
11.int CHocSinh::operator >= (const CHocSinh &x)
12.{
13.    if(DiemTrungBinh >= x.DiemTrungBinh)
14.        return 1;
15.    return 0;
16.}
```



5. Giải quyết vấn đề

— Định nghĩa các phương thức

```
11.int CHocSinh::operator <= (const CHocSinh &x)
12.{
13.    if(DiemTrungBinh <= x.DiemTrungBinh)
14.        return 1;
15.    return 0;
16.}
```



5. Giải quyết vấn đề

— Định nghĩa các phương thức

```
11.int CHocSinh::operator == (const CHocSinh &x)
12.{
13.    if(DiemTrungBinh == x.DiemTrungBinh)
14.        return 1;
15.    return 0;
16.}
```



5. Giải quyết vấn đề

— Định nghĩa các phương thức

```
11.int CHocSinh::operator != (const CHocSinh &x)
12.{
13.    if(DiemTrungBinh != x.DiemTrungBinh)
14.        return 1;
15.    return 0;
16.}
```



6. HƯỚNG DẪN SỬ DỤNG

Relational
Operators
in C++



6. Hướng dẫn sử dụng

— Xem xét đoạn chương trình

```
1. CHocSinh a,b;  
2. cin >> a >> b;  
3. if(a > b)  
4.     cout << "Diem Trung Binh a lon hon Diem Trung Binh b";  
5. else  
6.     cout << "Diem Trung Binh a ko lon hon b";
```

— Trong câu lệnh 3 của đoạn chương trình trên ta nói: đối tượng **a** gọi thực hiện phương thức toán tử so sánh lớn hơn với đối số là đối tượng **b**.

6. Hướng dẫn sử dụng

```
1. CHocSinh a,b;  
2. cin >> a >> b;  
3. if(a > b)  
4.     cout << "Diem Trung Binh a lon hon Diem Trung Binh b";  
5. else  
6.     cout << "Diem Trung Binh a ko lon hon b";
```

— Định nghĩa phương thức

```
1. int CHocSinh::operator > (CHocSinh &x)  
2. {  
3.     if(DiemTrungBinh > x.DiemTrungBinh)  
4.         return 1;  
5.     return 0;  
6. }
```

7. ỨNG DỤNG

Relational
Operators
in C++



7. Ứng dụng

— Bài toán: Hãy định nghĩa các toán tử so sánh nhỏ hơn và toán tử so sánh lớn hơn cho lớp đối tượng **CPhanSo**.

```
11.class CPhanSo
12.{
13.    private:
14.        int Tu;
15.        int Mau;
16.    public:
17.        CPhanSo operator-(CPhanSo);
18.        int operator > (const CPhanSo&);
19.        int operator < (const CPhanSo&);
20.};
```



7. Ứng dụng

```
11.CPhanSo CPhanSo::operator - (CPhanSo x)
12.{
13.    CPhanSo temp;
14.    temp.Tu = Tu * x.Mau - Mau * x.Tu;
15.    temp.Mau = Mau * x.Mau;
16.    return temp;
17.}
```



7. Ứng dụng

```
11.int CPhanSo::operator > (const CPhanSo &x)
12.{
13.    CPhanSo temp = *this - x;
14.    if(temp.Tu * temp.Mau > 0)
15.        return 1;
16.    else
17.        return 0;
18.}
```



7. Ứng dụng

```
11.int CPhanSo::operator < (const CPhanSo &x)
12.{
13.    CPhanSo temp = *this - x;
14.    if(temp.Tu * temp.Mau < 0)
15.        return 1;
16.    else
17.        return 0;
18.}
```



7. Ứng dụng

```
11.int CPhanSo::operator > (const CPhanSo &x)
12.{
13.|   CPhanSo temp = *this - x;
14.|   return (temp.Tu * temp.Mau > 0);
15.}
```



7. Ứng dụng

```
1. CPhanSo a,b;  
2. cin >> a >> b;  
3. if(a>b)  
4.     cout << "Diem Trung Binh a lon hon Diem Trung Binh b";  
5. else  
6.     cout << "Diem Trung Binh a ko lon hon b";
```

— Định nghĩa phương thức

```
11.int CPhanSo::operator > (const CPhanSo &x)  
12.{  
13.    CPhanSo temp = *this - x;  
14.    return (temp.Tu * temp.Mau >0);  
15.}
```

8. BÀI TẬP

Relational
Operators
in C++



8. Bài tập

- Bài 01: Hãy định nghĩa tất cả các phương thức toán tử so sánh cho lớp đối tượng **CPhanSo**.
- Bài 02: Hãy định nghĩa tất cả các phương thức toán tử so sánh cho lớp đối tượng **CHonSo**.
- Bài 03: Hãy định nghĩa tất cả các phương thức toán tử so sánh cho lớp đối tượng **CDiem** trong mặt phẳng *Oxy*. Biết rằng tiêu chuẩn so sánh 2 điểm là so sánh theo khoảng cách tới gốc tọa độ. Điểm nào ở xa gốc hơn thì lớn hơn.

8. Bài tập

— Bài 04: Hãy định nghĩa toán tử so sánh bằng và toán tử so sánh khác cho tất cả các đối tượng sau:

1. Lớp điểm (CDiem)
2. Lớp ngày (CNgay)
3. Lớp thời gian (CThoiGian)
4. Lớp đơn thức (CDonThuc)

5. Lớp điểm không gian (CDiemKhongGian)

6. Lớp đường thẳng (CDuongThang)

7. Lớp lớp tam giác (CTamGiac)

8. Lớp hình cầu (CHinhCau)

8. Bài tập

- Bài 05: Hãy định nghĩa tất cả các phương thức toán tử so sánh cho lớp số phức **CSoPhuc** trong toán học. **Biết rằng tiêu chuẩn so sánh 2 số phức là so sánh theo mô đun.** Mô đun của số phức nào lớn hơn thì số phức đó lớn hơn.
- **Nhắc lại:**
 - + Cho số phức $z = a + ib$.
 - + **Mô đun của số phức z là: $|z| = \sqrt{a^2 + b^2}$**

8. Bài tập

- Bài 06: Hãy định nghĩa tất cả các phương thức toán tử so sánh cho lớp đường tròn **CDuongTron** trong toán học. **Biết rằng tiêu chuẩn so sánh 2 đường tròn là so sánh theo diện tích. Diện tích của đường tròn nào lớn hơn thì đường tròn đó lớn hơn.**

Cảm ơn quý vị đã lắng nghe

Nhóm tác giả

TS. Nguyễn Tấn Trần Minh Khang

Chương 8

TOÁN TỬ SO SÁNH – PYTHON

- Nguyễn Hữu Lợi
- Đoàn Chánh Thống
- ThS. Nguyễn Thành Hiệp
- ThS. Trương Quốc Dũng
- ThS. Võ Duy Nguyên
- ThS. Nguyễn Văn Toàn
- TS. Nguyễn Duy Khánh
- TS. Nguyễn Tấn Trần Minh Khang

1. CÁC TOÁN TỬ SO SÁNH CƠ BẢN PYTHON

Các toán tử so sánh cơ bản Python

- Toán tử so sánh lớn hơn – greater than (`>`)
- Toán tử so sánh nhỏ hơn – less than (`<`)
- Toán tử so sánh bằng – equal to (`==`)
- Toán tử so sánh khác – not equal to (`!=`)
- Toán tử so sánh lớn hơn bằng – greater than or equal to (`>=`)
- Toán tử so sánh nhỏ hơn bằng – less than or equal to (`<=`)

2. ĐẶT VẤN ĐỀ

Đặt vấn đề

— Hãy định nghĩa các phương thức cần thiết cho lớp đối tượng `CHocSinh` để các câu lệnh sau có thể thực thi được.

— **Đoạn chương trình**

1. `a = CHocSinh()`

2. `b = CHocSinh()`

3. `a.Nhap()`

4. `b.Nhap()`

5. `a.XuLy()`

6. `b.XuLy()`

7. ...

Đặt vấn đề

— Hãy định nghĩa các phương thức cần thiết cho lớp đối tượng `CHocSinh` để các câu lệnh sau có thể thực thi được.

— Đoạn chương trình

1. ...

2. `if (a > b):`

3. `print("DTB a lon hon DTB b")`

4. `else:`

5. `print("DTB a ko lon hon DTB b")`

3. GIẢI QUYẾT VẤN ĐỀ

Giải quyết vấn đề

— Đoạn chương trình

1. ...
 2. `if (a > b):`
 3. `print("DTB a lon hon DTB b")`
 4. `else:`
 5. `print("DTB a ko lon hon DTB b")`
- Để giải quyết vấn đề trên ta phải khai báo và định nghĩa phương thức toán tử so sánh lớn hơn cho lớp đối tượng **CHocSinh**.

Giải quyết vấn đề

```
1. class CHocSinh:
2.     def __init__(self):
3.         self.HoTen = ""
4.         self.Toan = 0
5.         self.Van = 0
6.         self.DiemTrungBinh = 0
7.         ...
```


Giải quyết vấn đề

```
1. class CHocSinh:
2.     ...
3.     def Nhap(self):
4.         self.HoTen = input("Nhap ho ten: ")
5.         self.Toan = int(input("Nhap toan: "))
6.         self.Van = int(input("Nhap van: "))
7.         ...
```

Giải quyết vấn đề

```
1. class CHocSinh:
2.     ...
3.     def Xuat(self):
4.         print("Ho ten:", self.HoTen)
5.         print("Diem Toan:", self.Toan)
6.         print("Diem Van:", self.Van)
7.         print("Diem trung binh:", self.DiemTrungBinh)
8.     ...
```

Giải quyết vấn đề

```
1. class CHocSinh:
2.     ...
3.     def XuLy(self):
4.         self.DiemTrungBinh = (self.Toan+self.Van)/2
5.     ...
```

Giải quyết vấn đề

```
1. class CHocSinh:
2.     ...
3.     def __gt__(self, x):
4.         if (self.DiemTrungBinh > x.DiemTrungBinh):
5.             return True
6.         return False
7.     ...
```

Giải quyết vấn đề

```
1. class CHocSinh:
2.     ...
3.     def __lt__(self, x):
4.         if (self.DiemTrungBinh < x.DiemTrungBinh):
5.             return True
6.         return False
7.     ...
```

Giải quyết vấn đề

```
1. class CHocSinh:
2.     ...
3.     def __ge__(self, x):
4.         if (self.DiemTrungBinh >= x.DiemTrungBinh):
5.             return True
6.         return False
7.     ...
```


Giải quyết vấn đề

```
1. class CHocSinh:
2.     ...
3.     def __le__(self, x):
4.         if (self.DiemTrungBinh <= x.DiemTrungBinh):
5.             return True
6.         return False
7.     ...
```

Giải quyết vấn đề

```
1. class CHocSinh:
2.     ...
3.     def __eq__(self, x):
4.         if (self.DiemTrungBinh == x.DiemTrungBinh):
5.             return True
6.         return False
7.     ...
```

Giải quyết vấn đề

```
1. class CHocSinh:
2.     ...
3.     def __ne__(self, x):
4.         if (self.DiemTrungBinh != x.DiemTrungBinh):
5.             return True
6.         return False
```

5. HƯỚNG DẪN SỬ DỤNG

Giải quyết vấn đề

— Đoạn chương trình

1. ...

2. `if (a > b):`

3. `print("DTB a lon hon DTB b")`

4. `else:`

5. `print("DTB a ko lon hon DTB b")`

— Trong câu lệnh 2 của đoạn chương trình trên ta nói: đối tượng a gọi thực hiện phương thức toán tử so sánh lớn hơn với đối số là đối tượng b.

6. ỨNG DỤNG

Ứng dụng

—Bài toán: Hãy định nghĩa các toán tử so sánh nhỏ hơn và toán tử so sánh lớn hơn cho lớp đối tượng CPhanSo.

Ứng dụng

```
1. class CPhanSo:
2.     def __init__(self):
3.         self.Tu = 0
4.         self.Mau = 1
5.     ...
```

Ứng dụng

```
1. class CPhanSo:
2.     ...
5.     def Nhap(self):
6.         self.Tu = int(input("Nhap tu: "))
7.         self.Mau = int(input("Nhap mau: "))
8.     ...
```

Ứng dụng

```
1. class CPhanSo:
2.     ...
8.     def Xuat(self):
9.         print(self.Tu, "/", self.Mau)
10.    ...
```

Ứng dụng

```
1. class CPhanSo:
2.     ...
3.     def __lt__(self, x):
4.         a = self.Tu / self.Mau
5.         b = x.Tu / x.Mau
6.         return a < b
8.     ...
```

Ứng dụng

```
1. class CPhanSo:
2.     ...
3.     def __gt__(self, x):
4.         a = self.Tu / self.Mau
5.         b = x.Tu / x.Mau
6.         return a > b
8.     ...
```

7. BÀI TẬP

8. Bài tập

- Bài 01: Hãy định nghĩa tất cả các phương thức toán tử so sánh cho lớp đối tượng **CPhanSo**.
- Bài 02: Hãy định nghĩa tất cả các phương thức toán tử so sánh cho lớp đối tượng **CHonSo**.
- Bài 03: Hãy định nghĩa tất cả các phương thức toán tử so sánh cho lớp đối tượng **CDiem** trong mặt phẳng *Oxy*. Biết rằng tiêu chuẩn so sánh 2 điểm là so sánh theo khoảng cách tới gốc tọa độ. Điểm nào ở xa gốc hơn thì lớn hơn.

8. Bài tập

— Bài 04: Hãy định nghĩa toán tử so sánh bằng và toán tử so sánh khác cho tất cả các đối tượng sau:

1. Lớp điểm (CDiem)
2. Lớp ngày (CNgay)
3. Lớp thời gian (CThoiGian)
4. Lớp đơn thức (CDonThuc)

5. Lớp điểm không gian (CDiemKhongGian)

6. Lớp đường thẳng (CDuongThang)

7. Lớp lớp tam giác (CTamGiac)

8. Lớp hình cầu (CHinhCau)

8. Bài tập

- Bài 05: Hãy định nghĩa tất cả các phương thức toán tử so sánh cho lớp số phức **CSoPhuc** trong toán học. **Biết rằng tiêu chuẩn so sánh 2 số phức là so sánh theo mô đun.** Mô đun của số phức nào lớn hơn thì số phức đó lớn hơn.
- **Nhắc lại:**
 - + Cho số phức $z = a + ib$.
 - + **Mô đun của số phức z là: $|z| = \sqrt{a^2 + b^2}$**

8. Bài tập

- Bài 06: Hãy định nghĩa tất cả các phương thức toán tử so sánh cho lớp đường tròn **CDuongTron** trong toán học. **Biết rằng tiêu chuẩn so sánh 2 đường tròn là so sánh theo diện tích. Diện tích của đường tròn nào lớn hơn thì đường tròn đó lớn hơn.**

Cảm ơn quý vị đã lắng nghe

Nhóm tác giả

TS. Nguyễn Tấn Trần Minh Khang