

Chương 8

TOÁN TỬ SO SÁNH

0. MỤC TIÊU

- Hiểu được phương thức toán tử so sánh là gì?
- Cài đặt toán tử so sánh cho các lớp đối tượng.

1. KHÁI NIỆM

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

2. CÁC TOÁN TỬ SO SÁNH CƠ BẢN

- Toán tử so sánh lớn hơn (**operator >**)
- Toán tử so sánh nhỏ hơn (**operator <**)
- Toán tử so sánh lớn hơn bằng (**operator >=**)
- Toán tử so sánh nhỏ hơn bằng (**operator <=**)
- Toán tử so sánh bằng (**operator ==**)
- Toán tử so sánh khác (**operator !=**)

3. 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<<"DTB a lon hon DTB b";`
 5. `else`
 6. `cout<<"DTB a ko lon hon b";`

4. 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<<"DTB 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.

4. GIẢI QUYẾT VẤN ĐỀ

– Khai báo lớp

```
11. class CHocSinh
12. {
13.     private:
14.         char hoten[31];
15.         int toan;
16.         int van;
17.         float dtb;
18.     public:
19.         int operator> (CHocSinh);
20.         int operator< (CHocSinh);
21.         int operator>= (CHocSinh);
22.         int operator<= (CHocSinh);
23.         int operator== (CHocSinh);
24.         int operator!= (CHocSinh);
25. } ;
```

4. GIẢI QUYẾT VẤN ĐỀ

– Định nghĩa các phương thức

```
11. int CHocSinh::operator >
                                (CHocSinh x)
12. {
13.     if (dtb > x.dtb)
14.         return 1;
15.     return 0;
16. }
17. int CHocSinh::operator <
                                (CHocSinh x)
18. {
19.     if (dtb < x.dtb)
20.         return 1;
21.     return 0;
22. }
```


4. GIẢI QUYẾT VẤN ĐỀ

– Định nghĩa các phương thức

```
11. int CHocSinh::operator >=
                                (CHocSinh x)
12. {
13.     if (dtb >= x.dtb)
14.         return 1;
15.     return 0;
16. }
17. int CHocSinh::operator <=
                                (CHocSinh x)
18. {
19.     if (dtb <= x.dtb)
20.         return 1;
21.     return 0;
22. }
```

4. GIẢI QUYẾT VẤN ĐỀ

– Định nghĩa các phương thức

```
11. int CHocSinh::operator ==  
                                (CHocSinh x)  
12. {  
13.     if (dtb==x.dtb)  
14.         return 1;  
15.     return 0;  
16. }  
17. int CHocSinh::operator !=  
                                (CHocSinh x)  
18. {  
19.     if (dtb!=x.dtb)  
20.         return 1;  
21.     return 0;  
22. }
```

4. GIẢI QUYẾT VẤN ĐỀ

- Định nghĩa các phương thức

```
11. int CHocSinh::operator >  
                                (CHocSinh x)  
12. {  
13. |     return (dtb>x.dtb);  
14. }
```

5. ỨNG DỤNG

- Bài toán: Hãy định nghĩa các toán tử so sánh 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 > (CPhanSo) ;
19.         int operator < (CPhanSo) ;
20.         int operator >= (CPhanSo) ;
21.         int operator <= (CPhanSo) ;
22.         int operator == (CPhanSo) ;
23.         int operator != (CPhanSo) ;
24. } ;
```

5. ỨNG DỤNG

```
1. CPhanSo CPhanSo::operator-  
    (CPhanSo x)  
2. {  
3.     CPhanSo temp;  
4.     temp.tu=tu*x.mau-mau*x.tu;  
5.     temp.mau=mau*x.mau;  
6.     return temp;  
7. }
```

5. ỨNG DỤNG

– Định nghĩa các toán tử so sánh

```
1. int CPhanSo::operator>
                                   (CPhanSo x)
2. {
3.     CPhanSo temp = *this - x;
4.     if(temp.tu*temp.mau>0)
5.         return 1;
6.     else
7.         return 0;
8. }
```

5. ỨNG DỤNG

– Định nghĩa các toán tử so sánh

```
1. int CPhanSo::operator<
                                (CPhanSo x)
2. {
3.     CPhanSo temp = *this - x;
4.     if(temp.tu*temp.mau<0)
5.         return 1;
6.     else
7.         return 0;
8. }
```

5. ỨNG DỤNG

– Định nghĩa các toán tử so sánh

```
1. int CPhanSo::operator<=
                                (CPhanSo x)
2. {
3.     CPhanSo temp = *this - x;
4.     if(temp.tu*temp.mau<=0)
5.         return 1;
6.     else
7.         return 0;
8. }
```


5. ỨNG DỤNG

– Định nghĩa các toán tử so sánh

```
1. int CPhanSo::operator>=  
                                (CPhanSo x)  
  
2. {  
3.     CPhanSo temp = *this - x;  
4.     if(temp.tu*temp.mau>=0)  
5.         return 1;  
6.     else  
7.         return 0;  
8. }
```

5. ỨNG DỤNG

– Định nghĩa các toán tử so sánh

```
1. int CPhanSo::operator==  
                                (CPhanSo x)  
  
2. {  
3.     CPhanSo temp = *this - x;  
4.     if(temp.tu*temp.mau==0)  
5.         return 1;  
6.     else  
7.         return 0;  
8. }
```

5. ỨNG DỤNG

– Định nghĩa các toán tử so sánh

```
1. int CPhanSo::operator!=  
                                (CPhanSo x)  
  
2. {  
3.     CPhanSo temp = *this - x;  
4.     if(temp.tu*temp.mau!=0)  
5.         return 1;  
6.     else  
7.         return 0;  
8. }
```

6. 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 toạ độ. Điểm nào ở xa gốc hơn thì lớn hơn.**

6. 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:

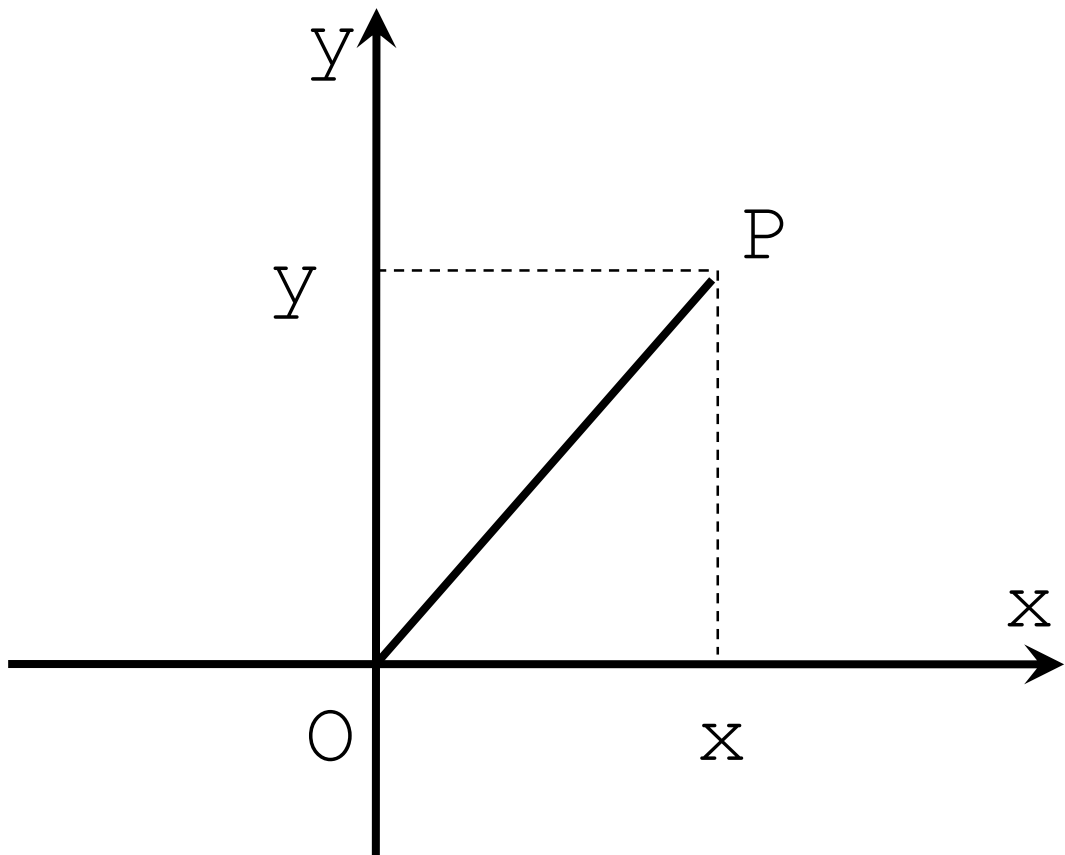
1. Lớp điểm (CDiem)
2. Lớp ngày (CNgay)
3. Lớp thời gian (CThoiGian)
4. Lớp đơn thức (CDonThuc)
5. Lớp điểm không gian (CDiemKhongGian)
6. Lớp đường thẳng (CDuongThang)
7. Lớp số phức (CSophuc)
8. Lớp đường tròn (CDuongTron)
9. Lớp lớp tam giác (CTamGiac)
10. Lớp hình cầu (CHinhCau)

GIẢI BÀI TẬP 03

```
11. class CDiem
12. {
13.     private:
14.         float x;
15.         float y;
16.     public:
17.         float KhoangCachGoc();
18.         int operator>(CDiem);
19.         int operator<(CDiem);
20.         int operator>=(CDiem);
21.         int operator<=(CDiem);
22.         int operator==(CDiem);
23.         int operator!=(CDiem);
24. };
```

GIẢI BÀI TẬP 03

```
11. float CDiem::KhoangCachGoc()  
12. {  
13. |     return sqrt(x*x + y*y);  
14. }
```



GIẢI BÀI TẬP 03

```
11. int CDiem::operator>(CDiem P)
12. {
13.     float a =KhoangCachGoc();
14.     float b =P.KhoangCachGoc();
15.     if(a>b)
16.         return 1;
17.     return 0;
18. }
```

1. CDiem A,B;
2. cin>>A>>B;
3. if (A>B)
4. cout<<"...";
5. else
6. cout<<"...";

GIẢI BÀI TẬP 03

```
11. int CDiem::operator<(CDiem P)
12. {
13.     float a =KhoangCachGoc();
14.     float b =P.KhoangCachGoc();
15.     if(a<b)
16.         return 1;
17.     return 0;
18. }
```

GIẢI BÀI TẬP 03

```
11. int CDiem::operator>=(CDiem P)
12. {
13.     float a =KhoangCachGoc();
14.     float b =P.KhoangCachGoc();
15.     if(a>=b)
16.         return 1;
17.     return 0;
18. }
```

GIẢI BÀI TẬP 03

```
11. int CDiem::operator<=(CDiem P)
12. {
13.     float a =KhoangCachGoc();
14.     float b =P.KhoangCachGoc();
15.     if(a<=b)
16.         return 1;
17.     return 0;
18. }
```

GIẢI BÀI TẬP 03

```
11. int CDiem::operator==(CDiem P)
12. {
13.     float a =KhoangCachGoc();
14.     float b =P.KhoangCachGoc();
15.     if(a==b)
16.         return 1;
17.     return 0;
18. }
```

GIẢI BÀI TẬP 03

```
11. int CDiem::operator!=(CDiem P)
12. {
13.     float a =KhoangCachGoc();
14.     float b =P.KhoangCachGoc();
15.     if(a!=b)
16.         return 1;
17.     return 0;
18. }
```