Chương 8
TOÁN TỬ SO SÁNH

GV. Nguyễn Sơn Hoàng Quốc
ThS. Nguyễn Tấn Trần Minh Khang

BÀI TẬP VỀ NHÀ

- 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**.

BÀI TẬP VỀ NHÀ

- 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)

LỚP PHÂN SỐ

- Bài 1 : Hãy định nghĩa tất cả các phương thức toán tử so sánh cho lớp đối tượng CPhanSo.

LỚP PHÂN SỐ

- Khai báo lớp

```
1.  class CPhanSo
2.  {
3.    private:
4.      int tu;
5.      int mau;
6.    public:
7.      CPhanSo operator-(CPhanSo);
8.      int operator > (CPhanSo);
9.      int operator < (CPhanSo);
10.     int operator >= (CPhanSo);
11.     int operator <= (CPhanSo);
12.     int operator == (CPhanSo);
13.     int operator != (CPhanSo);
14. };
```

LỚP PHÂN SỐ

```
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. }
```

LỚP PHÂN SỐ

- 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. }
```

LỚP PHÂN SỐ

- 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. }
```

LỚP PHÂN SỐ

- Định nghĩa các toán tử so sánh

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

LỚP PHÂN SỐ

- Định nghĩa các toán tử so sánh

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

LỚP PHÂN SỐ

- 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. }
```

LỚP PHÂN SỐ

- Định nghĩa các toán tử so sánh

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

LỚP HỖN SỐ

- **Bài 2 : Hãy định nghĩa tất cả các phương thức toán tử so sánh cho lớp đối tượng CHonSo.**

GV. Nguyễn Sơn Hoàng Quốc
ThS. Nguyễn Tấn Trần Minh Khang

LỚP HỖN SỐ

- Khai báo lớp

```
1.  class CHonSo
2.  {
3.    private:
4.      int nguyen;
5.      int tu;
6.      int mau;
7.    public:
8.      CHonSo operator-(CHonSo);
9.      int operator > (CHonSo);
10.     int operator < (CHonSo);
11.     int operator >= (CHonSo);
12.     int operator <= (CHonSo);
13.     int operator == (CHonSo);
14.     int operator != (CHonSo);
15. };
```

LỚP HỖN SỐ

```
1. CHonSo CHonSo::operator-
                        (CHonSo x)
2. {
3.     CHonSo temp;
4.     temp.nguyen=nguyen–x.nguyen;
5.     temp.tu=tu*x.mau-mau*x.tu;
6.     temp.mau=mau*x.mau;
7.     return temp;
8. }
9. a(b/c)-d(e/f) = a+(b/c)–
   d(e/f)=a-d + ((bf-ce)/(cf))
```

LỚP HỖN SỐ

- Định nghĩa các toán tử so sánh

```
1. int CHonSo::operator>
                        (CHonSo x)
2. {
3.     CHonSo temp = *this – x;
4.     float dau = temp.nguyen +
        (float)temp.tu/temp.mau;
5.     if(dau > 0)
6.         return 1;
7.     else
8.         return 0;
9. }
```

LỚP HỖN SỐ

- Định nghĩa các toán tử so sánh

```
1. int CHonSo::operator<
                        (CHonSo x)
2. {
3.     CHonSo temp = *this – x;
4.     float dau = temp.nguyen +
         (float)temp.tu/temp.mau;
5.     if(dau < 0)
6.         return 1;
7.     else
8.         return 0;
9. }
```

LỚP HỖN SỐ

- Định nghĩa các toán tử so sánh

```
1. int CHonSo::operator>=
                        (CHonSo x)
2. {
3.     if ((*this < x)==0)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP HỖN SỐ

- Định nghĩa các toán tử so sánh

```
1. int CHonSo::operator<=
                        (CHonSo x)
2. {
3.      if ((*this > x)==0)
4.          return 1;
5.      else
6.          return 0;
7. }
```

LỚP HỖN SỐ

- Định nghĩa các toán tử so sánh

```
1. int CHonSo::operator !=
                        (CHonSo x)
2. {
3.     CHonSo temp = *this – x;
4.     float dau = temp.nguyen +
         (float)temp.tu/temp.mau;
5.     if(dau != 0)
6.         return 1;
7.     else
8.         return 0;
9. }
```

LỚP HỖN SỐ

- Định nghĩa các toán tử so sánh

```
1. int CHonSo::operator ==
                        (CHonSo x)
2. {
3.      if ((*this != x)==0)
4.          return 1;
5.      else
6.          return 0;
7. }
```

LỚP ĐIỂM

- Bài 3 : 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.

LỚP ĐIỂM

- Khai báo lớp

```
1.  class CDiem
2.  {
3.    private:
4.      float x;
5.      float y;
6.    public:
7.      float operator-(CDiem);
8.      int operator > (CDiem);
9.      int operator < (CDiem);
10.     int operator >= (CDiem);
11.     int operator <= (CDiem);
12.     int operator == (CDiem);
13.     int operator != (CDiem);
14. };
```

LỚP ĐIỂM

```
1. float CDiem::operator-
                        (CDiem d)
2. {
3.     float d1=sqrt(x*x+y*y);
4.     float d2=
          sqrt(d.x*d.x+d.y*d.y);
5.     return abs(d1-d2);
6. }
```

LỚP ĐIỂM

- Định nghĩa các toán tử so sánh

```
1. int CDiem::operator>
                        (CDiem x)
2. {
3.     float kc = *this – x;
4.     if(kc > 0)
5.         return 1;
6.     else
7.         return 0;
8. }
```

LỚP ĐIỂM

- Định nghĩa các toán tử so sánh

```
1. int CDiem::operator<
                        (CDiem x)
2. {
3.     float kc = *this – x;
4.     if(kc < 0)
5.         return 1;
6.     else
7.         return 0;
8. }
```

LỚP ĐIỂM

- Định nghĩa các toán tử so sánh

```
1. int CDiem::operator>=
                        (CDiem x)
2. {
3.     if ((*this < x) == 0)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP ĐIỂM

- Định nghĩa các toán tử so sánh

```
1. int CDiem::operator<=
                        (CDiem x)
2. {
3.     if ((*this > x) == 0)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP ĐIỂM

- Định nghĩa các toán tử so sánh

```
1. int CDiem::operator!=
                        (CDiem x)
2. {
3.     float kc = *this – x;
4.     if(kc != 0)
5.         return 1;
6.     else
7.         return 0;
8. }
```

LỚP ĐIỂM

- Định nghĩa các toán tử so sánh

```
1. int CDiem::operator==
                        (CDiem x)
2. {
3.     if((*this != x) == 0)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP ĐIỂM

- **Bài 4.1 : Hãy định nghĩa toán tử so sánh bằng và toán tử so sánh khác cho tất cả các đối tượng CDiem.**

LỚP ĐIỂM

- Khai báo lớp

```
1. class CDiem
2. {
3.   private:
4.     float x;
5.     float y;
6.   public:
7.     int operator == (CDiem);
8.     int operator != (CDiem);
9. };
```

LỚP ĐIỂM

- Định nghĩa các toán tử so sánh

```
1. int CDiem::operator ==
                        (CDiem d)
2. {
3.     if (x == d.x && y == d.y)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP ĐIỂM

- Định nghĩa các toán tử so sánh

```
1. int CDiem::operator !=
                        (CDiem d)
2. {
3.     if ((*this == d) == 0)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP NGÀY

- **Bài 4.2 : Hãy định nghĩa toán tử so sánh bằng và toán tử so sánh khác cho tất cả các đối tượng CNgay.**

LỚP NGÀY

- Khai báo lớp

```
1. class CNgay
2. {
3.   private:
4.     int ng;
5.     int th;
6.     int nm;
7.   public:
8.     int operator == (CNgay);
9.     int operator != (CNgay);
10. };
```

LỚP NGÀY

- Định nghĩa các toán tử so sánh

```
1. int CNgay::operator ==
                        (CNgay x)
2. {
3.     if (ng == x.ng &&
           th == x.th &&
           nm == x.nm)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP NGÀY

- Định nghĩa các toán tử so sánh

```
1. int CNgay::operator !=
                        (CNgay x)
2. {
3.     if ((*this == x) == 0)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP THỜI GIAN

- **Bài 4.3 : Hãy định nghĩa toán tử so sánh bằng và toán tử so sánh khác cho tất cả các đối tượng CThoiGian.**

LỚP THỜI GIAN

- Khai báo lớp

```
1. class CThoiGian
2. {
3.   private:
4.     int gio;
5.     int phut;
6.     int giay;
7.   public:
8.     int operator== (CThoiGian);
9.     int operator!= (CThoiGian);
10. };
```

LỚP THỜI GIAN

- Định nghĩa các toán tử so sánh

```
1. int CThoiGian::operator ==
                        (CThoiGian x)
2. {
3.     if (gio == x.gio &&
           phut == x.phut &&
           giay == x.giay)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP THỜI GIAN

- Định nghĩa các toán tử so sánh

```
1. int CThoiGian::operator !=
                        (CThoiGian x)
2. {
3.     if ((*this == x) == 0)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP ĐƠN THỨC

- **Bài 4.4 : Hãy định nghĩa toán tử so sánh bằng và toán tử so sánh khác cho tất cả các đối tượng CDonThuc.**

LỚP ĐƠN THỨC

- Khai báo lớp

```
1. class CDonThuc
2. {
3.   private:
4.     float heso;
5.     int somu;
6.   public:
7.     int operator == (CDonThuc);
8.     int operator != (CDonThuc);
9. };
```

LỚP ĐƠN THỨC

- Định nghĩa các toán tử so sánh

```
1. int CDonThuc::operator ==
                        (CDonThuc d)
2. {
3.     if (heso == d.heso &&
           somu == d.somu)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP ĐƠN THỨC

- Định nghĩa các toán tử so sánh

```
1. int CDonThuc::operator !=
                        (CDonThuc x)
2. {
3.     if ((*this == x) == 0)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP ĐIỂM KHÔNG GIAN

- **Bài 4.5 : Hãy định nghĩa toán tử so sánh bằng và toán tử so sánh khác cho tất cả các đối tượng CDiemKhongGian.**

LỚP ĐIỂM KHÔNG GIAN

- Khai báo lớp

```
1. class CDiemKhongGian
2. {
3.   private:
4.     float x;
5.     float y;
6.     float z;
7.   public:
8.     int operator==
           (CDiemKhongGian);
9.     int operator!=
           (CDiemKhongGian);
10. };
```

LỚP ĐIỂM KHÔNG GIAN

- Định nghĩa các toán tử so sánh

```
1. int CDiemKhongGian::operator ==
   (CDiemKhongGian d)
2. {
3.     if (x == d.x &&
           y == d.y &&
           z == d.z)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP ĐIỂM KHÔNG GIAN

- Định nghĩa các toán tử so sánh

```
1. int CDiemKhongGian::operator !=
   (CDiemKhongGian d)
2. {
3.     if ((*this == d) == 0)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP ĐƯỜNG THẲNG

- **Bài 4.6 : Hãy định nghĩa toán tử so sánh bằng và toán tử so sánh khác cho tất cả các đối tượng CDuongThang.**

LỚP ĐƯỜNG THẲNG

- Khai báo lớp

```
1. class CDuongThang
2. {
3.   private:
4.     float a;
5.     float b;
6.     float c;
7.   public:
8.     int operator==
                (CDuongThang);
9.     int operator !=
                (CDuongThang);
10. };
```

LỚP ĐƯỜNG THẲNG

- Định nghĩa các toán tử so sánh

```
1. int CDuongThang::
        operator ==(CDuongThang x)
2. {
3.     if (a == x.a &&
           b == x.b &&
           c == x.c)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP ĐƯỜNG THẲNG

- Định nghĩa các toán tử so sánh

```
1. int CDuongThang::operator !=
                    (CDuongThang x)
2. {
3.     if ((*this == x) == 0)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP SỐ PHỨC

- **Bài 4.7 : Hãy định nghĩa toán tử so sánh bằng và toán tử so sánh khác cho tất cả các đối tượng CSoPhuc.**

LỚP SỐ PHỨC

- Khai báo lớp

```
1. class CSoPhuc
2. {
3.   private:
4.     float thuc;
5.     float ao;
6.   public:
7.     int operator == (CSoPhuc);
8.     int operator != (CSoPhuc);
9. };
```

LỚP SỐ PHỨC

- Định nghĩa các toán tử so sánh

```
1. int CSoPhuc::operator ==
                        (CSoPhuc x)
2. {
3.     if (thuc == x.thuc &&
           ao == x.ao)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP SỐ PHỨC

- Định nghĩa các toán tử so sánh

```
1. int CSoPhuc::operator !=
                        (CSoPhuc x)
2. {
3.     if ((*this == x) == 0)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP ĐƯỜNG TRÒN

- **Bài 4.8 : Hãy định nghĩa toán tử so sánh bằng và toán tử so sánh khác cho tất cả các đối tượng CDuongTron.**

LỚP ĐƯỜNG TRÒN

- Khai báo lớp

```
1. class CDiem
2. {
3.   private:
4.     float x;
5.     float y;
6.   public:
7.     int operator == (CDiem);
8. };
```

LỚP ĐƯỜNG TRÒN

- Định nghĩa các toán tử so sánh

```
1. int CDiem::operator ==
                        (CDiem d)
2. {
3.     if (x == d.x && y == d.y)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP ĐƯỜNG TRÒN

- Khai báo lớp

```
1. class CDuongTron
2. {
3.   private:
4.     CDiem tam;
5.     float bankinh;
6.   public:
7.     int operator==(CDuongTron);
8.     int operator!=(CDuongTron);
9. };
```

LỚP ĐƯỜNG TRÒN

- Định nghĩa các toán tử so sánh

```
1. int CDuongTron::operator ==
                        (CDuongTron x)
2. {
3.     if (tam == d.tam &&
           bankinh == x.bankinh)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP ĐƯỜNG TRÒN

- Định nghĩa các toán tử so sánh

```
1. int CDuongTron::operator !=
                        (CDuongTron x)
2. {
3.     if ((*this == x) == 0)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP TAM GIÁC

- **Bài 4.9 : Hãy định nghĩa toán tử so sánh bằng và toán tử so sánh khác cho tất cả các đối tượng CTamGiac.**

LỚP TAM GIÁC

- Khai báo lớp

```
1. class CDiem
2. {
3.   private:
4.     float x;
5.     float y;
6.   public:
7.     int operator == (CDiem);
8. };
```

LỚP TAM GIÁC

- Định nghĩa các toán tử so sánh

```
1. int CDiem::operator ==
                        (CDiem d)
2. {
3.     if (x == d.x && y == d.y)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP TAM GIÁC

- Khai báo lớp

```
1. class CTamGiac
2. {
3.   private:
4.     CDiem A;
5.     CDiem B;
6.     CDiem C;
7.   public:
8.     int operator==
                 (CTamGiac);
9.     int operator !=
                 (CTamGiac);
10. };
```

LỚP TAM GIÁC

- Định nghĩa các toán tử so sánh

```
1. int CTamGiac::
        operator ==(CTamGiac x)
2. {
3.     if (A == x.A &&
           B == x.B &&
           C == x.C)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP TAM GIÁC

- Định nghĩa các toán tử so sánh

```
1. int CTamGiac::operator !=
                    (CTamGiac x)
2. {
3.     if ((*this == x) == 0)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP HÌNH CẦU

- Bài 4.10 : Hãy định nghĩa toán tử so sánh bằng và toán tử so sánh khác cho tất cả các đối tượng CHinhCau.

LỚP HÌNH CẦU

- Khai báo lớp

```
1. class CDiemKhongGian
2. {
3.   private:
4.     float x;
5.     float y;
6.     float z;
7.   public:
8.     int operator==
           (CDiemKhongGian);
9. };
```

LỚP HÌNH CẦU

- Định nghĩa các toán tử so sánh

```
1. int CDiemKhongGian::operator ==
                (CDiemKhongGian d)
2. {
3.     if (x == d.x &&
           y == d.y &&
           z == d.z)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP HÌNH CẦU

- Khai báo lớp

```
1. class CHinhCau
2. {
3.   private:
4.     CDiem tam;
5.     float bankinh;
6.   public:
7.     int operator==
                 (CHinhCau);
8.     int operator !=
                 (CHinhCau);
9. };
```

LỚP HÌNH CẦU

- Định nghĩa các toán tử so sánh

```
1. int CHinhCau::
        operator ==(CHinhCau x)
2. {
3.     if (tam == x.tam &&
           bankinh == x.bankinh)
4.         return 1;
5.     else
6.         return 0;
7. }
```

LỚP HÌNH CẦU

- Định nghĩa các toán tử so sánh

```
1. int CHinhCau::operator !=
                    (CHinhCau x)
2. {
3.     if ((*this == x) == 0)
4.         return 1;
5.     else
6.         return 0;
7. }
```