Lập trình hướng đối tượng

Toán tử vào, ra

Nguyễn Khắc Huy

10/2018
BM Công Nghệ Phần Mềm

Dẫn nhập

- Thư viện <stdio.h>

```
int a; float b;
printf("Nhap gia tri:");
scanf("%d", &a); scanf("%f", &b);
printf("So nguyen vua nhap:%d, %f", a, b);
```

- Thư viện <iostream>

```
int a; float b;
cout<<"Nhap gia tri: ";
cin >> a; cin >> b;
cout << "Cac so vua nhap: " << a << "," << b;
```

Dẫn nhập

- Khai báo lớp PhanSo

```
class PhanSo
{
        int tu;
        int mau;
    public:
        void Nhap();
        void Xuat();
};
```

- Khai báo lớp PhanSo

```
PhanSo a;
a.Nhap();
a.Xuat();
```

Dẫn nhập

- Nhập xuất với iostream

```
CPhanSo a;
cin >> a;
cout << a;
```

- Ký hiệu **>>** được gọi là **toán tử vào**.
- Ký hiệu **<<** được gọi là **toán tử ra**.

Giải quyết vấn đề

- Định nghĩa
  - Toán tử vào (operator >>)
  - Toán tử ra (operator <<)

cho lớp đối tượng PhanSo.

Khai báo

```
class PhanSo
{
private:
        int tu;
        int mau;
public:
        istream& operator >>(istream &is);
        ostream& operator <<(istream &os);
};
```

Vấn đề

- Sử dụng
  - a.operator >>(cin);
  - a.operator << (cout);

  hoặc
  - a >> cin;
  - a << cout;

Vấn đề (tt.)

- Làm thế nào?
  - cin >> a;
  - cout << a;
- Bổ sung thêm phương thức vào lớp istream và ostream
  - class istream {

    istream& opertator >> (const PhanSo ps&);

    }
  - class ostream {

    ostream& opertator >> (const PhanSo ps&);

    }

Giải quyết vấn đề

- Kết hợp sử dụng hàm friend
- "hàm friend" của lớp đối tượng được phép truy xuất đến tất cả các thành phần của đối tượng thuộc về lớp đó bất chấp thành phần được khai báo trong phạm vi nào

Khai báo

```
class PhanSo
{
private:
        int tu;
        int mau;
public:
        friend istream& operator >>(
                        istream &is,PhanSo &x);
        friend ostream& operator <<(
                        ostream &os,PhanSo &x);
};
```

Định nghĩa

□ Định nghĩa toán tử vào

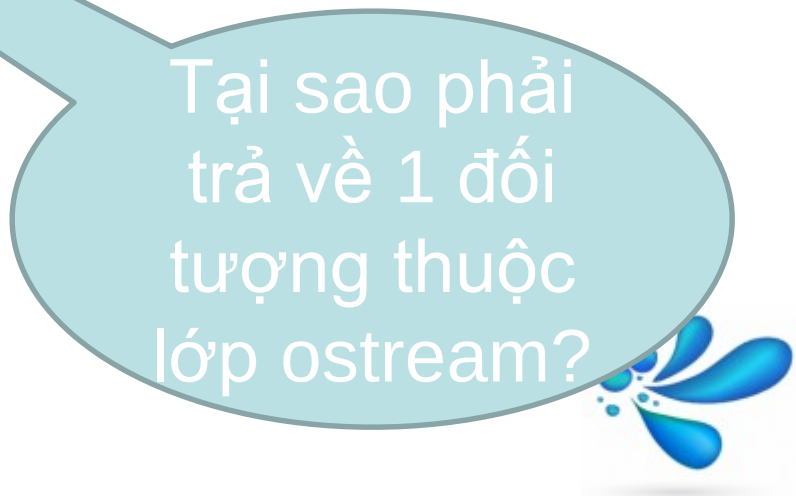
```
istream& operator >>(
                    istream &is, PhanSo &x)
{
    cout << “Nhap tu”;
    is >> x.tu;
    cout << “Nhap mau”;
    is >> x.mau;
    return is;
}
```

Tại sao phải trả về 1 đối tượng thuộc lớp istream?

Định nghĩa

□ Định nghĩa toán tử ra

```
ostream& operator <<(
                ostream &os, PhanSo &x)
{
    os<< x.tu<<“/”<<x.mau;
    return os;
}
```



Sử dụng

- Xét đoạn chương trình sau:

```
CPhanSo a;
cin >> a;
cout <<a ;
```

- Trong câu lệnh thứ hai của đoạn chương trình trên ta nói: hàm operator >> được gọi thực hiện với 2 đối số là cin và đối tượng a.
- Trong câu lệnh thứ ba của đoạn chương trình trên ta nói: hàm operator << được gọi thực hiện với 2 đối số là cout và đối tượng a.

Một ví dụ khác

- Toán tử 1 ngôi: prefix và postfix
  - x++
  - ++x
- Khai báo và sử dụng hai toán tử trên như thế nào?

Lưu ý thêm

- Xét đoạn chương trình sau:

```
CPhanSo a,b,c;
cin >> a >> b >> c;
cout << a << b << c;
```

- Trong câu lệnh thứ hai của đoạn chương trình trên ta nói: hàm operator >> được gọi thực hiện 3 lần.
- Trong câu lệnh thứ ba của đoạn chương trình trên ta nói: hàm operator << được gọi thực hiện 3 lần.

Demo

- Yêu cầu: Hãy định nghĩa toán tử vào và toán tử ra cho lớp đối tượng CNgay.

Bài tập

□ Bài tập 4.1:

Bổ sung vào lớp **phân số** những phương thức sau:

(Nhóm tạo hủy)

- ✓ Khởi tạo mặc định phân số = 0.
- ✓ Khởi tạo với tử và mẫu cho trước.
- ✓ Khởi tạo với giá trị phân số cho trước.
- ✓ Khởi tạo từ một phân số khác.

(Nhóm toán tử)

- ✓ Toán tử xử lý +, -, *, /
- ✓ Toán tử một ngôi: ++, --
- ✓ Toán tử nhập, xuất: >>, <<
- ✓ Toán tử so sánh >, <, ==, !=, >=, <=

Bài tập

□ Bài tập 4.2:

Bổ sung vào lớp **số phức** những phương thức sau:

(Nhóm tạo hủy)

- ✓ Khởi tạo mặc định số phức = 0.
- ✓ Khởi tạo với phần thực và phần ảo cho trước.
- ✓ Khởi tạo với giá trị thực cho trước.
- ✓ Khởi tạo từ một số phức khác.

(Nhóm toán tử)

- ✓ Toán tử xử lý +, -, *, /
- ✓ Toán tử một ngôi: ++, --
- ✓ Toán tử gán =
- ✓ Toán tử nhập, xuất: >>, <<
- ✓ Toán tử so sánh >, <, ==, !=, >=, <=

Bài tập

□ Bài tập 4.3:

Bổ sung vào lớp **đơn thức** những phương thức sau:

(Nhóm tạo hủy)

- ✓ Khởi tạo mặc định đơn thức = 0.
- ✓ Khởi tạo với hệ số và số mũ cho trước.
- ✓ Khởi tạo với hệ số cho trước, số mũ = 0.
- ✓ Khởi tạo từ một đơn thức khác.

(Nhóm toán tử)

- ✓ Toán tử xử lý +, -
- ✓ Toán tử gán =
- ✓ Toán tử một ngôi: ++, --.
- ✓ Toán tử nhập, xuất: >>, <<.

Bài tập

□ Bài tập 4.4:

Bổ sung vào lớp **học sinh** những phương thức sau:

(Nhóm tạo hủy)

- ✓ Khởi tạo với họ tên và điểm văn, toán cho trước.
- ✓ Khởi tạo với họ tên cho trước, điểm văn, toán = 0.
- ✓ Khởi tạo từ một học sinh khác.

(Nhóm toán tử)

- ✓ Toán tử gán: =.
- ✓ Toán tử nhập, xuất: >>, <<.
- ✓ Toán tử so sánh >, <, ==, !=, >=, <=

Bài tập

Bài tập 4.5:

Bổ sung vào lớp **Mảng Số Nguyên** những phương thức sau:

(Nhóm tạo hủy)

- ✓ Khởi tạo mặc định mảng kích thước = 0.
- ✓ Khởi tạo với kích thước cho trước, các phần tử = 0.
- ✓ Khởi tạo từ một mảng int [ ] với kích thước cho trước.
- ✓ Khởi tạo từ một đối tượng IntArray khác.
- ✓ Hủy đối tượng IntArray, thu hồi bộ nhớ.

(Nhóm toán tử)

- ✓ Toán tử gán: =.
- ✓ Toán tử nhập, xuất: >>, <<