Đa năng hóa toán tử và hàm

Mục tiêu:

Tìm hiểu về hàm toán tử trong lớp đối tượng.

Yêu cầu:

Nắm vững những nội dung đã trình bày trong các bài hướng dẫn thực hành từ bài 1 đến bài 5

5.1 Hàm toán tử trong lớp đối tượng

Cũng như các hàm thông thường, chúng ta được phép định nghĩa các hàm toán tử cho lớp đối tượng. Để hiểu rõ hơn về hàm toán tử trong lớp đối tượng, chúng ta xét ví dụ định nghĩa các hàm toán tử + và so sánh > cho lớp đối tượng PhanSo đã xây dựng ở trên.

Bước 1: vào file PhanSo.h, trong phần public của lớp PhanSo, thêm vào khai báo cho hàm dựng và 2 hàm toán tử + và so sánh > như sau:

```
PhanSo();
PhanSo(int iTuSo, int iMauSo);
PhanSo operator +(const PhanSo &a);
bool operator >(const PhanSo &a);
```

Bước 2: vào file PhanSo.cpp, thêm phần cài đặt cho các hàm dựng và 2 toán tử + và so sánh > như sau:

```
PhanSo::PhanSo()
{
        m_iTuSo = 0;
        m_iMauSo = 1;
 }
 PhanSo::PhanSo(int iTuSo, int iMauSo)
 {
        if (iMauSo == 0)
        {
                cout << "Loi: mau so nhap vao bang 0." return;
        }

        m_iTuSo = iTuSo;
        m_iMauSo = iMauSo;
 }
```

```
PhanSo PhanSo::operator +(const PhanSo &a)
{
	return this->cong(a);
}
bool PhanSo::operator >(const PhanSo &a)
{
	long	lMauSoQuyDong = (long)this->iMauSo * a.iMauSo;
	long	lTuSoQuyDong = (long)this->iTuSo * a.iMauSo;
	long	lTuSoQuyDongA = (long)a.iTuSo * this->iMauSo;

	if (lMauSoQuyDong > 0)
		return lTuSoQuyDong > lTuSoQuyDongA;
	return lTuSoQuyDong < lTuSoQuyDongA;
}
```

Bước 3: sửa lại file main.cpp để sử dụng lớp PhanSo với những toán tử vừa tạo như sau:

```
#include "iostream"
#include "PhanSo.h"
using namespace std;
void main()
{
	PhanSo		a(1, 2);
	PhanSo		b(1, 3);
	PhanSo		c;
	c = a + b;
	cout << "Tu so c = " << c.LayTuSo() << endl;
	cout << "Mau so c = " << c.LayMauSo() << endl;
	if (a > b)
		cout << "Phan so a lon hon phan so b";
	else
	cout << "Phan so a nho hon hoac bang phan so b.";
}
```

5.2 Bài tập

1. ***Làm lại bài số phức với một phương thức thiết lập duy nhất cho phép quan điểm một số thực như một số phức đặc biệt (phần ảo bằng 0). Định nghĩa các phép toán +, -, *, /, = =, !=,*** ! trên số phức. Định nghĩa phép toán << và >> để xuất và nhập dữ liệu vào số phức.
2. Làm lại bài phân số với các phương thức thiết lập cho phép sử dụng một số nguyên như một phân số đặc biệt (mẫu số bằng 1). Định nghĩa các phép toán +, -, *, /, = =, !=, ! trên phân số. Định nghĩa phép toán << và >> để xuất và nhập dữ liệu vào phân số.
3. Định nghĩa lớp dữ liệu CTimeSpan để biểu diễn khái niệm khoảng thời gian, các hàm thành phần và các phép toán cần thiết.
4. Định nghĩa lớp CTime biểu diễn khái niệm thời điểm có các thành phần giờ phút giây. Định nghĩa các phép toán +, - (cộng, trừ thêm một số nguyên giây), - (phép trừ hai CTime để được một CTimSpan), ++, -- (thêm bớt một giây). Phép toán <<, >> để xuất, nhập dữ liệu loại CTime. Áp dụng lớp CTime để tạo đồng hồ in ở góc trên bên phải màn hình.
5. Định nghĩa lớp CDate biểu diễn khái niệm ngày, tháng, năm với các phép toán +, - (cộng, trừ thêm một số ngày), ++, -- (thêm bớt một ngày), - (khoảng cách giữa hai CDate tính bằng ngày). Phép toán <<, >> để xuất, nhập dữ liệu loại CDate. Áp dụng lớp CDate để giải bài toán tính lãi suất gởi ngân hàng.
6. Hãy định nghĩa lớp CString biểu diễn khái niệm chuỗi ký tự với các phương thức thiết lập, huỷ bỏ, các hàm thành phần và các phép toán cần thiết (+, gán, so sánh hai chuỗi).
7. Định nghĩa lớp biểu diễn khái niệm đa thức có bậc bất kỳ với các hàm thành phần và phép toán cần thiết.
8. Định nghĩa lớp CVector biểu diễn khái niệm vector trong không gian có số chiều bất kỳ với các hàm thành phần và các phép toán cần thiết.
   Định nghĩa lớp CMatrix biểu diễn khái niệm ma trận có kích thước bất kỳ với các hàm thành phần và các phép toán cần thiết.
   Viết hàm tính tích của một ma trận và một vector. Tích của hai ma trận.
9. Hãy định nghĩa lớp INTEGER có thể hoạt động như để mỗi INTEGER giống hệt như một 'int' của ngôn ngữ C/C++.
10. Hãy định nghĩa lớp MYINT có hoạt động như kiểu dữ liệu 'int' nhưng phép cộng hai MYINT hoạt động như phép trừ hai int và ngược lại.
11. Định nghĩa lớp CExpr để biểu diễn một biểu thức toán học ở dạng trung tố với các phép toán thông dụng +, -, *, / và cho phép có dấu ngoặc. Phần giao diện có thể như sau:

```
class CExpr {
      char *expr;
public:
```

```
CExpr();
        CExpr(char *s);
        ~CExpr();
        double eval();
        //...
 };
main(){
        CExpr("2 + 3 * 5 - (3 + (7 + 2) * 3) / 2");
        cout << "Gia tri cua bieu thuc: " << expr->eval();
}
```

12. Cho đoạn chương trình sau:

```
#include <iostream.h>
void main(){
        cout << "Hello, world.\n";
}
```

Hãy sửa lại chương trình trên, nhưng không sửa chữa gì hàm main, để chương trình có thể tạo ra kết xuất:

Entering the Hello program saying...
Hello, world.
Then exiting...