CSDN 学院

C++语言基础

迂者 - 贺利坚

http://blog.csdn.net/sxhelijian/

http://edu.csdn.net

本节主题：

重载流插入运算符和流提取运算符

流插入“<<”和流提取“>>”

- 我们已经有：

```
#include<iosream>
int a;
cin>>a;  //由istream类支持
cout<<a; //由ostream类支持
```

- 内幕：作为输入/输出的<<和>>重载了按位移的运算符
- 本节目标: 通过重载，直接用“<<”和“>>”来输出和输入用户自己定义的类型的数据。

```
Time t1, t2;
cin>>t1>>t2;
cout<<t1<<t2;
```

```
Complex c1(2,3), c2(3,-1);
cin>>c1>>c2;
cout<<"c1+c2="<<(c1+c2)<<endl;
```

对“<<”和“>>”重载的函数形式

- 形式

ostream &operator<<(ostream&, const 自定义类&);

istream &operator>>(istream&,自定义类&);

运算符	第一个参数	第二个参数	函数的类型
<<	ostream&	要进行输出操作的类	ostream&
>>	istream&	要进行输入操作的类	istream&
将重载提取“>>”和插入“<<”的函数**作为友元函数或普通的函数**，而**不能**将它们定义为**成员函数**。			

重载流插入运算符“<<”

```
#include <iostream>
using namespace std;
class Complex
{
public:
   Complex( ){real=0;imag=0;}
   Complex(double r,double i){real=r;imag=i;}
   Complex operator + (Complex &c2);
   friend ostream& operator << (ostream&, const Complex&);
private:
   double real;
   double imag;
};
```

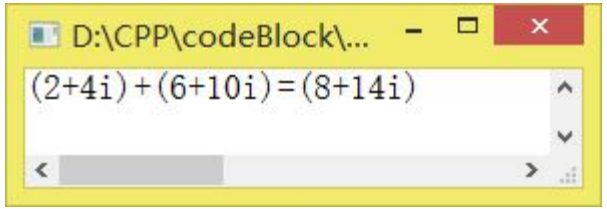
```
Complex Complex::operator + (Complex &c2)
{
   return Complex(real+c2.real,imag+c2.imag);
}
```

```
ostream& operator << (ostream& output,const Complex& c)
 {
   output<<"("<<c.real<<"+"<<c.imag<<"i)";
   return output;
}
```

cout<<c1<<c2;等同于

operator<<(operator<<(cout,c1),c2)

```
int main( )
{
   Complex c1(2,4),c2(6,10),c3;
   c3=c1+c2;
   cout<<c1<<'+'<<c2<<'='<<c3<<endl;
   return 0;
}
```



重载流提取运算符“>>”

```
#include <iostream>
using namespace std;
class Complex
{
public:
   friend ostream& operator << (ostream&,const Complex&);
   friend istream& operator >> (istream&,Complex&);
private:
   double real;
   double imag;
};
```

```
ostream& operator << (ostream& output,const Complex& c)
{
   output<<"("<<c.real<<"+"<<c.imag<<"i)";
   return output;
}
```

```
istream& operator >> (istream& input,Complex& c)
{
   cout<<"input real part and imaginary part: ";
   input>>c.real>>c.imag;
   return input;
}
```

```
int main( )
{
   Complex c1,c2;
   cin>>c1>>c2;
   cout<<"c1="<<c1<<endl;
   cout<<"c2="<<c2<<endl;
   return 0;
}
```

D:\CPP\codeBlock\example\bin\Deb...

```
input real part and imaginary part: 3 4
input real part and imaginary part: 6 -2
c1=(3+4i)
c2=(6+-2i)
```

对复数输入输出的完善

```
istream& operator >> (istream& input,Complex& c)
{
    int a,b;
    char sign,i;
    do
    {
        cout<<"input a complex number(a+bi或a-bi):";
        input>>a>>sign>>b>>i;
    }
    while(!((sign=='+'||sign=='-')&&i=='i'))
    c.real=a;
    c.imag=(sign=='+')?b:-b;
    return input;
}
```

```
ostream& operator << (ostream& output, const Complex& c)
{
    output<<"("<<c.real;
    if(c.imag>=0) output<<"+";
    output<<c.imag<<"i)";
    return output;
}
```

D:\CPP\codeBlock\example\bin\...

```
input a complex number(a+bi或a-bi):3+4t
input a complex number(a+bi或a-bi):2-4i
c1=(2-4i)
c2=(6+10i)
c1+c2=(8+6i)
```

体会：运算符重载的重要性与实用意义

- 运算符重载使类的设计更加丰富多彩，扩大了类的功能和使用范围，使程序易于理解，易于对对象进行操作，它体现了为用户着想、方便用户使用的思想。
- 有了运算符重载，在声明了类之后，人们就可以像使用标准类型一样来使用自己声明的类。
  - 类的声明往往是一劳永逸的，有了好的类，用户在程序中就不必定义许多成员函数去完成某些运算和输入输出的功能，使主函数更加简单易读。
- 好的运算符重载能体现面向对象程序设计思想。

CSDN 学院

THANKS

本课程由 迁者-贺利坚 提供

CSDN网站：www.csdn.net
企业服务：http://ems.csdn.net/
人才服务：http://job.csdn.net/
CTO俱乐部：http://cto.csdn.net/
高校俱乐部：http://student.csdn.net/
程序员杂志：http://programmer.csdn.net/

CODE平台：https://code.csdn.net/
项目外包：http://www.csto.com/
CSDN博客：http://blog.csdn.net/
CSDN论坛：http://bbs.csdn.net/
CSDN下载：http://download.csdn.net/