

第9章 统一接口 不同实现—多态性

1、多态性的基本概念

2、派生类对象替换基类对象

3、虚函数的定义

4、抽象类的定义

5、宠物类的设计

6、运算符重载

7、日期类对象判断大小

8、分数类对象运算符重载

运算符重载

□ **运算符重载**指赋予运算符新的操作功能，主要用于对类的对象的操作

□ **运算符+**意味着多少对象类型的加法呢？

□ **还可以定义新的对象类型加法**

□ **运算符重载定义形式：**

〈类型〉 〈类名〉::operator 〈操作符〉(〈参数表〉)

{

函数体

}

两个虚数的运算符重载

- 首先定义虚数类

- 虚数可以描述为： $a+bi$

- a 与 b 看成实数，定义成double类型

- 成员函数除了构造与析构外，还有：

- 输出虚数、修改虚数、得到实部 a 、得到虚部 b

- 相加 $+$ 、判相等 $==$

虚数类的定义

```
#include <iostream>
using namespace std;
class Complex
{
private:
    double real, imag;
public:
    Complex(double r = 0, double i = 0): real(r), imag(i){ }
    double Real(){return real;}
    double Imag(){return imag;}
    Complex operator +(Complex&);
    Complex operator +(double);
    bool operator ==(Complex);
    ~Complex(){ };
};
```

运算符+和==重载

```
Complex Complex::operator + (Complex &c)    // 重载运算符 +，两边是虚数对象
{
    Complex temp;
    temp.real = real+c.real;
    temp.imag = imag+c.imag;
    return temp;
}

Complex Complex::operator + (double d)      // 重载运算符+，左边是虚数对象，右边是双精度数
{
    Complex temp;
    temp.real = real+d;
    temp.imag=imag;
    return temp;
}

bool Complex::operator ==(Complex c)       // 重载运算符==
{
    if (real == c.real && imag == c.imag)
        return true;
    else
        return false;
}
```

测试复数相加与判相等运算符重

```
int main()
```

```
{
```

```
    Complex c1(3,4),c2(5,6),c3;
```

```
    cout << "C1 = " << c1.Real() << "+j" << c1.Imag() << endl;
```

```
    cout << "C2 = " << c2.Real() << "+j" << c2.Imag() << endl;
```

```
    c3 = c1+c2;
```

```
    cout << "C3 = " << c3.Real() << "+j" << c3.Imag() << endl;
```

```
    c3 = c3+6.5;
```

```
    cout << "C3 + 6.5 = " << c3.Real() << "+j" << c3.Imag() << endl;
```

```
    if ( c1==c2 )
```

```
        cout<<"两个复数相等";
```

```
    else
```

```
        cout<<"两个复数不相等";
```

```
    return 0;
```

```
}
```

C1 = 3+j4

C2 = 5+j6

C3 = 8+j10

C3 + 6.5 = 14.5+j10

c1和c2两个复数不相等

微软拼音 半 :

单目运算符重载

□ 运算符++分前置运算符和后置运算符

□ 例如： ++Y与Y++

□ 前置运算符定义

```
Complex Complex::operator ++ ()  
{ real+=1;  
  return *this;  
}
```

□ 后置运算符定义

```
Complex Complex::operator ++ (int)  
{ real+=1;  
  return *this;  
}
```

感谢收看！