

# 第7章 数据的抽象与封装

1、实体、对象与类的概念	2、类的定义	3、对象声明与引用	4、私有、公有与保护
5、日期类的设计	6、两种程序设计思想	7、汽车类的设计	8、几何图形圆类的设计
9、构造函数的定义	10、重载构造函数	11、析构函数的定义	12、整数翻译函数
13、实际意义的析构函数	14、Person类的设计	15、对象与指针	16、this指针

# this指针

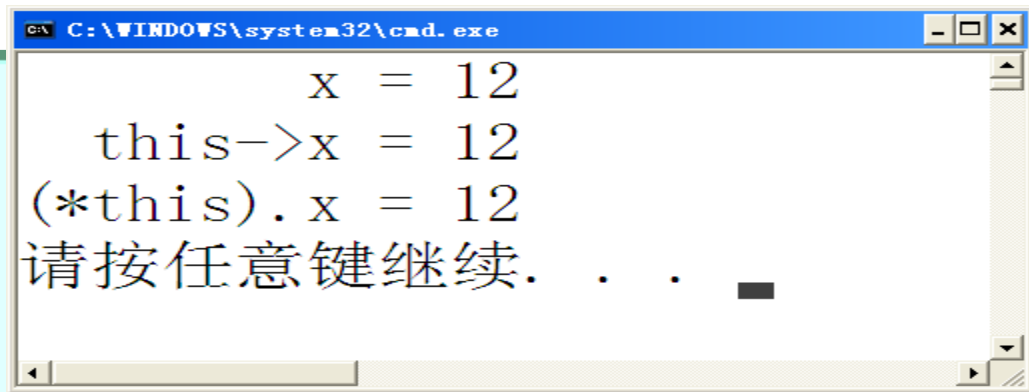
---

- 每一个类的成员函数都包含一个指向本类对象的指针
- 指针名为this
- 该指针指向本类对象的起始地址

# this指针的使用

```
#include <iostream>
using namespace std;
class Test
{
    int x;
public:
    Test( int = 0 );
    void print();
};
Test::Test( int a ) { x = a; }           // 构造函数
void Test::print()
{
    cout << "    x = " << x << endl;
    cout << "this->x = " << this->x << endl;
    cout << "(*this).x = " << ( *this ).x << endl;
}
int main()
{
    Test testObject( 12 );
    testObject.print();
    return 0;
}
```

}//如果在**print()**函数中增加**int x=13;**输出结果如何变化?



```
C:\WINDOWS\system32\cmd.exe
x = 12
this->x = 12
(*this).x = 12
请按任意键继续. . .
```

# this指针再举例

- 当类中数据成员名与成员函数中的形参名相同时，用this指针加以区分
- 例如下面程序，如果没有this->，形参赋给形参

```
Class Time
{
private:
    int hour,minute,second;
public:
    void set(int hour,int minute,int second)
    {
        this->hour = hour;
        this->minute = minute;
        this->second = second;
    }
}
```

# 分数类中This指针具体使用

---

//两个整数为参数的构造函数

**Fraction::Fraction(int a,int b)**

**{**

**set(a,b);** //调用成员函数

**}**

//设置分子、分母

**void Fraction:: set(int a,int b)**

**{**

**this->a=a;**

**this->b=b;**

**}**

# 分数类中This指针具体使用

//分数相加，本类对象加u

**Fraction Fraction::add(Fraction u)**

{

int tmp;

Fraction v;

v.a=a\*u.b+b\*u.a; //分子

v.b=b\*u.b;//分母

tmp=divisor(v.a ,v.b);//计算分

v.a=v.a/tmp;//约去公约数

v.b=v.b/tmp;//约去公约数

return v; //返回结果

}

//分数相加，本类对象加u

**Fraction Fraction::add(Fraction u)**

{

int tmp;

Fraction v;

//

a=a\*u.b+b\*u.a; //分子

b=b\*u.b;//分母

tmp=divisor(a ,b);//计算分子、分母的公约数

a=a/tmp;//约去公约数

b=b/tmp;//约去公约数

return \*this; //返回结果

}

**z=x+y; 表达式计算应该不改变x和y的值**

**右边add ( ) 函数改变了本类的值**

---

感谢收看！