

拷贝构造函数

- C++中除普通的构造函数外，还有一类特殊的构造函数——拷贝构造函数。

拷贝构造函数的作用是用一个已经存在的对象来初始化一个正在创建的新对象。拷贝构造函数有如下特征：

- （1）拷贝构造函数名与类名相同，形参只有一个，是对象的引用，所以，不能重载拷贝构造函数。拷贝构造函数的原形为：
- <类名>(<类名> &对象名);

- (2) 拷贝构造函数无任何函数返回类型说明。
- (3) 如果在类声明中没有给出拷贝构造函数，系统会自动给出一个默认的拷贝构造函数，该拷贝构造函数只进行对象数据成员间的对位拷贝，即所谓的“浅拷贝”。
- (4) 在某些情况下，用户必须在类定义中给出一个显式的拷贝构造函数，以实现用户指定的用一个对象初始化另一个对象的功能，即所谓的“深拷贝”。

- (5) 在以下3种情况下，系统会自动调用拷贝构造函数：
- ①当使用下面的声明语句用一个已存在的对象初始化一个新对象时，系统会自动调用拷贝构造函数：
- <类名> <新对象名>(<已存在对象名>);
- 或
- <类名> <新对象名> = <已存在对象名>;
- ②对象作为实参，在函数调用开始进行实参和形参结合时，会自动调用拷贝构造函数，完成由已知的实参对象初始化形参新对象的功能。

- ③如果函数的返回值是类的对象，在函数调用完成返回时，系统自动调用拷贝构造函数，用return后面的已知对象来初始化一个临时新对象（所创建的临时对象只在外表达式范围内有效，表达式结束时，系统将自动调用析构函数撤销该临时对象）。
- 【例2-10】对于例2-9设计的类，如果在主函数中需要用已知IntArray类对象来初始化一个新的IntArray类对象，直接用系统提供的默认拷贝构造函数，看看会有什么问题以及如何解决。

```
#include <iostream>
using namespace std;
class IntArray
{
public:
    IntArray(int sz) //有参构造函数
    {
        m_size=sz;
        m_ptr=new int[sz];
    }
    IntArray() //无参构造函数
    {
        m_size=10;
        m_ptr=new int[m_size];
    }
    void displayArraySize( ) //显示数组信息
    {
        cout<<"The size of this array is: "
            <<m_size<<endl;
    }
}
```

```
~IntArray()          //析构函数
{
    cout<<"Destructing Array with size "
        <<m_size<<endl;
    delete []m_ptr;    //收回额外空间
}
private:
    int m_size;
    int *m_ptr;
};

int main()
{
    IntArray x(20);
    IntArray y(x); //用已存在对象x初始化新建对象y
    x.displayArraySize();
    y.displayArraySize();
    return 0;
}
```

- 下图是对象x和对象y的数据成员取值示意图。



// 添加拷贝构造函数示例

class IntArray

{

public:

.....

IntArray(IntArray &x) //拷贝构造函数

{

 m_size=x.m_size;

 m_ptr=new int[m_size];

}

.....

private:

int m_size;

int *m_ptr;

};