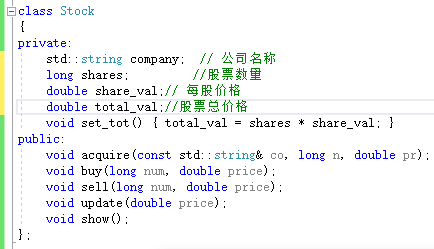
1、定义公有函数调用形参 形参是 私有成员的变量名 2、在main()中调用函数参数的数据类型对应 私有成员类型

3、私有函数可以编写代码 修改私有变量的值 而外部要修改 私有变量值 则需要调用公有函数 { 中调用私有函数 } 来间接修改

实现类成员函数 使用作用域解析运算符

Void 类名：：成员函数名(double 参数名)



在类声明和方法定义之外使用updata()时，需要采取特殊的措施 //后面补充

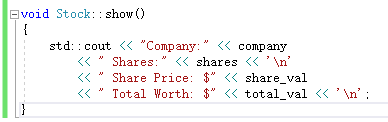
使用类创建对象

Stock kate, joe;

使用对象成员函数 和使用结构成员一样 使用成员运算符

kate.show() //每个对象都像结构一样使用自己的定义的数据

joe.show()



现在 调用函数 joe.show() 使用 show()方法 可以看成 joe.shares joe.share\_val



地址的差别

joe. Shares 将占据一个内存块 kate.Shares将占用另一个内存块 两者地址不一样

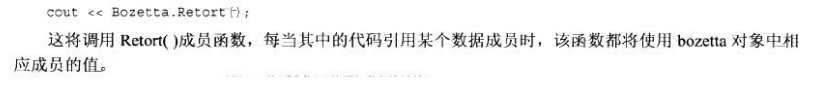
但是

joe.show()和 kate.show() 调用的是同一个方法，他们执行同一个代码块 只是用于不同的数据 show只有一个地址

类名 对象名；

对象{类似数据结构的数据 };

对象名.类公有函数（这里的参数是对象定义好的参数 其类型根据 私有成员的类型匹配）

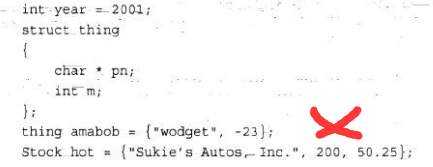


char \* 类名::成员函数()

成员函数不仅是一个char\*类型的函数 还是 一个属于类的 char\*函数

构造析构

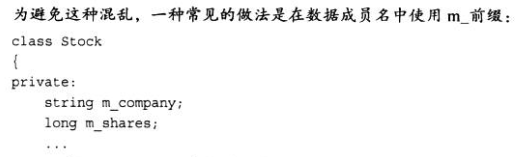
类对象无法像 结构直接通过对象初始化数据

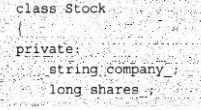


要以 访问公有成员函数来初始化

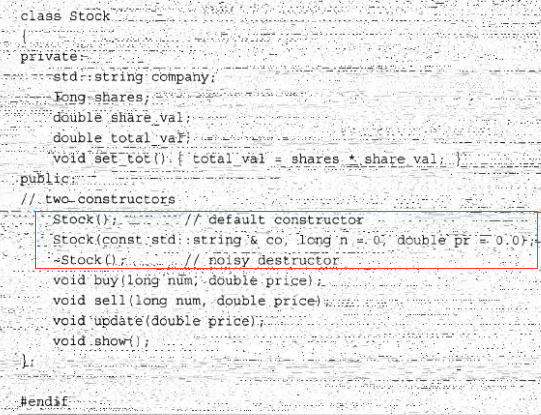


定义构造函数



或者

使用构造函数

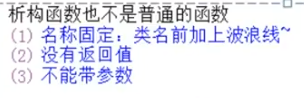


构造函数 constructor

类当中有多个public 时叫做 重载写法是 public: 不是两个冒号public::

析构函数destructor //无法重载只能写一个 析构函数

释放 构造函数 其作用域看构造函数调用时所存在的作用域



构造函数 和释放析构函数的

创建和释放的先后顺序是

成员对象先创建 然后是 mian函数的类对象被创建 先释放 main函数中的 再 释放 成员对象 其中 成员对象释放 是先创建的后释放 后创建的先释放

初始化列表

//基本类型的初始化

Class Object

Public:

Object():x(1),y(2) //这种是初始化

{ x=1; y=2; } //这种是赋值

Private:

{ int x, y; }

//class类型的初始化

Class Child

{

Public:

Child (int x,int y)

Private:

Child m\_child;

}

Class Object

{

Public:

Object(): m\_child( 1,2)

Private:

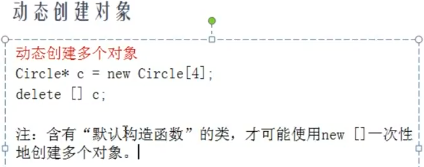
int x, y;

}

两种初始化的区别是 类初始化

New 构造的对象函数 函数调用 类的构造函数并执行构造函数的{内容}

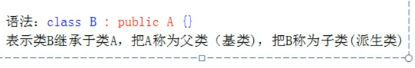
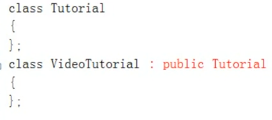
创建 类的函数 必须要有默认构造函数



malloc创建的是一块空间 为地址指针不是 对象



类的继承



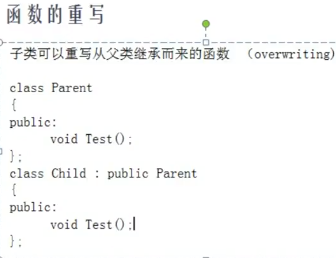
子类包含了基类中的所以成员只是不显象 可以 B对象.A类的成员

//输出时候是 先输出基类的才输出子类的 public只是继承了公有的 而私有的不能被继承 也就无法调用但是会出现在内存中且内存块是先public A的公有 、私有、子类

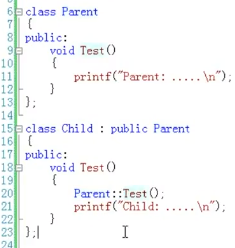
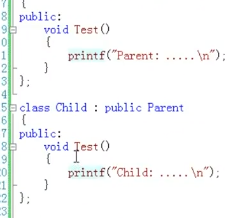
原因是被编译器限制了

修饰符protected 可以被继承但是没法被外部访问

**虚拟继承 虚函数 virtual**

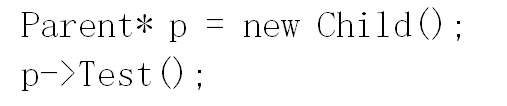


Child 继承Parent Child中觉得Test要修改则 重写了一遍void Test

这是显式的写法 == 这是隐式的写法 效果都一样

父类指针指向子类对象



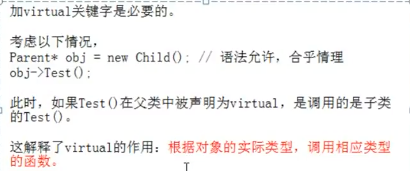
父类\* p = new 子类();

P ->Test(); //由于p 指向的是对象是子类成员函数 本应该调用的是子类的Test()

结果是调用了父类的Test成员函数 这时候我们使用 virtual来解决 重新指向子类Test

虚拟继承virtual 用法 // 在继承语法 的时候 在调用的成员函数 加上virtual 关键字

**Virtual只能用在成员函数 不能用在构造函数**

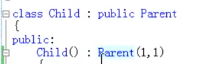
 

继承构造与析构

当main中 调用的构造函数 在父类 和子类中同时存在 且 子类继承父类

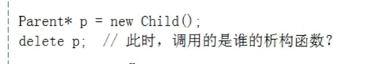
创建的顺序是 先 父类、子类 析构的顺序是 子类 、父类

如何调用有参数的构造函数 类名（）:构造函数名（参数对应有参数构造函数格式）



Virtual析构函数

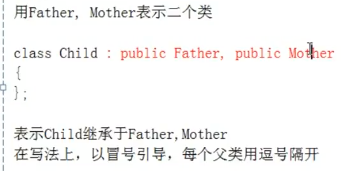
如果不加virtual会出现的问题



当 delete 时编译器无法知道调用delete谁的析构函数 所以手动添加 Virtual析构函数

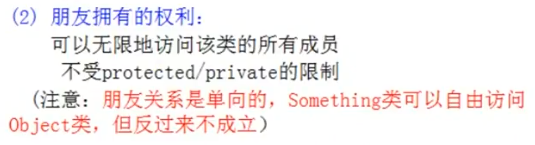
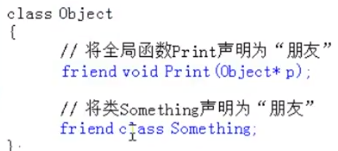
写在 virtual ~ 父类（）

多重继承 //用的很少几乎不用 太麻烦容易出问题



类的朋友friend关键字

friend关键字 写在被调用的类里面 friend不太经常使用尽量不使用



作用 //可以访问 protected/private