单继承：只有一个父类 多继承：有多个父类

构造函数不能是虚函数

有纯虚函数的类叫抽象类

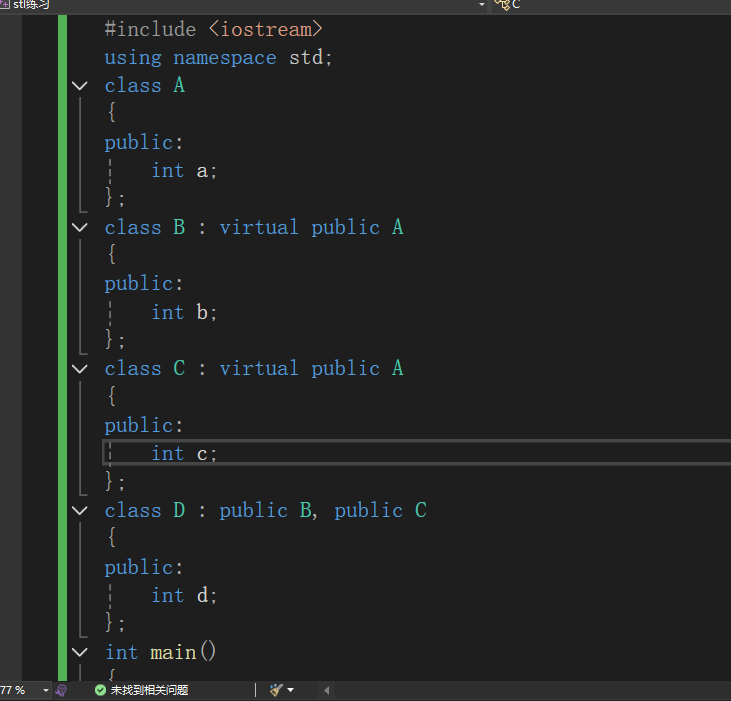
virtual void fun() = 0; 纯虚函数，有纯虚函数的类不能实例化对象

如果子类继承了父类，但是没有实现父类的纯虚函数，子类也依然不可以实例化对象。

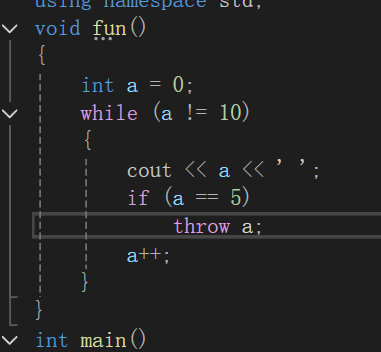
全是纯虚函数的类叫接口类，析构函数也包括 virtual ~father() = 0;

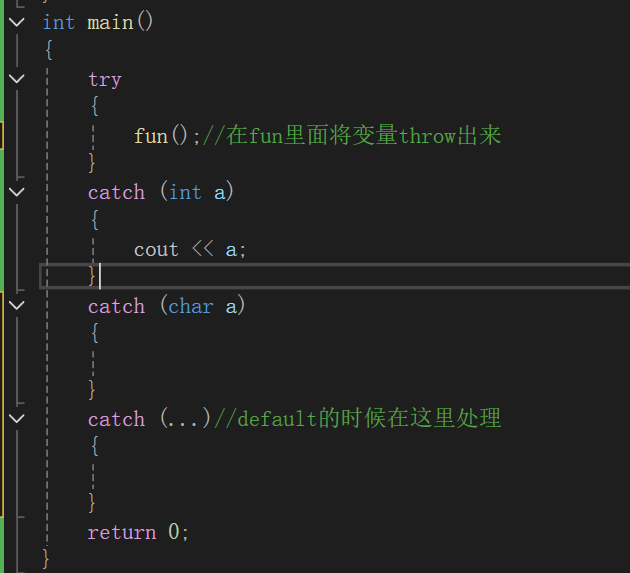
**虚继承**

如下图出现的继承方式就需要虚继承，这样就到类D的时候就不会有两个变量a了



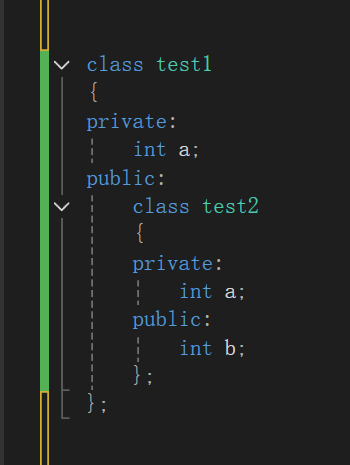
**异常处理**



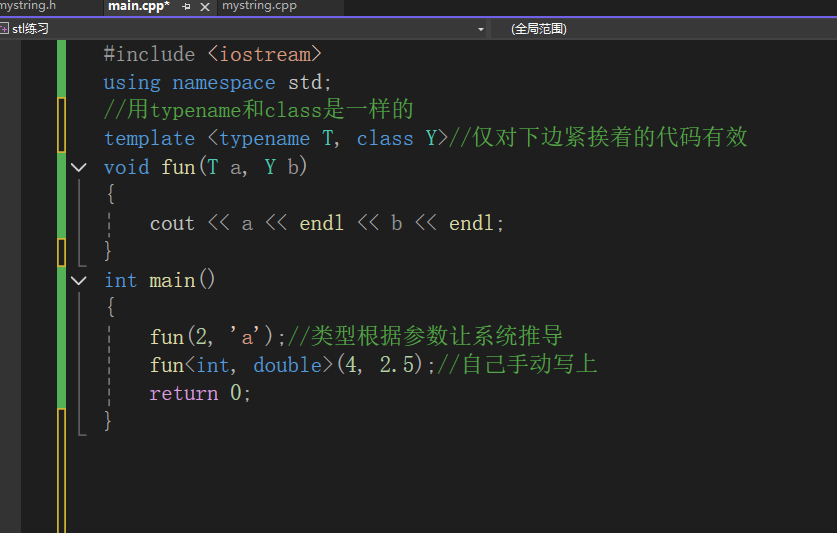


**内部类**

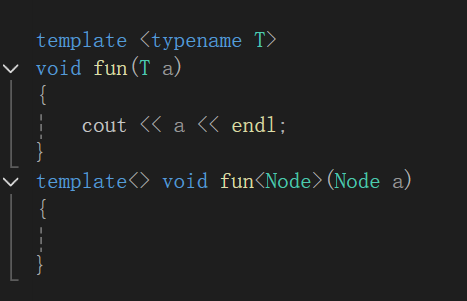
相当于两个独立的类，只是在其内部声明了而已，所以在全局范围内要使用test2的话要加上test1的类名作用域，比如声明test2的变量：test1::test2 tes;



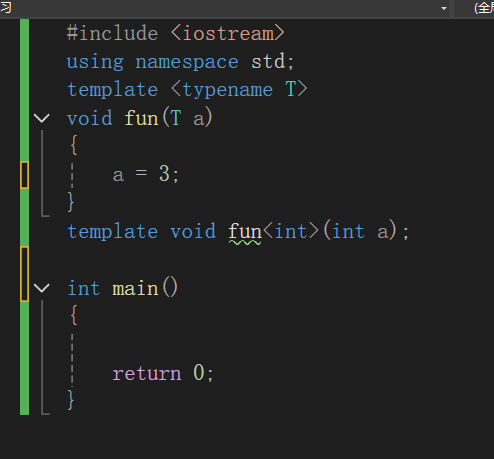
**函数模板**



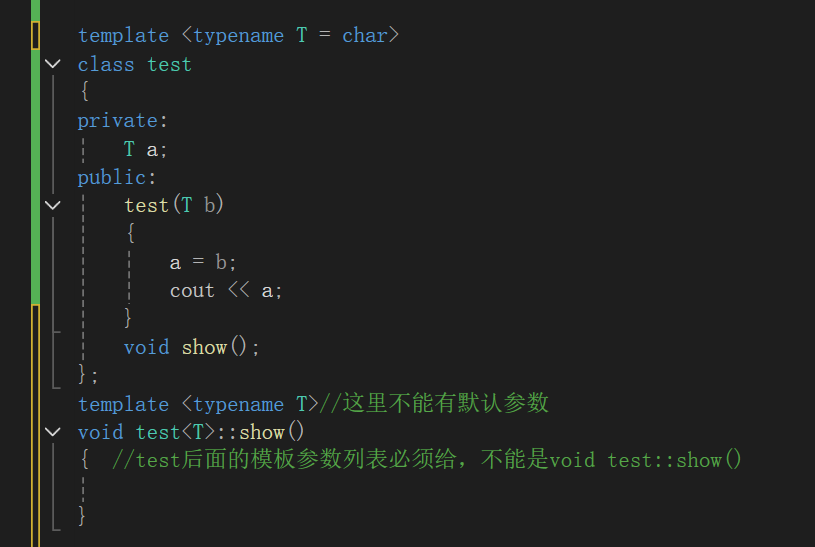
**函数模板的具体化**



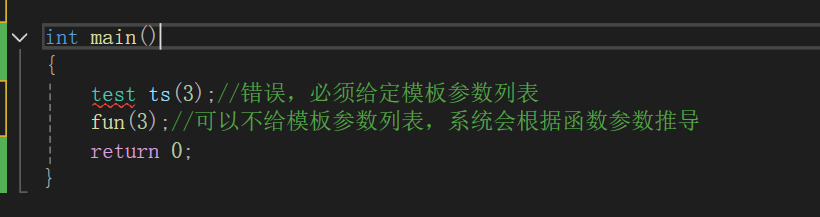
**函数模板的实例化，现在好像废弃了不能用了**



**类模板**



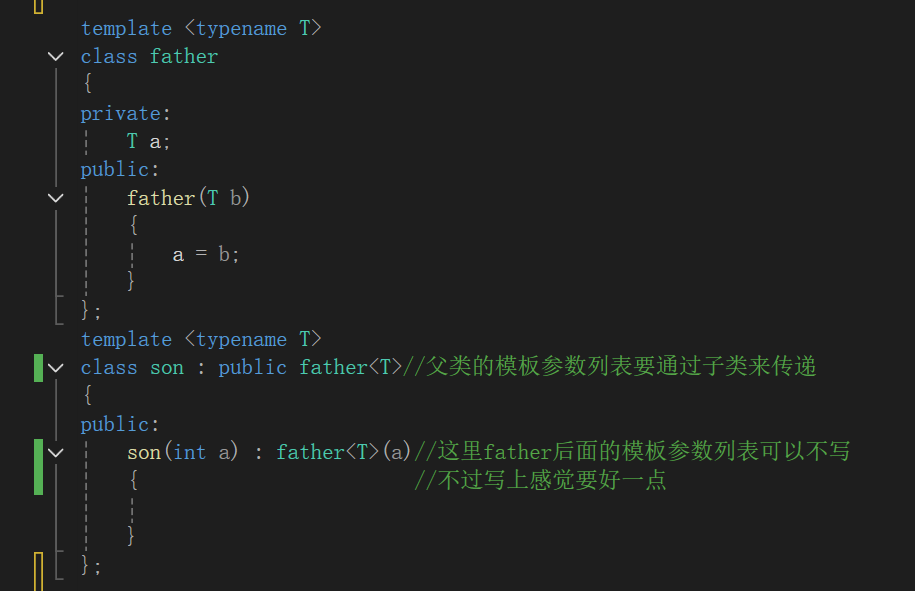
**函数有模板类型推导，但类没有模板类型推导，必须手动指定**



**在定义函数模板和类模板的时候都可以有默认参数，但函数的默认参数给了跟没给效果是一样的**



**继承的模板**



**多态的模板**



**类模板的模板参数列表有默认参数，定义对象的时候不想给参数怎么办**



**覆盖：当父类和子类中出现同名成员时，子类就要覆盖父类的这些同名成员。**

**运算符重载**

后置++类外重载示例

void operator++(test& ts, int n)

{

}

重载类型转换，必须写在类内，建议加const，没有返回值但要写返回值，没有参数

operator int ()

{

return 3;

}