C++编程实验指导

学 院 软件工程学院

年级专业 软工182

学生学号  2018081062

学生姓名  邓棋

提交日期 2019年12月2日

**实验二 类的继承练习**

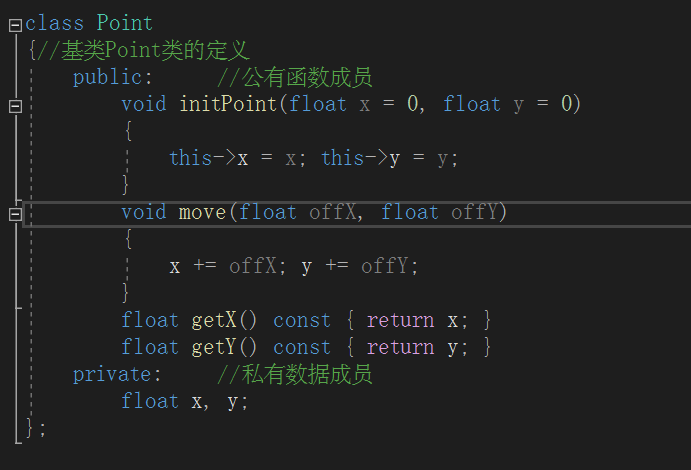
**一、实验目的**

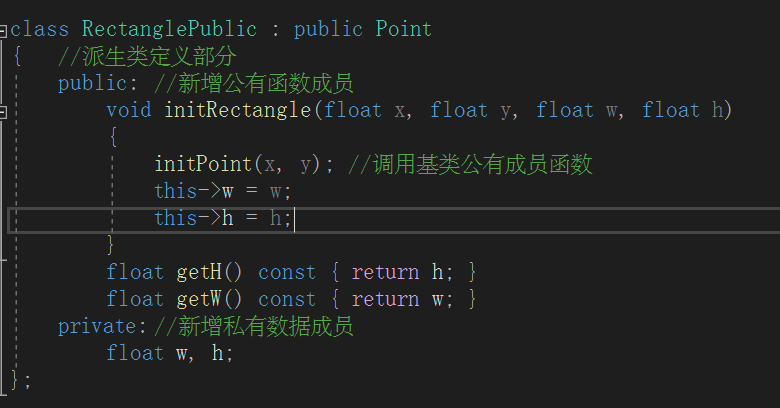
1． 掌握继承的三种方式。

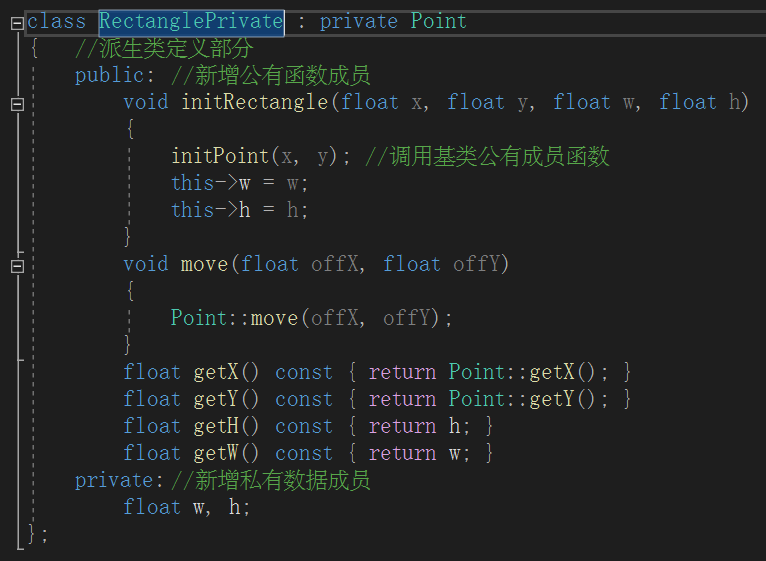
2． 掌握单继承、多继承、虚基类的基本用法。

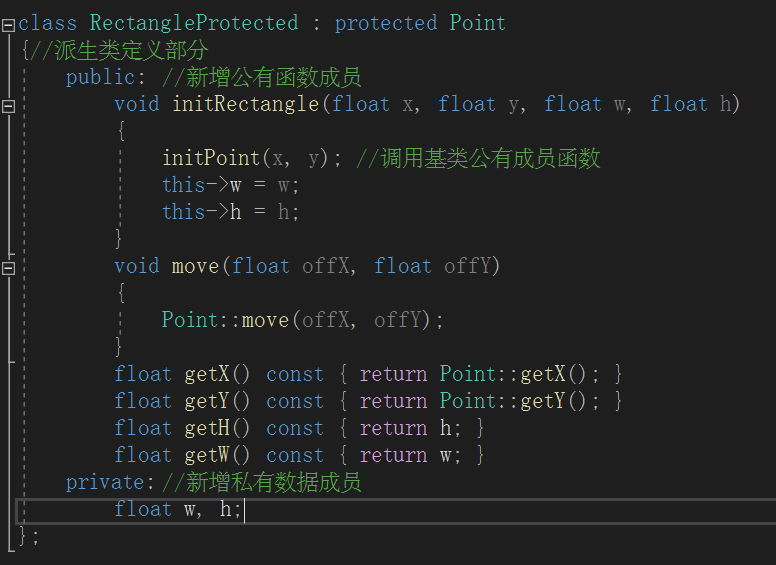
**二、实验内容**

**1．继承的三种方式**

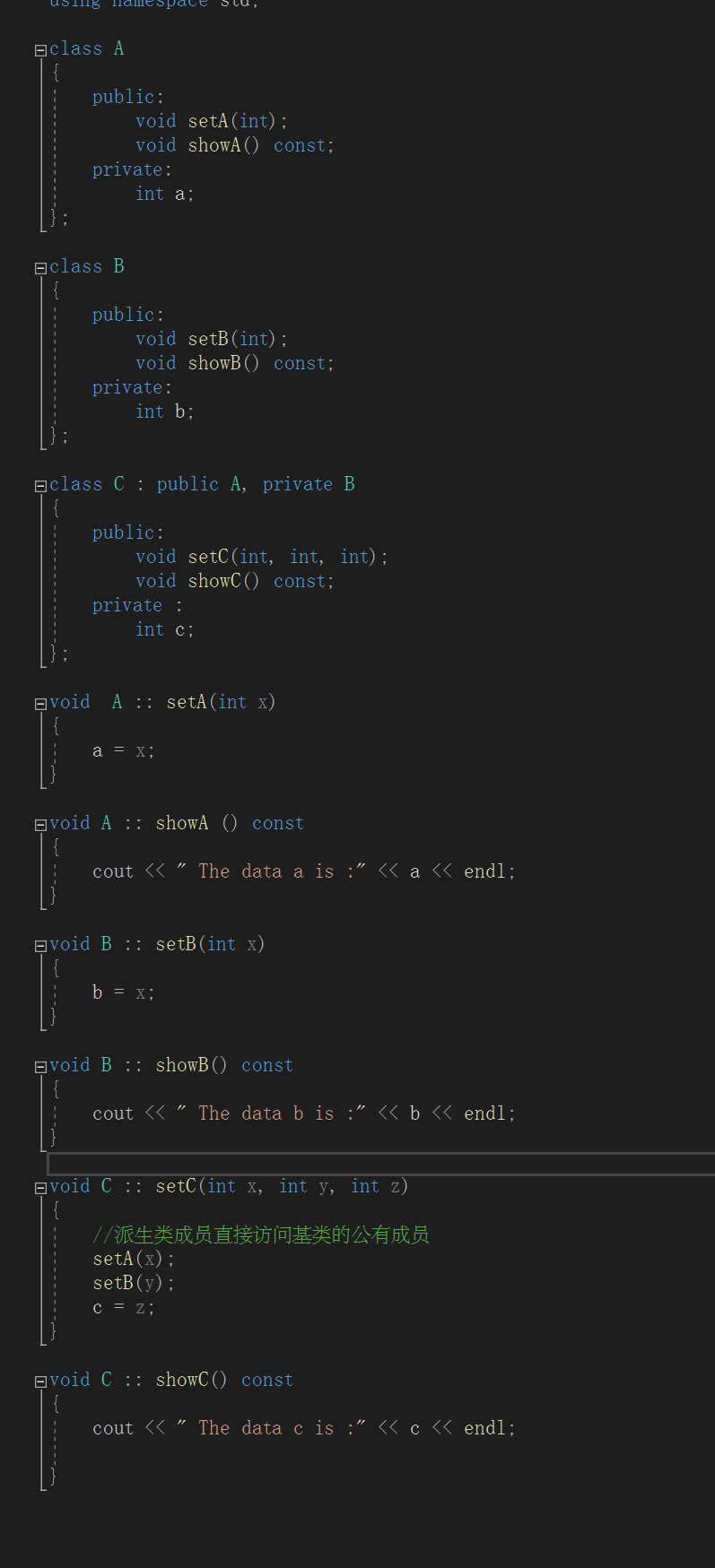
**基本父类**

**公有继承**

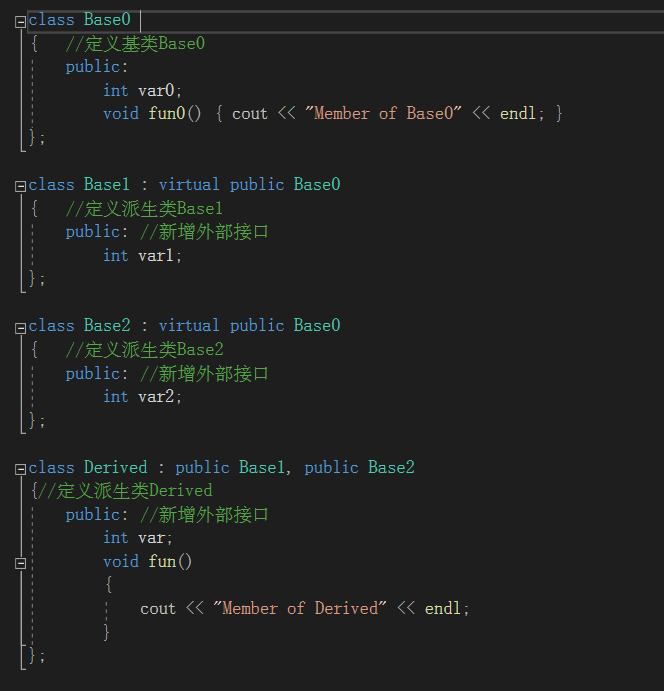
**私有继承**

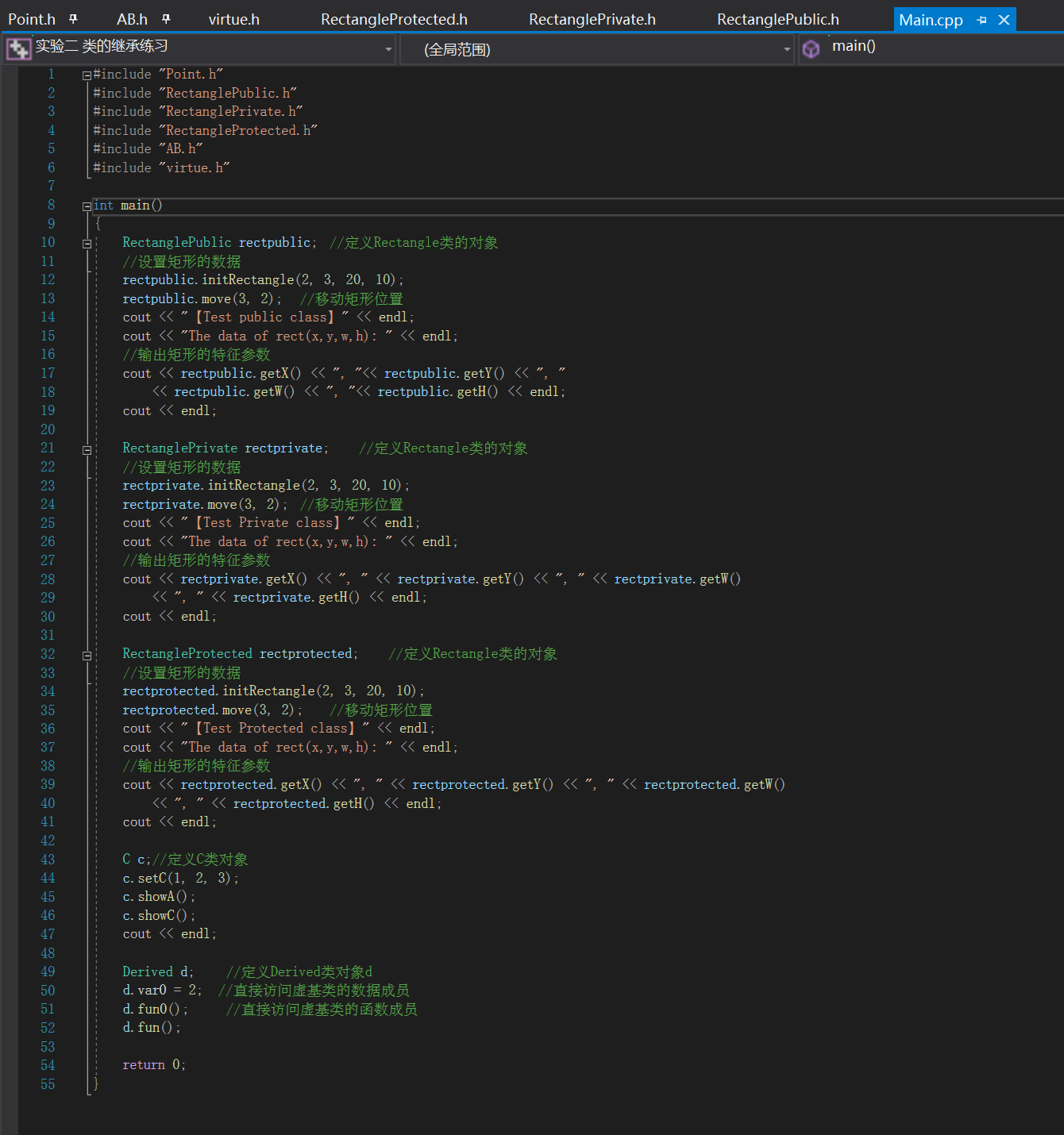
**保护继承**

**2． 多继承**

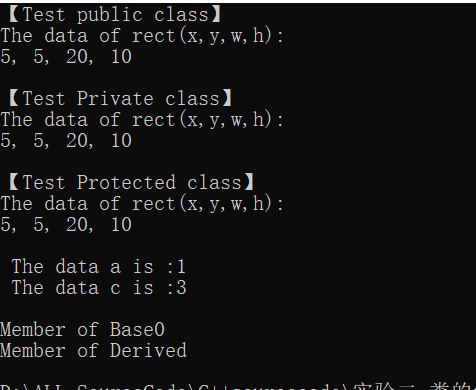


**3． 虚基类**



**实例化以上各类**

**实验结果**



**三、实验过程**

1、设计父类Point，设计Rectangle类，分别用公有、私有、保护继承三种方式继承父类Point，实现定义一个矩形，并移动它的位置。

2、设计两个类A和B，让C公有继承A，私有继承C，每个类都是确定一个参数然后打印参数，创建C类的对象，实现打印过程。

3、设计一个类，再设计两个派生类，用virtual修饰基类，再设计一个派生类公有继承自前文两个派生类，实现它对虚基类数据成员、函数成员的访问。

**四、实验心得**

为了实现代码重用，C++类中定义了继承和派生特性，保持已有类的特性而构造新类的过程称为继承，在已有类的基础上新增自己的特性而产生新类的过程称为派生。继承的目的：实现代码重用；派生的目的：当新的问题出现，原有程序无法解决（或不能完全解决）时，需要对原有程序进行改造。不同继承方式的影响主要体现在：派生类成员对基类成员的访问权限、通过派生类对象对基类成员的访问权限。C++主要有三种继承方式：公有继承、私有继承、保护继承；

公有继承：基类的public和protected成员的访问属性在派生类中保持不变，但基类的private成员不可直接访问；派生类中的成员函数可以直接访问基类中的public和protected成员，但不能直接访问基类的private成员；通过派生类的对象只能访问基类的public成员。

私有继承：基类的public和protected成员都以private身份出现在派生类中，但基类的private成员不可直接访问；派生类中的成员函数可以直接访问基类中的public和protected成员，但不能直接访问基类的private成员；通过派生类的对象不能直接访问基类中的任何成员。

保护继承：基类的public和protected成员都以protected身份出现在派生类中，但基类的private成员不可直接访问；派生类中的成员函数可以直接访问基类中的public和protected成员，但不能直接访问基类的private成员；通过派生类的对象不能直接访问基类中的任何成员。

保护继承和私有继承是很类似地，派生类的对象都不能直接访问基类中的任何成员，我们可以通过在派生类中设计函数来引用基类中的数据成员、函数成员。

虚基类主要用来解决多继承时可能发生的对同一基类继承多次而产生的二义性问题，为最远的派生类提供唯一的基类成员，而不重复产生多次拷贝，但要注意在在第一级继承时就要将共同基类设计为虚基类。建立对象时所指定的类称为最（远）派生类。

虚基类的成员是由最派生类的构造函数通过调用虚基类的构造函数进行初始化的。在整个继承结构中，直接或间接继承虚基类的所有派生类，都必须在构造函数的成员初始化表中给出对虚基类的构造函数的调用。如果未列出，则表示调用该虚基类的默认构造函数。在建立对象时，只有最（近）派生类的构造函数调用虚基类的构造函数，该派生类的其他基类对虚基类构造函数的调用被忽略。