心得总结（第三章）

电自2003 宿颢宝

类的成员有成员变量和成员函数两种。成员函数之间可以互相调用，成员函数内部可以访问成员变量。私有成员只能在类的成员函数内部访问。默认情况下，class 类的成员是私有的，struct 类的成员是公有的。可以用“对象名.成员名”、“引用名.成员名”、“对象[指针](http://c.biancheng.net/c/80/" \t "http://c.biancheng.net/view/_blank)->成员名”的方法访问对象的成员变量或调用成员函数。成员函数被调用时，可以用上述三种方法指定函数是作用在哪个对象上的。对象所占用的存储空间的大小等于各成员变量所占用的存储空间的大小之和（如果不考虑成员变量对齐问题的话）。定义类时，如果一个构造函数都不写，则编译器自动生成默认（无参）构造函数和复制构造函数。如果编写了构造函数，则编译器不自动生成默认构造函数。一个类不一定会有默认构造函数，但一定会有复制构造函数。任何生成对象的语句都要说明对象是用哪个构造函数初始化的。即便定义对象数组，也要对数组中的每个元素如何初始化进行说明。如果不说明，则编译器认为对象是用默认构造函数或参数全部可以省略的构造函数初始化。在这种情况下，如果类没有默认构造函数或参数全部可以省略的构造函数，则编译出错。对象在消亡时会调用析构函数。  
每个对象有各自的一份普通成员变量，但是静态成员变量只有一份，被所有对象所共享。静态成员函数不具体作用于某个对象。即便对象不存在，也可以访问类的静态成员。静态成员函数内部不能访问非静态成员变量，也不能调用非静态成员函数。常量对象上面不能执行非常量成员函数，只能执行常量成员函数。包含成员对象的类叫封闭类。任何能够生成封闭类对象的语句，都要说明对象中包含的成员对象是如何初始化的。如果不说明，则编译器认为成员对象是用默认构造函数或参数全部可以省略的构造函数初始化。  
在封闭类的构造函数的初始化列表中可以说明成员对象如何初始化。封闭类对象生成时，先执行成员对象的构造函数，再执行自身的构造函数；封闭类对象消亡时，先执行自身的析构函数，再执行成员对象的析构函数。const 成员和引用成员必须在构造函数的初始化列表中初始化，此后值不可修改。友元分为友元函数和友元类。友元关系不能传递。成员函数中出现的 this 指针，就是指向成员函数所作用的对象的指针。因此，静态成员函数内部不能出现 this 指针。成员函数实际上的参数个数比表面上看到的多一个，多出来的参数就是 this 指针。