习题4

三种继承方式，私有继承，保护继承，公有继承。

派生类不能直接访问基类的私有成员，但是可以通过基类提供的公有成员函数间接地访问基类的私有成员。

公有继承时，基类中的所有保护成员在派生类中还是保护成员，在派生类内可以访问这些成员，但派生类外部不能访问它们，而在下一层派生类内可以访问它们。

私有继承时，基类中的所有保护成员在派生类中变成了私有成员，在派生类内可以访问这些成员，但派生类外部不能访问它们。

通常情况下，当创建派生类对象时，首先执行基类的构造函数，随后再执行派生类的构造函数；当撤销派生类对象时，则先执行派生类的析构函数，随后再执行基类的析构函数。

当一个派生类具有多个基类时，这种派生方法称为多继承。多重继承的构造函数的执行顺序与单继承构造函数的执行顺序相同，也是遵循先执行基类的构造函数，再执行对象成员的构造函数，最后执行派生类构造函数体的原则。处于同一层次的各个基类构造函数的执行顺序，取决于声明派生类时所指定的各个基类的顺序，与派生类构造函数中所定义的成员初始化列表的各项顺序没有关系。析构函数的执行顺序则刚好与构造函数的执行顺序相反。

如果一个派生类是从多个基类派生出来的，而这些基类又有一个共同的基类，则在这个派生类中访问这个共同的基类中的成员时，可能会产生二义性。为了解决这种二义性，C++引入了虚基类的概念。若同一层次中同时包含虚基类和非虚基类，应先调用虚基类的构造函数，再调用非虚基类的构造函数，最后调用派生类构造函数。对于多个虚基类，构造函数的执行顺序仍然是先左后右，自上而下。对于非虚基类，构造函数的执行顺序仍是先左后右，自上而下。若虚基类由非虚基类派生而来，则仍然先调用基类构造函数，再调用派生类的构造函数。