多态概念：  
一个借口，多种方法，程序在运行时才决定调用的函数，它是面向对象编程领域的核心概念。多态(polymorphism)，字面意思多种形状。  
C++多态性是通过虚函数来实现的，虚函数允许子类重新定义成员函数  
子类重新定义为override，或者称为重写  
重写的话可以有两种，直接重写成员函数和重写虚函数，只有重写了虚函数的才能算作是体现了C++多态性  
多态与非多态的实质区别就是函数地址是早绑定还是晚绑定。如果函数的调用，在编译器编译期间就可以确定函数的调用地址，并生产代码，是静态的，就是说地址是早绑定的。而如果函数调用的地址不能在编译器期间确定，需要在运行时才确定，这就属于晚绑定。  
作用  
而多态的目的则是为了接口重用。（ 封装可以使得代码模块化，继承可以扩展已存在的代码）  
也就是说，不论传递过来的究竟是那个类的对象，函数都能够通过同一个接口调用到适应各自对象的实现方法。  
最常见的用法就是声明基类的指针，利用该指针指向任意一个子类对象，调用相应的虚函数，可以根据指向的子类的不同而实现不同的方法。如果没有使用虚函数的话，即没有利用C++多态性，则利用基类指针调用相应的函数的时候，将总被限制在基类函数本身，而无法调用到子类中被重写过的函数。