

Inheritance and Polymorphism

继承

继承机制使得派生类能够获得基类的成员数据和方法

Public

- 是一种接口继承，子类可以代替父类完成父类接口所声明的行为
- Public继承会保留父类中成员，包括函数和变量等的可见性不变

Protected

- 是一种实现继承，子类不能代替父类完成父类接口所声明的行为，此时子类不能自动转换为父类的接口
- 从语法角度来说，protected继承会将父类中的public可见性的成员修改成为protected可见性，相当于在子类中引入了protected成员

Private

- 是一种实现继承，子类不能替代父类完成父类接口所声明的行为，此时子类不能自动转换为父类的接口
- 从语法角度来说，private继承会将父类的public和protected可见性的成员修改成为private可见性，这样尽管子类可以调用父类的protected和public成员，但是子类的子类就不能再调用被private继承的父类的成员
- 编译器一般不会讲派生类对象转换成基类对象
- 私有继承时派生类与基类不是"is a"关系，而是意味着"Is-Implement-In-Terms-Of"（以。。。实现），如果使类D私有继承与类B，这样做是因为你想利用类B中已经存在的某些代码，而不是因为内B的对象与类D的对象之间有什么概念上的关系
- 私有继承在软件"设计"过程中毫无意义，只是在软件"实现"时才有用
- 私有继承下派生类会获得基类的一份备份，同时得到了访问基类的公共以及保护接口的权力和重写基类函数的能力。

组合

- 使用组合表示"有一个(has a)的关系"，如果在组合中需要使用一个对象的某些方法，则完全可以利用私有继承代替
- 私有继承中派生类能访问基类的protected成员，并且可以重写基类的虚函数，甚至当基类是抽象类的情况，组合不具备这些功能
- 尽可能使用组合，万不得已才使用私有继承

多态

是建立在继承上的，即动态绑定技术，其核心思想是父类对象调用子类对象的方法

统一操作作用于不同的对象，可以有不同的解释，产生不同的执行结果

编译时的多态性

- 是通过重载来实现的，对于非虚的成员来说，系统在编译时，根据传递的参数、返回的类型等信息决定实现何种操作

运行时的多态性

- 是指直到系统运行时，才根据实际情况实现何种操作，通过虚成员来实现