阅读C++编程思想第九章“多继承”，总结虚基类背后的实现机制， 阅读C++ prime plus 第14 章并总结关键内容.

《C++编程思想》第九章 多继承

多重继承（MI）的基本概念：通过继承多个基类来创建一个新类。

一、接口继承

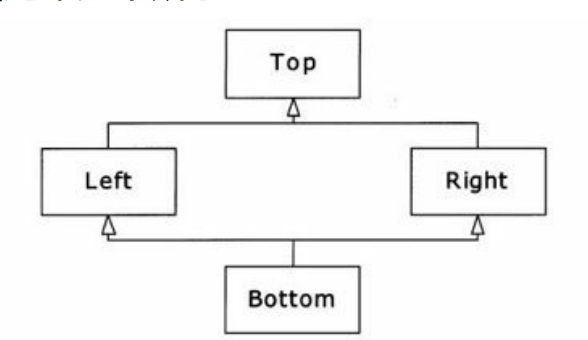
接口继承：在一个派生类接口中加入了成员函数的声明（declaration），C++中不直接支持这种使用方法。

1. 实现继承

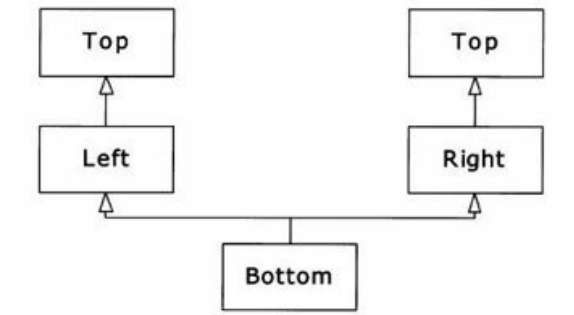
C++仅仅提供了实现继承，意味着所有的内容总是继承自基类。

例如：假定Database类，使用这个类需要一个数据库连接串，并调用Database：open（）来连接数据库，通过调用Database：close（）断开连接。

1. 重复子对象
2. 注意：在子对象和完整类型间来回转换时，要应用适当的偏移量。应先用过逻辑检查查看指针是否为零，不为零则可应用偏移量，反之则放弃使用偏移量，以防产生无效地址。
3. 菱形继承



1. 虚基类



1.使Top成为Left和Right的一个虚基类，就能让上图变成菱形继承。

Left和Right子对象各有一个指向共享的Top子对象的指针，对两个成员函数中那个子对象的所有引用都要通过这些指针来完成。

2.一个带有多重继承的派生对象必须表现出它好像有多个VPTR，它的每个含有虚函数的直接基类都有一个。

3.子对象的初始化顺序遵循如下的规则递归地进行：

（1）所有虚基类子对象，按照它们在类定义中出现的位置，从上到下、从左到右初始化。

（2）然后非虚基类按通常顺序初始化。

（3）所有的成员对象按声明的顺序初始化。

（4）完整的对象的构造函数执行。

消除二义性调用的方法：以基类名来限定函数的调用。

1. 使用多重继承的条件
2. 需要通过新类来显示两个类的公共接口。
3. 向上类型转换成为两个基类类型。

同时满足以上两个条件则使用多重继承，否则不必使用。

六、扩充一个接口

作为基类的接口缺少解决问题所必要的函数，为了解决这个问题，用户可以创建一个自己的类接口，并且采用多重继承方法产生一组新的派生类。