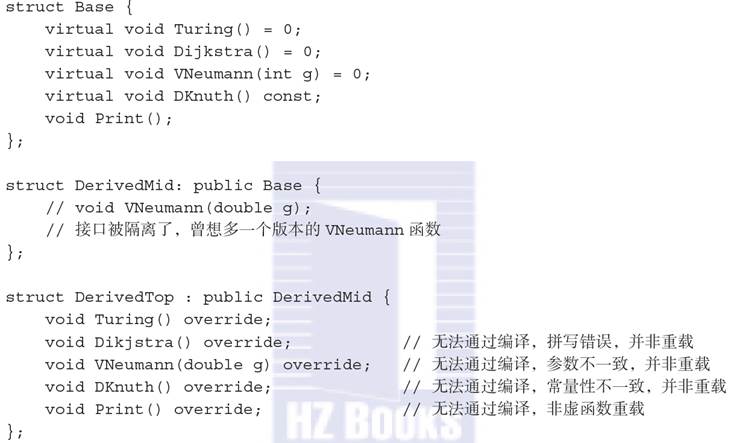
Override是在C++11 中为了帮助程序员写继承结构复杂的类型，引入了虚函数描述符override，如果派生类在虚函数声明时使用了override 描述符，那么该函数必须重载其基类中的同名函数，否则代码将无法通过编译。

示例代码：



在基类Base中定义了一些virtual的函数（接口）以及一个非virtual的函数Print。其派生类DerivedMid中，基类的Base的接口都没有重载，不过通过注释可以发现，DerivedMid的作者曾经想要重载出一个“void VNeumann(double g)”的版本。这行注释显然迷惑了编写DerivedTop的程序员，所以DerivedTop的作者在重载所有Base类的接口的时候，犯下了3种不同的错误：

函数名拼写错，Dijkstra误写作了Dikjstra。

函数原型不匹配，VNeumann函数的参数类型误做了double类型，而DKnuth的常量性在派生类中被取消了。

重写了非虚函数Print。

如果没有override修饰符，DerivedTop的作者可能在编译后都没有意识到自己犯了这么多错误。因为编译器对以上3种错误不会有任何的警示。这里override修饰符则可以保证编译器辅助地做一些检查。我们可以看到，DerivedTop作者的4处错误都无法通过编译。