**C++学习笔记——继承和动态多态**

**1. override**

C++11的新关键字。

在继承关系中，子类需要重写父类的virtual函数时，可以对子类中的该函数添加override关键字。这样，编译器会进行如下检查：

1. 父类确实存在该同名函数。

2. 父类的同名函数的返回值类型、参数列表、const性质、引用限定，都与子类的override函数完全相同。

3. 父类的同名函数是virtual函数。

只有上述三个条件都满足，才能够通过编译。

override关键字能够提升代码可读性，以及避免忘记或错误重写的情况发生——一旦出现此情形，编译器也会给出更加明确的错误提示。

**2. final**

C++11的新关键字。

可用于修饰类，表示该类不可被继承；或用于成员虚函数，表示该成员函数不可被派生类重写。注意，final不能用于非虚成员函数。

**3. 纯虚析构函数（将抽象类析构函数声明为纯虚函数）**

在动态多态中，如果需要一个抽象类作为基类，但该类本身没有合适的函数可作为纯虚函数，则可以将其析构函数声明为纯虚函数。

图形用户界面, 文本, 应用程序

描述已自动生成

注意，抽象类的纯虚析构函数仍然必须要实现。这是因为，基类的析构函数会被派生类的析构函数间接调用。如果没有实现抽象类的纯虚析构函数，则会出现连接错误。

顺带一提，将一个函数声明为纯虚函数，并不意味着它不能被实现。通常的纯虚函数不会被调用（在动态多态下，通常只有派生类的同名函数被调用），故没有必要实现。但是，虚析构函数总是会被调用，所以必须实现它。