```
//面向对象：多态-虚函数
//函数重载
// 1. 同作用域
// 2. 同名
// 3. 参数列表不同，返回值和调用约定不考虑

//函数隐藏
// 1. 作用域不同
// 2. 同名
// 3. 参数列表，返回值和调用约定不考虑

//函数覆盖 --- 指的就是虚调用
// 本质： 子类重写父类的虚函数，子类中的虚表对应的位置会被子类的对应的虚函数覆盖
// 重写含义：
// 函数声明完全相同（函数名，函数列表，返回值，调用约定）
// virtual关键字可写可不写，建议加上，增加代码可读性
// 位置含义：
// 1. 父类中虚函数声明的顺序决定了子类中虚函数的顺序
// 2. 子类新增的虚函数在父类定义的虚函数的后面
// 覆盖：
// 子类虚函数将父类对应位置的虚函数替换了。

// 1. 作用域不同
// 2. 函数声明完全相同
// 3. 至少父类中有virtual关键字。

//虚调用：
// 1. 函数本身是虚函数（virtual）
// 在类外部：调用者是指针或者引用，则会是间接调用
// 在类内部（类的成员函数中)：用类域直接访问是直接调用，用this指针是间接调用
// 在类的构造和析构中调用虚函数不会多态：
// （1）调用虚函数都是直接调用
// (2) 调用的是成员函数，通过成员函数调虚函数仍然没有多态：
// 构造&析构时，会把虚表指针改为自己类的虚表指针。

//虚析构的原因：
// delete 类对象指针时，可以有多态，能够调用到子类对象的析构函数
```

示例代码

链接：`Github`

https://github.com/yeahlife/CR32/tree/master/c%2B%2B/13virtual2/Test

文件：`TestInherit`