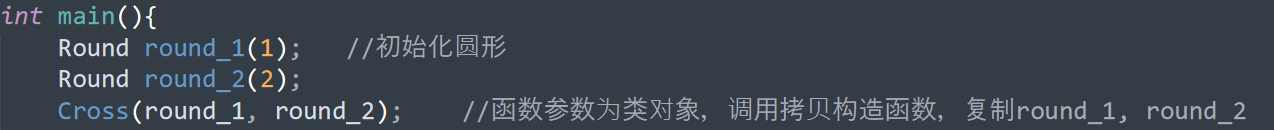
C++基础题目笔记

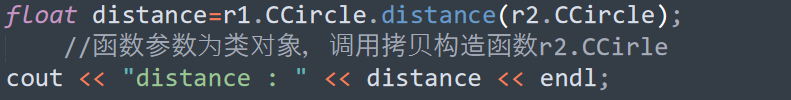
318-2018210990-薛锦隆

基础题目部分

2-1.cpp







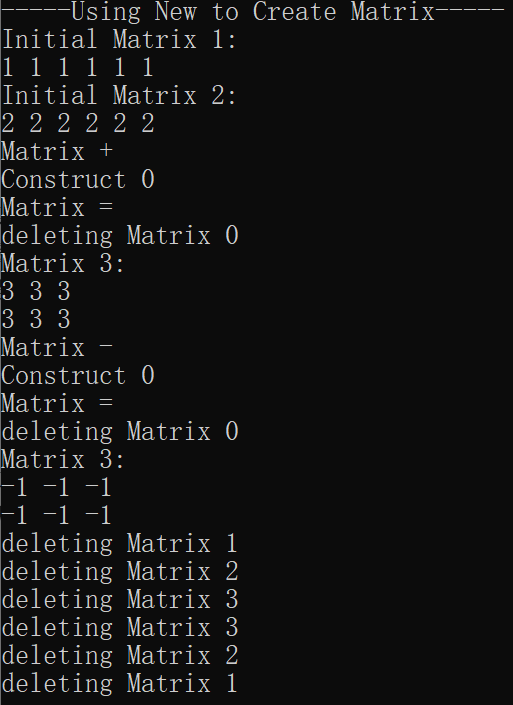
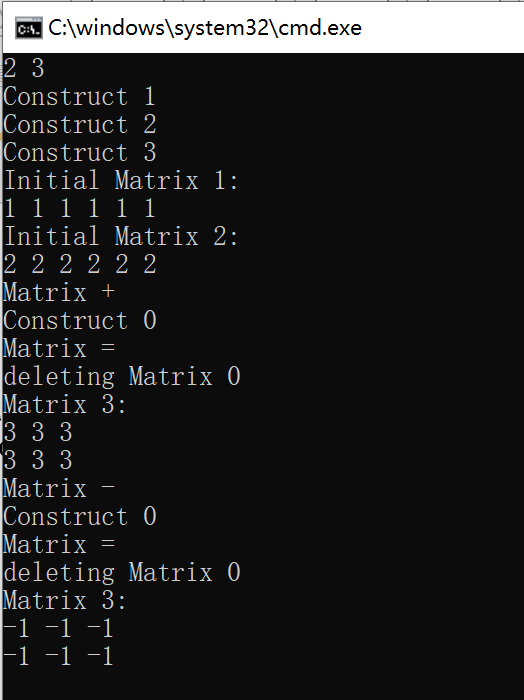
c++运行时堆栈的过程：

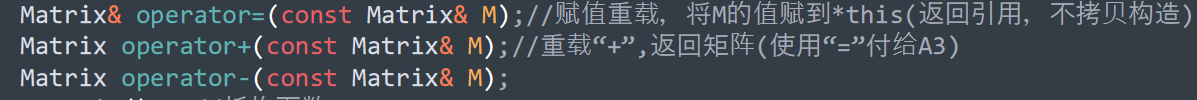
堆：初始化round\_1, round\_2（先产生Point类，后Round类）

函数调用复制后又产生round\_1,round\_2（拷贝构造）

函数参数为类对象，调用拷贝构造函数r2.CCircle

析构时，按FILO弹出

2-2.cpp

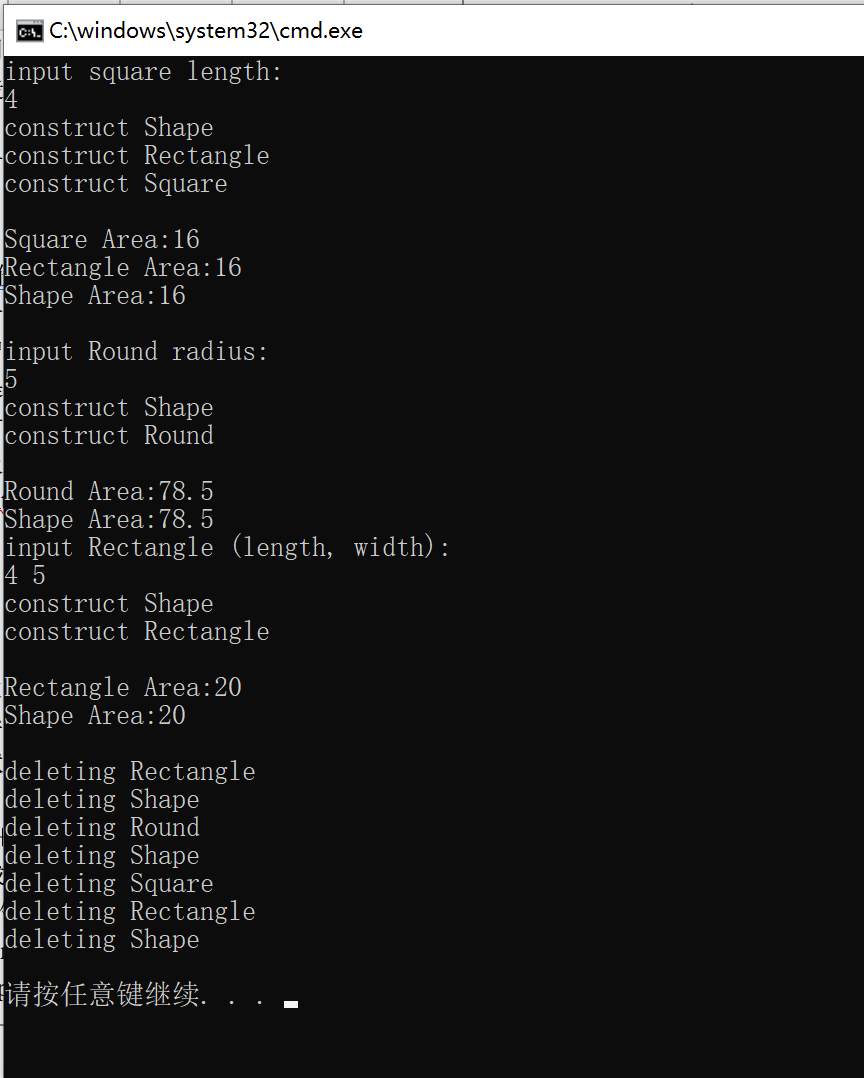


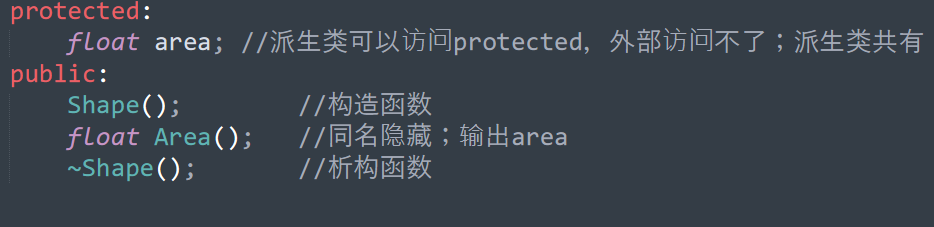
重载=, +, -

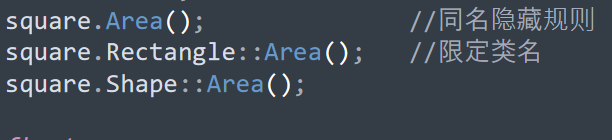
+和-返回一个新的矩阵，使得新矩阵赋值给A3，函数运行完后新矩阵被释放掉

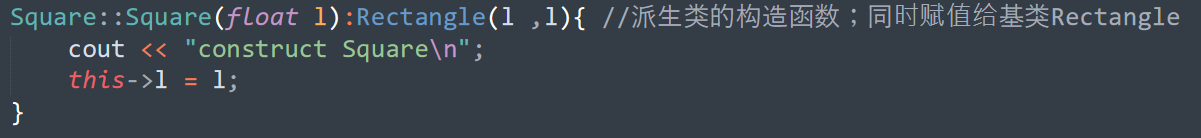
=返回自身的引用，其实是在函数中将对方的数据赋值到自己的数据中，没有拷贝构造的过程

3.cpp









派生类从基类开始构造（基类、派生类、成员对象）

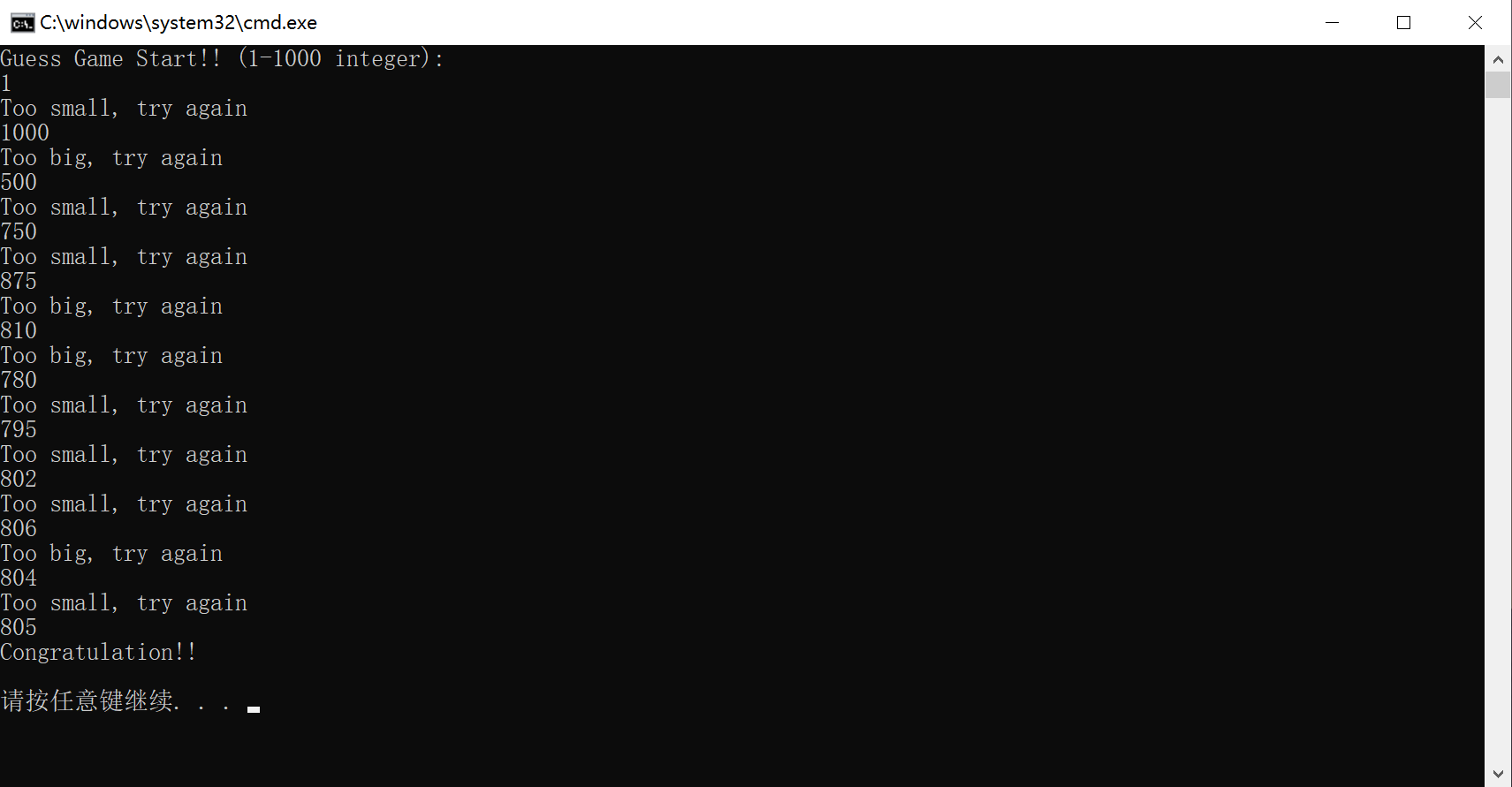
析构函数的调用次序与构造函数相反（成员对象、派生类、基类）

area：派生类可以访问protected，外部访问不了，被派生类所共有

同名隐藏规则和限定类名

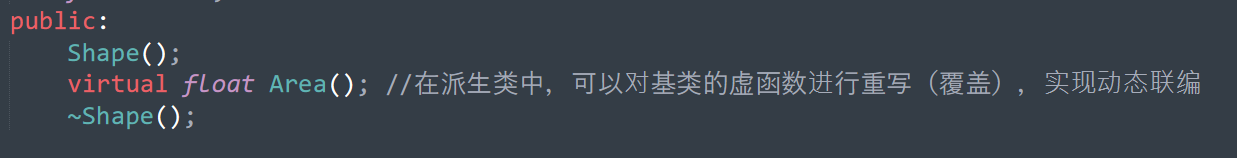
Square派生类的构造函数，对继承的基类Rectangle初始化，向基类构造函数传递参数(l, l）

4.cpp



使用cin、cout，由于平时就有使用，很熟悉。

5-1.cpp



/\* 虚函数

1.虚函数为非静态，动态绑定（在程序运行时绑定所属的类）

2.在调用对象虚函数时，根据对象所属的派生类，决定调用哪个函数实现(动态联编)

3.只需要重写派生类的特殊部分的代码，其他都从基类中继承(最大限度的重用)

/\* 抽象类

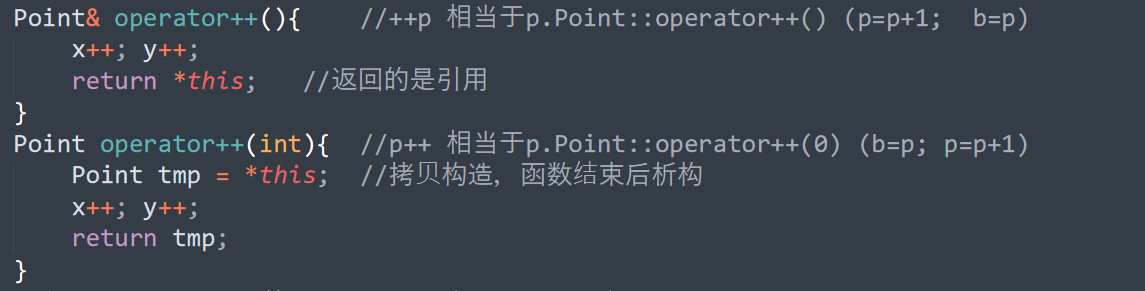
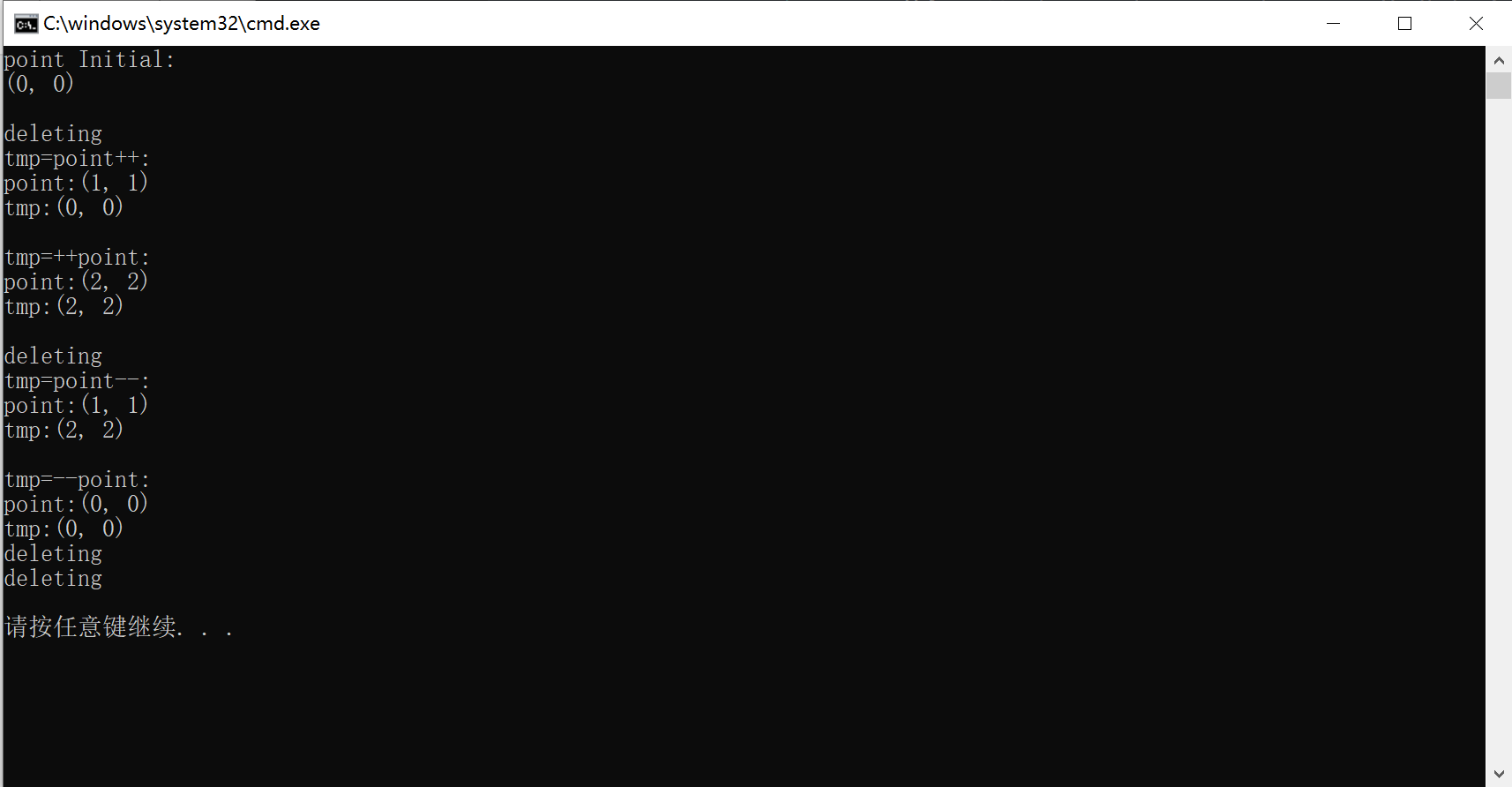
1.如果类中至少有一个函数被声明为纯虚函数，则这个类就是抽象类

2.纯虚函数, 只定义方法调用的格式，通过在声明中使用 "= 0" 来指定的

3.不能声明抽象类的对象，必须要在派生类中重写虚函数

4.抽象类的目的，是为了给其他类提供一个可以继承的适当的基类，函数留给派生类去实现

5-2.cpp



1. 对于前置一元运算符：++p 相当于p.Point::operator++()。

b=++p：p=p+1, b=p。所以返回\*this（即p），Point类的对象的引用。

2. 后置一元运算符 ++ 和 --，需要特殊处理。operator-- 被重载为成员函数，有一个整型参数(但未用）。

p++ 相当于 p.P::operator++(0)。（整形参数0没有实际作用）

b=p++：b=p; p=p+1。所以要先把p保存（赋值）在tmp对象中，再p++，最后返回tmp对象。