

深度探索C 对象模型(1) PDF转换可能丢失图片或格式，建议
阅读原文

https://www.100test.com/kao_ti2020/133/2021_2022__E6_B7_B1_E5_BA_A6_E6_8E_A2_E7_c97_133881.htm 第一章：关于对象（Object Lessons）读完这一章使我想到了一个很久以前看到的一个笑话，编写一个HELLO WORLD的程序，随着水平和职务的不一样，程序代码也随着变化。当初看时完全当作笑话来看，现在看来写此笑话的人水平不一般。如果要使你的代码能够最大限度的适应不同的运行环境，和最大限度的复用，则在设计和编写的过程中需要考虑的问题很多，因此代码已变的不在具有C语言的简洁，高效。而牺牲了这些优势换来的是更好的封装。当然如果你只是要打印Hello World则不必这样做了。以C 的思维方式解决问题，对于对C语言已经很熟悉的人来说会很不能适应。需要一段时间来适应，不然会将代码写的似是而非。而且不能邯郸学步，必须从思想上彻底的C（OO），如果只是依葫芦画瓢，那结果很可能是用C 的语法编写C式的程序。本人曾经犯的典型的低级的错误之一，就是无意识的一个类无限制的扩充，完全没有考虑到类的多层结构（基类-派生类），需要属性或方法便在类中增加，虽然也用到了多态、重载等一些OO的设计方式，但最后这个类庞大无比，除了在当前系统中任劳任怨的工作外，一点复用的可能都没有，如果另一个系统还需要一个类似的东西，那只能重新设计实现一个新的类。并且最致命的是在维护更新时带来得麻烦，需要不断全部编译不说，而且代码在用了大量注释后，在过一段时间读起来也是一件重脑力劳动。及失去了C的简洁清晰和高效，也不完全具备C 的面向对象

的特性。这根本不能叫C程序。（我想有时间重写一下以前代码也会有很多收获，温故而知新吗）C和C++在编程思想上是相互矛盾的。这也就是说如果你想学C++，完全可以不学C，只需要一本好书和一个不太笨的大脑再加上努力就可以了，如果你已有C的经验在一定的情况下反而会捣乱。本章是对对象模型的一个大略浏览。既然我们选择了C++而不是C作为开发工具，那我们的编程思想也应该转为C++的，而不能再延续C的Procedural方式。我们必须学会C++的思考方式。采用抽象数据类型或用一个多层的class体系对数据以及数据处理函数进行封装，只有摆脱C程序的使用全局数据的惯性，才能充分发挥出C++对象模型的强大威力。在C++中有两种数据成员static和nonstatic，以及三种成员函数static、nonstatic和virtual。C++对象模型对内存空间和存取时间做了优化，nonstatic的数据成员被置于类对象之内，而static数据成员被置于类对象之外。static和nonstatic成员函数被放在类对象之外。而virtual函数是由类对象的一个指向vtbl（虚函数表）的指针vptr来进行支持。而vptr的设定和重置由类的构造函数、析构函数以及copy assignment运算符自动完成。我们设计的每一个类几乎都要有一个或多个构造函数、析构函数和一个Assignment运算符。他们的作用是构造函数产生一个新的对象并确定它被初始化。析构函数销毁一个对象并确定它已经被适当的清理（避免出现内存泄露的问题），Assignment运算符给对象一个新值。这是第一章的第一部分，由于雷神最近几天在做模式小组的主页，时间周转不开了。本想写完整个一章再发，考虑一下还是先发一部分吧。原因有2。1、第一章的后半部可能又要拖上10天半个月的。2、笔记实在难写，我不愿意将笔记做成

将书上的重点再抄一边，而是喜欢尽量将自己的理解描述出来，谁知第一章便如此的难以消化，已经反复读了3遍，还是有些夹生。所以本着对大家和自己负责的态度，雷神准备再看它3遍在说。突然发现自己的C还差的很远，好可怕呀。

100Test 下载频道开通，各类考试题目直接下载。详细请访问
www.100test.com