第四章 数据抽象

1、动态内存分配

Dynamic

程序内存的分配 { 动态 在运行时随用随分
静态 在编译时就分配完毕

Static

C中动态分配内存的函数 → malloc / free

CPP中 动态分配内存的函数 → new / delete

new 函数 → 为指定的数据类型(或数组等结构)分配需要的内存，并返回内存的首地址

注意：千万别忘记释放

如果是自行定义的类对象，将调用析构函数

delete 函数 → 释放由new函数分配的内存

但其变量本身依旧存在，可继续调用

※应注意，在申请/释放连续的内存空间(如数组)时，使用 new[] 和 delete[]

※应注意，new申请时可能由于内存空间已满而失败

动态分配内存是在堆上进行的

此时请引入判断 p == nullptr [指针是否为空]

2、C库 C-Library

C中的库一般包含了多个有效的数据结构及其所对应的函数等

存放在头文件中，并在程序一开始调用

请参照，教材 第四章中 CStash库 和 Stack库 理解

储藏堆　　链栈

但C库也有一些不足：

① 某个结构的操作函数定义在全局，理论上其他结构也能调用

② 对象中的所有成员变量都是公开的，可被任意修改

3、CPP中的 类 class →体现封装性

即将某个结构及其操作函数封装在一起使用，使函数成为该类的成员函数

注意，调用时必须以对象(.符)来调用

或通过强调作用域(::符)来调用

一个类可拆分两部分 → 有哪些属性 → 成员变量

→ 有哪些操作 → 成员函数

同类的对象可以相互赋值