

## 第六部分 对象的初始化和清理

### constructor

#### 1. 利用构造函数初始化

定义在类中, 是类的成员函数

负责对对象的成员变量进行初始化

要求在编译过程中, 被编译器自动调用

对象无法在主函数中调用

标准格式

public:

类名 (参数表) { 操作 }

① 无返回类型

② 函数名与类名一致

注意: 当一个类无人为定义构造函数时,

缺省构造函数

编译器会自动添加一个无参无操作的构造函数

可以利用函数重载实现 同一个类内多个构造函数,  
编译器会根据参数的不同调用正确的函数

满足不同的  
初始化  
需要

### destructor

#### 2. 利用析构函数消亡对象

定义在类中, 是类的成员函数

负责在对象消亡时对对象内资源进行释放

同样由编译器自动调用

但内存尚未重写  
数据还存在

可以 手动调用析构函数

标准格式

对象不会消失  
释放操作

public:

~类名 ( ) { 操作 }

① 无返回类型 ② 无参数 ③

※ 基本数据结构 会由 编译器 自动回收  
这样的类 可以不写析构函数

同一类中析构函数是唯一的

但返回类型是指针

※ 注意: new 和 delete 可用来 创建/销毁对象  
过程中会调用 构造函数 和 析构函数  
[而 malloc 和 free 只会单纯的开辟和释放内存]

可能造成  
循环调用

### 3. 聚集类型 的初始化和赋值

数组

对于 某个类 的 对象数组

在 初始化 时可用 列举方式 调用 构造函数 初始化

Point[2] = { Point(1,2), Point(3,4) }

构造函数, 返回匿名对象

对于 某个类 的 对象数组 中的 元素

在 赋值 时 可以构造匿名对象 来赋值

Point[0] = Point(3,4)

匿名对象, 创建 → 赋值给 Point[0]

匿名对象被消灭