

# 第五部分 隐藏实现 封装性

## 1. 访问权限

设置访问权限 并进行封装的原因:

- ① 不希望用户 访问某些内容, 可进行权限设置
- ② 某些部分无需由用户关心

对象私有部分 [无法由对象直接调用] 对外公共接口 [即可以通过对象直接调用]

private: 一般为

↓  
以对象、成员形式

public: 一般为

↓  
以对象、成员形式

→ 数据结构

→ 成员操作函数

→ 成员属性变量

可以在类的内部实现定义

类中不写, 默认到 private 域 也可以在类的外部

可由该对象的 public 部分的函数调用

通过 :: 符 定义

(给定作用域)

double Cpoint::Distance( )

属性 作用域: 函数名(参数) 值

相同类的两个对象 → 其成员属性变量是独立存储的  
其成员函数是共享使用的

## 2. 友员函数 friend 关键字

一种破坏封装性的方式

通过在类的内部 加关键字 friend 来声明/定义函数

是全局函数 ← 友员函数不属于类的成员

但可以直接访问类的私有域

可以在多个类内定义/声明相同的友元函数  
一般跨类