

- 1 设计类 抽象类
 - 1.1 `class` 类名{
 - 1.2 `public` 公共权限
 - 1.3 设置 成员属性
 - 1.4 设置 成员函数
 - 1.5 }
 - 1.6 使用类 创建对象 实例化对象
 - 1.7 类名 对象名
 - 1.8 通过对象 来设置属性 调用成员函数
 - 1.9 类和对象 关系???
 - 1.9.1 类是对对象的抽象
 - 1.9.2 对象是对类的实例
- 2 内联函数 解决宏缺陷问题
 - 2.1 给编译器一个建议，加上关键字，编译器不一定按照内联处理
 - 2.2 不加关键字，也许编译器还偷摸的给你加 `inline`
 - 2.3 成员函数 默认加上关键字
 - 2.4 函数声明加了关键字，函数实现也要加 `inline` 关键字
- 3 函数默认参数
 - 3.1 参数可以有默认值
 - 3.2 如果有一个位置有了默认值，那么从这个位置开始，从左往右都必须有默认值
 - 3.3 函数声明和实现 只能有一个有默认值
- 4 函数占位参数
 - 4.1 `void func(int)` 占位参数 调用时候必须要提供这个参数
 - 4.2 占位参数也可以有默认值
 - 4.3 c 语言中没有默认参数 和 占位参数
- 5 函数重载的基本语法
 - 5.1 函数名称相同 又在同一个作用域下
 - 5.2 函数参数个数不同、类型不同、顺序不同都可以满足重载条件
 - 5.3 函数的返回值可以作为函数重载条件吗？ 不可以
 - 5.4 当函数重载碰到了函数默认参数 要注意避免二义性
- 6 `extern C` 浅析
 - 6.1 解决了 C++文件中调用 C 语言的代码
 - 6.2 `ifdef __cplusplus extern "C" {`
 - 6.3 }
- 7 C++语言的封装
 - 7.1 将属性和行为作为一个整体，来表示生活中具体的事物
 - 7.2 有访问权限
 - 7.3 `class` 和 `struct` 唯一区别 默认权限不同
 - 7.3.1 `class` 默认是 `private`
 - 7.3.2 `struct` 默认是 `public`
 - 7.4 `public` 是类内类外都可以访问到
 - 7.5 `protected` 类内可以，类外不可以
 - 7.6 `private` 类内可以，类外不可以
- 8 建议将所有成员属性设置为私有

8.1 自己提供公共的对外接口来进行 `set` 或者 `get` 方法访问