1. 设计一个类，求圆的周长
   1. class + 类名 { 成员变量 成员函数 }
   2. 公共权限 public
   3. 设计成员属性
      1. 半径 int m\_R
   4. 设计成员函数
      1. 获取圆周长 int calculateZC(){}
      2. 获取圆半径 int getR()
      3. 设置圆半径 void setR()
   5. 通过类创建对象过程 称为 实例化对象
2. 内联函数
   1. 内联函数引出---宏缺陷
      1. 宏缺陷：
         1. 必须要加括号保证运算完整
         2. 即使加括号，有些运算依然与预期结果不符
      2. 普通函数不会出现缺陷
   2. C++提供 内联函数代替宏函数
   3. 关键字 inline
   4. 在函数声明和实现中同时加入关键字 inline 才称为内联
   5. 在成员函数前 都隐式加了关键字inline
   6. 有些特殊情况下 ，写了关键字inline也不会按照内联方式处理
      1. 出现循环
      2. 出现判断
      3. 函数体过于庞大
      4. 对函数进行取地址
   7. 总结： 内联函数只是给编译器一个建议，但是编译器不一定接受这个建议，好的编译器会自己给短小的函数前加上关键字inline
3. 函数的默认参数和占位参数
   1. 默认参数
      1. 可以给函数的形参添加默认值
      2. 语法 形参 类型 变量 = 默认值
      3. int func(int a, int b = 10 , int c = 10)
      4. 注意事项 ，如果有一个位置有了默认参数，那么从这个位置起，从左到右都必须有默认值
      5. 函数的声明和实现 只能有一个 提供默认参数，不可以同时加默认参数
   2. 占位参数
      1. 只写一个类型进行占位，调用时候必须要传入占位值
      2. void func2(int a , int = 1)
      3. 占位参数也可以有默认值
4. 函数重载
   1. 满足条件
      1. 同一个作用域下
      2. 函数名称相同
      3. 函数参数个数、类型、顺序不同
   2. 函数的返回值 不可以作为重载条件
   3. 注意事项
      1. 加const和不加const的引用可以作为重载条件
      2. 函数重载碰到默认参数 注意避免二义性出现
5. extern C 浅析
   1. 用途：在C++中调用C语言文件
   2. C++中有函数重载，会对函数名称做修饰，导致调用C语言的函数链接失败
   3. 利用extern C可以解决问题
      1. 方法1：
         1. 在C++代码中加入
         2. 告诉编译器 show函数用C语言方式 做链接
         3. //extern "C" void show();
      2. 方法2:
         1. 在C语言的头文件中加入6行代码
         2. #ifdef \_\_cplusplus // 两个下划线 \_\_ c plus plus
         3. extern "C" {
         4. #endif
         5. #ifdef \_\_cplusplus // 两个下划线 \_\_ c plus plus
         6. }
         7. #endif
6. 封装
   1. C语言的封装
      1. 缺陷 将属性和行为分离
   2. C++语言的封装
      1. 将属性和行为作为一个整体，来表现生活中的事物
      2. 将属性和行为 加以权限控制
   3. 访问权限
      1. 公共权限 public 类内 类外 都可以访问
      2. 私有权限 private 类内可以访问 类外不可以访问
      3. 保护权限 protected类内可以访问 类外不可以访问
   4. class 默认权限 私有权限 而 struct默认权限是 公共权限
7. 尽量将成员属性设置为私有
   1. 自己可以控制读写权限
   2. 可以对设置内容 加有效性验证