

- 1 关系运算符重载
 - 1.1 自定义数据类型 不会内部做 比较 `==` `!=`
 - 1.2 所以要重载 `==` `!=`
- 2 函数调用运算符重载
 - 2.1 `()` 仿函数 对象`()` 看似像函数调用
 - 2.2 `MyAdd()` 匿名对象
- 3 不要重载 `&&` 和 `||`
 - 3.1 本身有短路特性我们无法去实现这种特性
 - 3.2 所以不要重载
- 4 强化训练 一字符串封装
 - 4.1 `cout` 输入 自定义的字符串
 - 4.2 `cin` 让用户输入字符串内容
 - 4.3 重载 `=` 运算符
 - 4.4 重载 `+` 运算符
 - 4.5 重载 `[]` 运算符
 - 4.6 重载 `==` 运算符
- 5 继承的引出
 - 5.1 网页 很多公共部分
 - 5.2 导致实现时候有很多重复的代码
 - 5.3 引出继承, 基类 (父类) 公共网页
 - 5.4 具体子类 实现不同的内容
 - 5.5 语法 `class 子类 :继承方式 父类`
- 6 继承方式
 - 6.1 不管公有继承 保护 还是私有 基类中的私有属性 , 都不可以继承下去
 - 6.2 公有继承
 - 6.2.1 父类中的 `protected` 在子类中是 `protected`
 - 6.2.2 父类中的 `public` 在子类中是 `public`
 - 6.3 保护继承
 - 6.3.1 父类中的 `protected` 在子类中是 `protected`
 - 6.3.2 父类中的 `public` 在子类中是 `protected`
 - 6.4 私有继承
 - 6.4.1 父类中的 `protected` 在子类中是 `private`
 - 6.4.2 父类中的 `public` 在子类中是 `private`
- 7 继承中的对象模型
 - 7.1 子类会继承父类中所有的内容 , 包括了 私有属性
 - 7.2 只是我们访问不到, 编译器给隐藏了
 - 7.3 `cd /d 1` `reportSingleClassLayout` 类名 文件名
- 8 继承中的构造和析构顺序
 - 8.1 子类创建对象时, 先调用父类的构造, 然后调用自身构造
 - 8.2 析构顺序与构造顺序相反
 - 8.3 子类是不会继承父类的构造函数和析构函数
 - 8.4 补充内容, 如果父类中没有合适默认构造, 那么子类可以利用初始化列表的方式

显示的调用父类的其他构造

- 9 继承中的同名处理
 - 9.1 成员属性 直接调用先调用子类，如果想调用父类 需要作用域
 - 9.2 成员函数 直接调用先调用子类，父类的所有版本都会被隐藏，除非显示用作用域运算符去调用
- 10 继承中静态成员的处理
 - 10.1 类似非静态成员函数处理
 - 10.2 如果想访问父类中的成员，加作用域即可
- 11 多继承的概念以及问题
 - 11.1 `class A : public B1, public B2,....`
 - 11.2 引发二义性问题
 - 11.3 想解决二义性问题，就需要通过作用域来进行区分
- 12 菱形继承问题以及解决
 - 12.1 解决问题利用虚基类
 - 12.2 `sheepTuo` 内部结构
 - 12.2.1 `vbptr` 虚基类指针
 - 12.2.2 指向一张 虚基类表
 - 12.2.3 通过表找到偏移量
 - 12.2.4 找到共有的数据
 - 12.3 内部工作原理 （可以不掌握）