面向对象程序设计

复习

2021.11.8

内容

C++的扩展（重点是引用）	第3章
类与对象	第9章
类与对象（四大函数）	第10章
构造函数与赋值运算	第10章
运算符重载	第12章
静态成员与友元	第11章
继承与多态	第13章
单向链表	第8章
模板	第9章

课程

什么是oop?

类、对象、消息

抽象、封装、继承、多态

程序=过程+调用

程序=对象+消息

现实事物=属性+行为

抽象事物=数据+函数

C++在非面向对象方面的扩充

函数原型　new和delete　输入输出

内联函数　引用　局部变量说明

函数的默认值　函数模板　const修饰符

函数重载

面向对象

结构体和类的区别　类和类模板

四大函数　运算符重载

静态成员　友元

类的组合　类的继承与派生

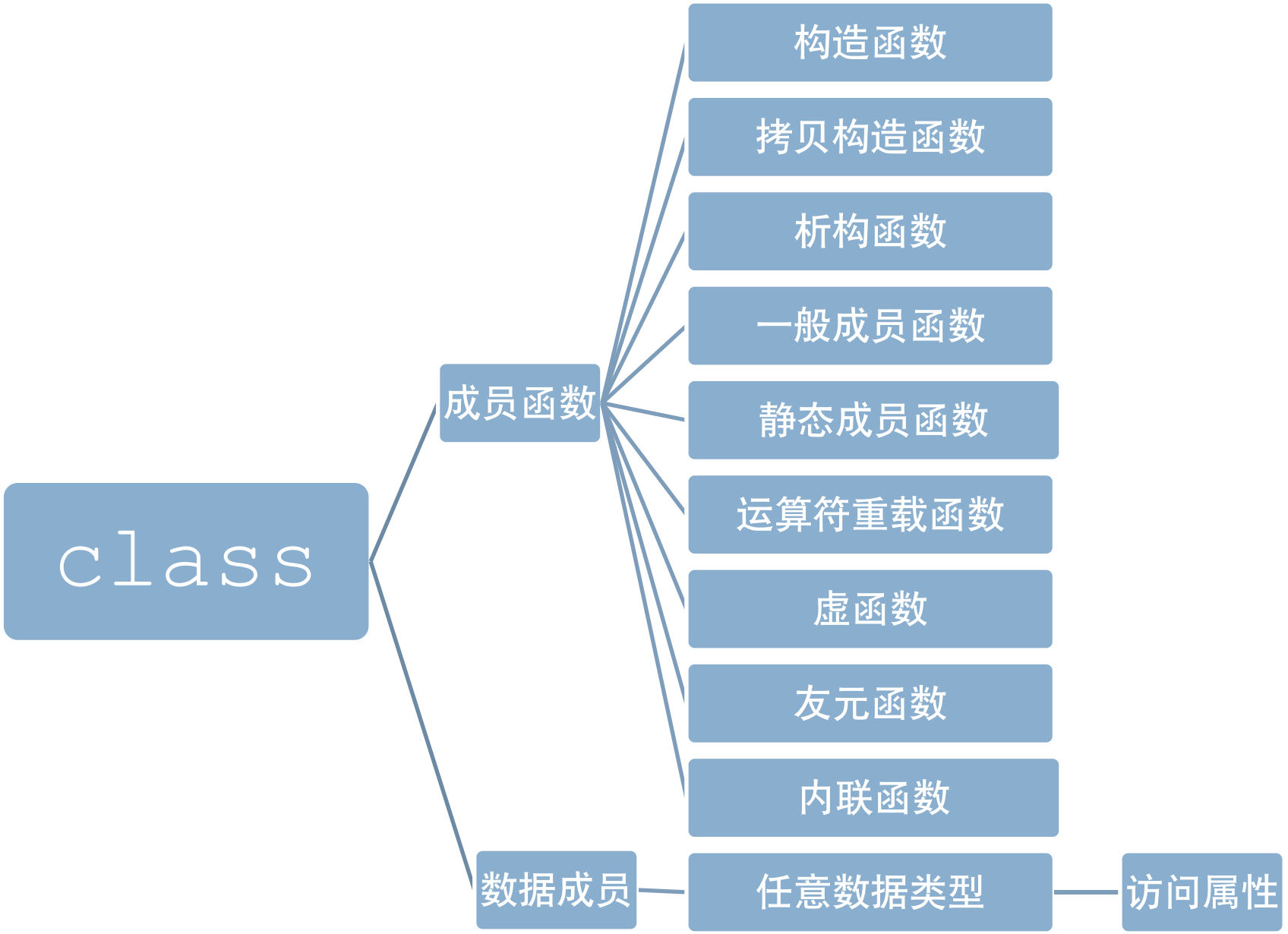
多态　多重继承

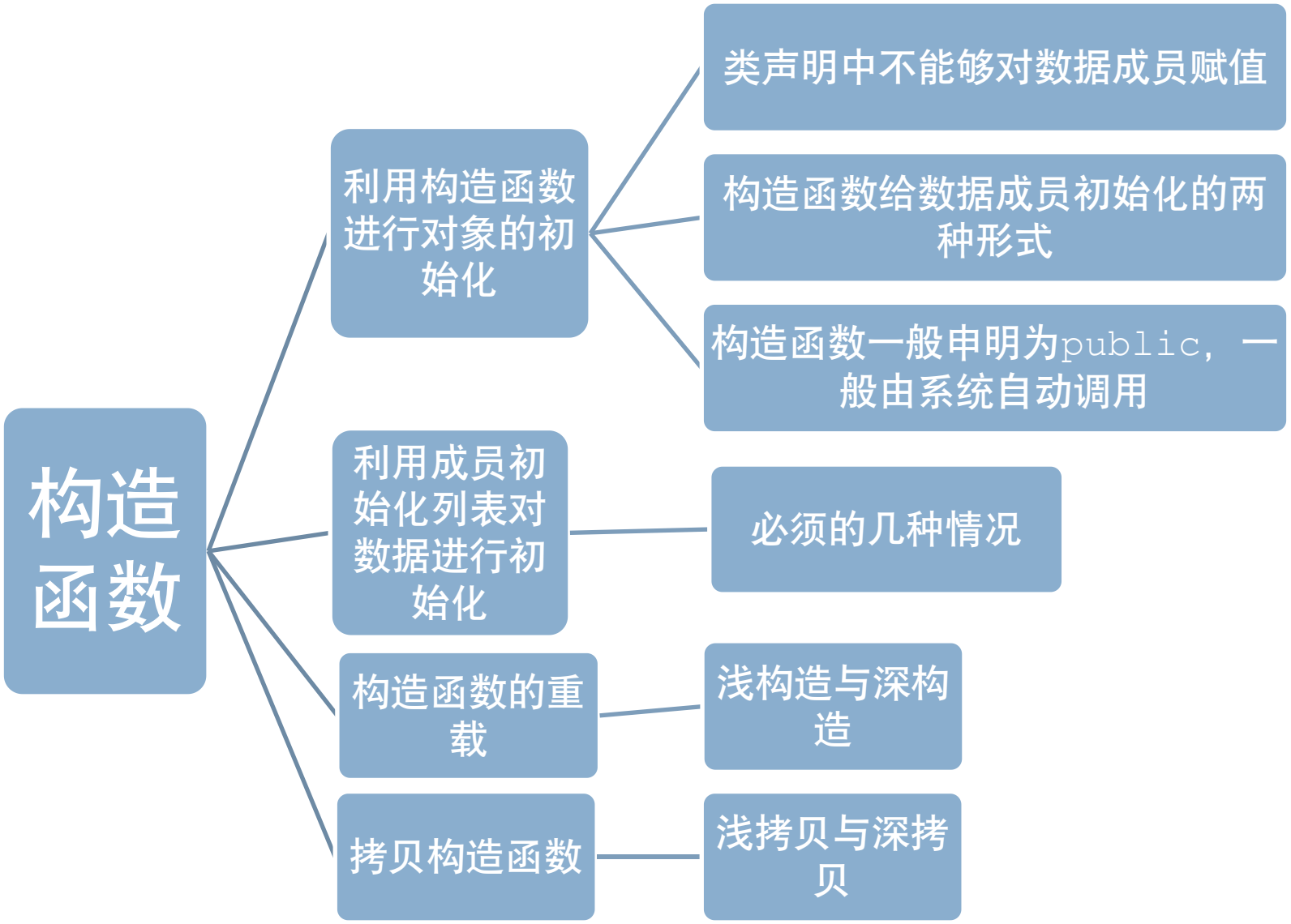
引用 &

- swap的理解
- 在函数的参数传递和返回类型设计时，注意与指针和值传递的区别，牢记，不创建新对象！！
- 函数原型设计及函数中调用时与指针的不同

结构体与类

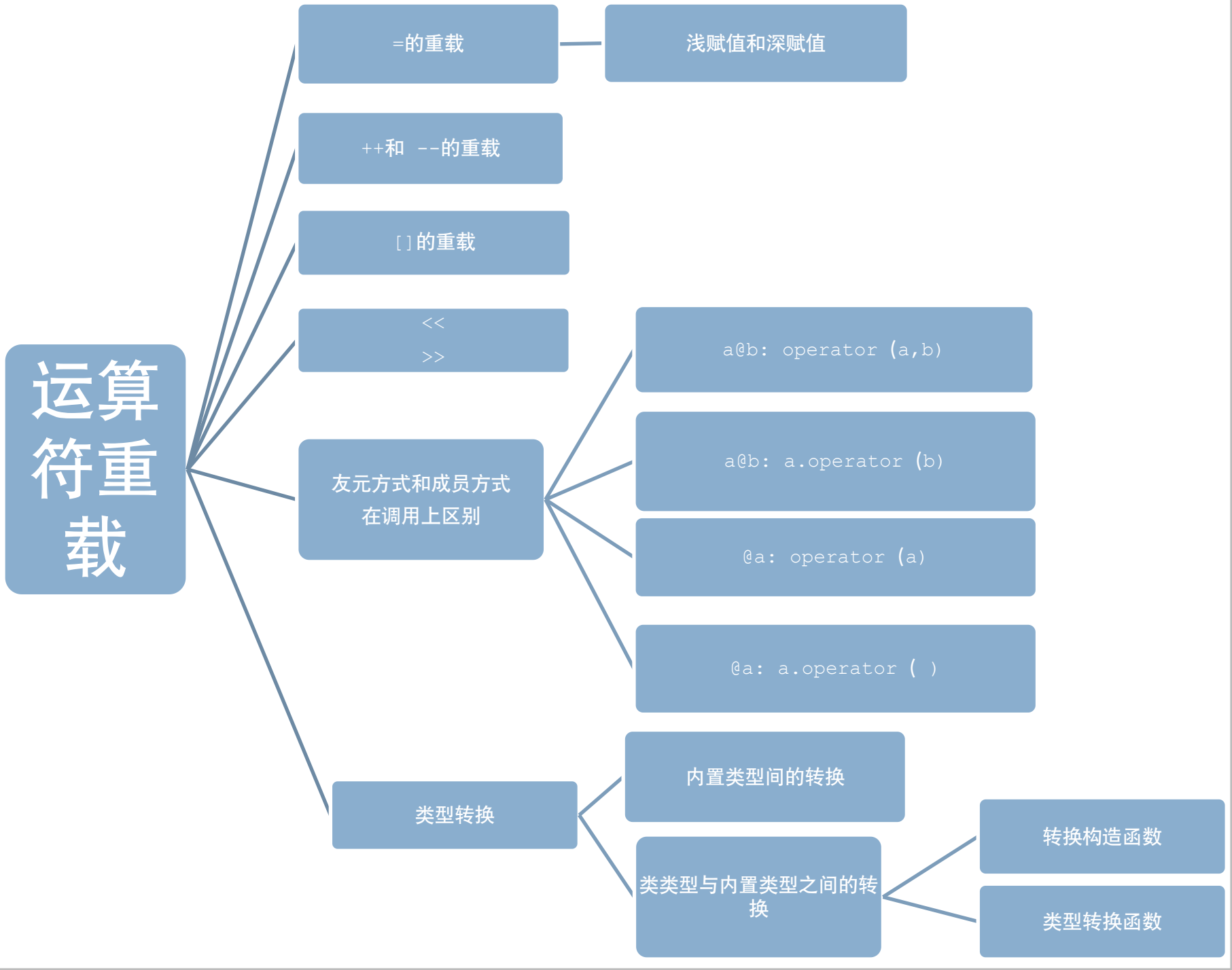
- 类是结构体的补充
- 均体现C++类型概念的自由性
- public、protected和private
- 成员函数的两种定义方式
- 对象的两种定义方式
- 对象成员的访问方式
- 数据成员和成员函数的定义

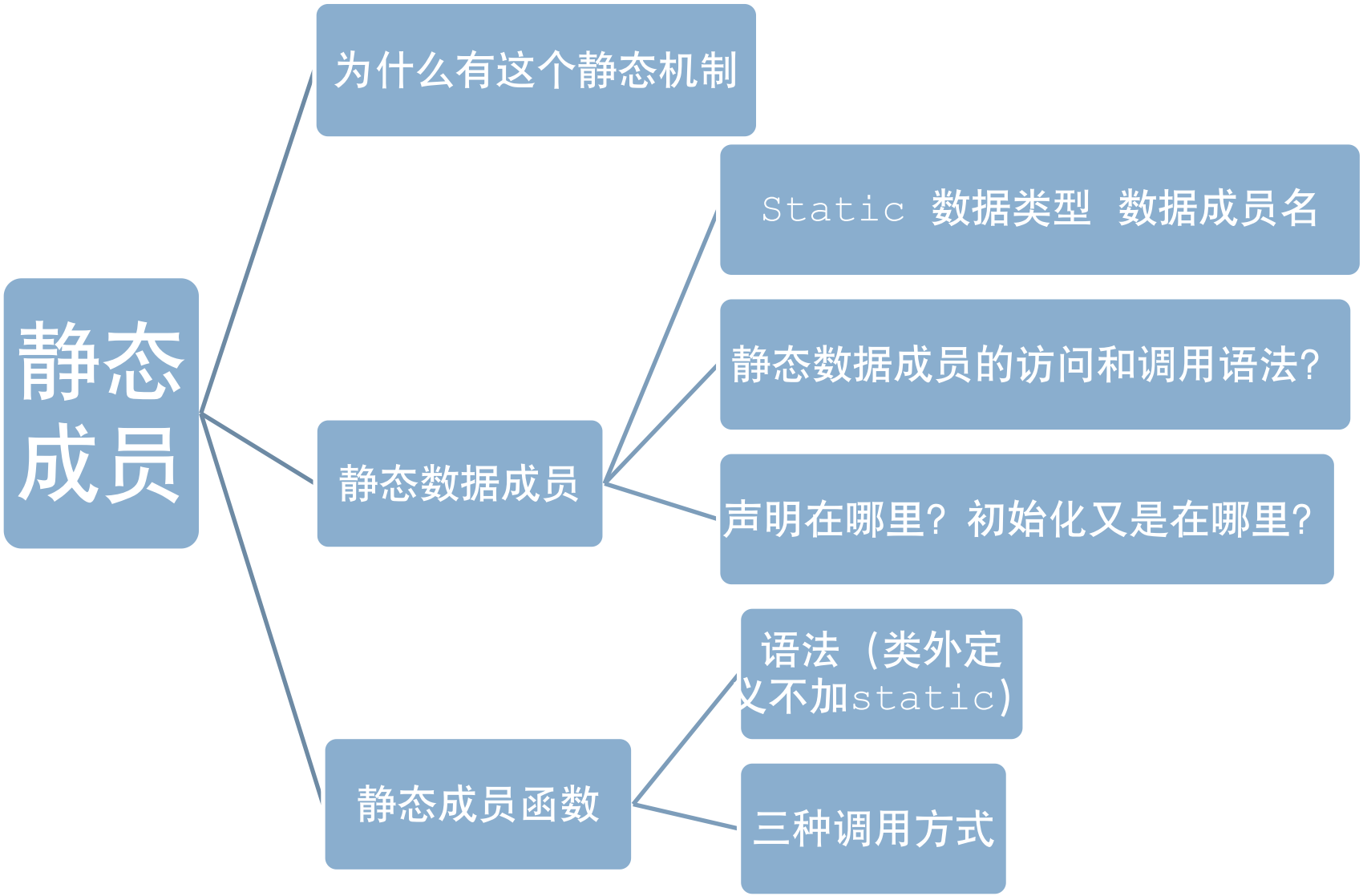

class
成员函数
构造函数
拷贝构造函数
析构函数
一般成员函数
静态成员函数
运算符重载函数
虚函数
友元函数
内联函数
数据成员
任意数据类型
访问属性

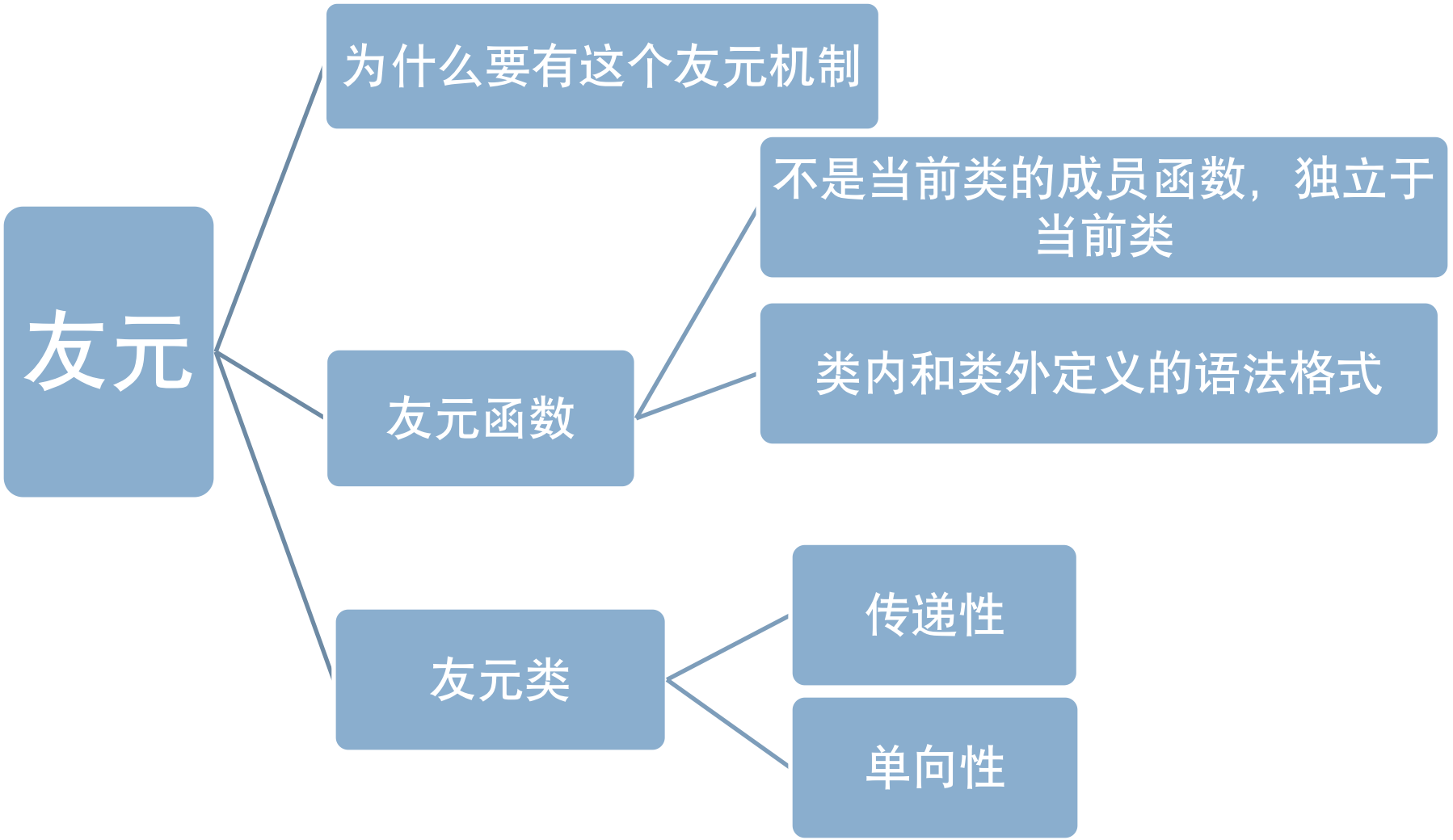

构造函数
利用构造函数进行对象的初始化
类声明中不能够对数据成员赋值
构造函数给数据成员初始化的两种形式
构造函数一般申明为public，一般由系统自动调用
利用成员初始化列表对数据进行初始化
必须的几种情况
构造函数的重载
浅构造与深构造
拷贝构造函数
浅拷贝与深拷贝

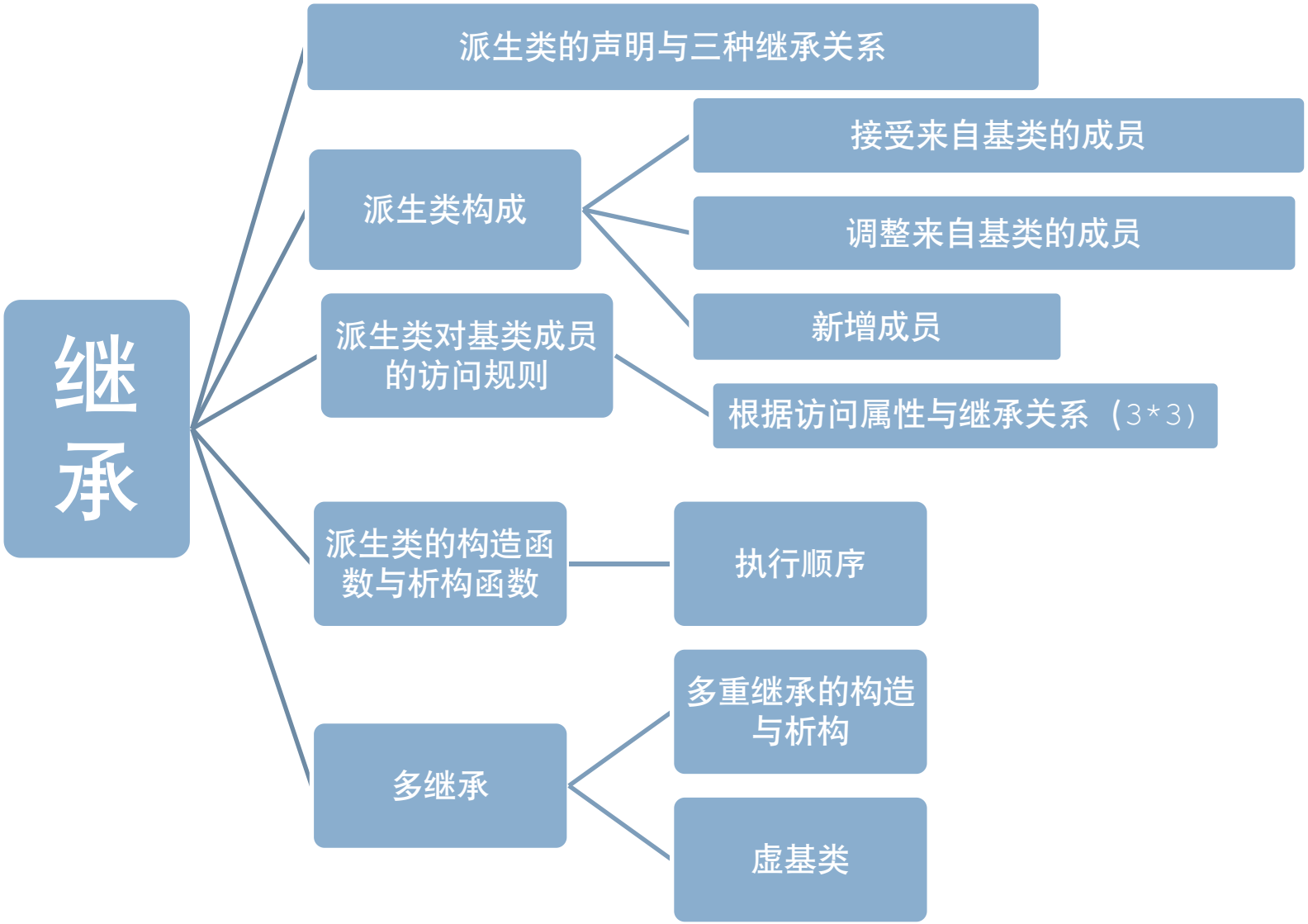
析构函数

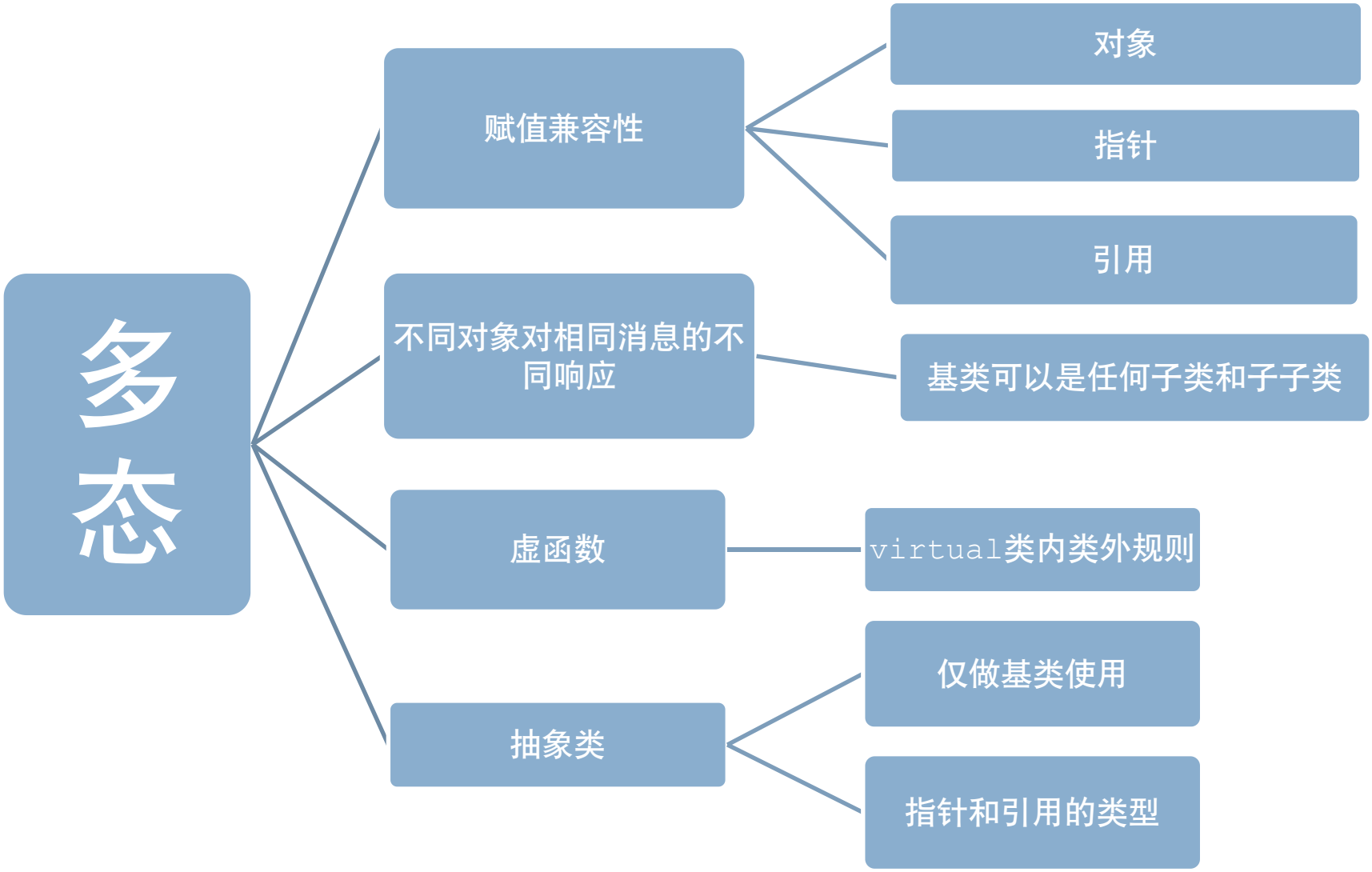
- 析构函数:
  ~类名（）
- 析构函数不返回任何值
  最好声明为virtual
- 没有参数且不可重载
- 只能由系统隐式调用


运算符重载
=的重载
浅赋值和深赋值
++和 --的重载
[]的重载
<<
>>
友元方式和成员方式在调用上区别
a@b: operator (a,b)
a@b: a.operator (b)
@a: operator (a)
@a: a.operator ( )
类型转换
内置类型间的转换
类类型与内置类型之间的转换
转换构造函数
类型转换函数


静态成员
为什么有这个静态机制
静态数据成员
Static 数据类型 数据成员名
静态数据成员的访问和调用语法?
声明在哪里？初始化又是在哪里?
静态成员函数
语法（类外定义不加static)
三种调用方式


友元
为什么要有这个友元机制
友元函数
不是当前类的成员函数，独立于当前类
类内和类外定义的语法格式
友元类
传递性
单向性


继承
派生类的声明与三种继承关系
派生类构成
接受来自基类的成员
调整来自基类的成员
新增成员
派生类对基类成员的访问规则
根据访问属性与继承关系（3*3）
派生类的构造函数与析构函数
执行顺序
多继承
多重继承的构造与析构
虚基类


多态
赋值兼容性
对象
指针
引用
不同对象对相同消息的不同响应
基类可以是任何子类和子子类
虚函数
virtual类内类外规则
抽象类
仅做基类使用
指针和引用的类型


Thank You !