



逻辑教育
Logic education

大师班第十三天

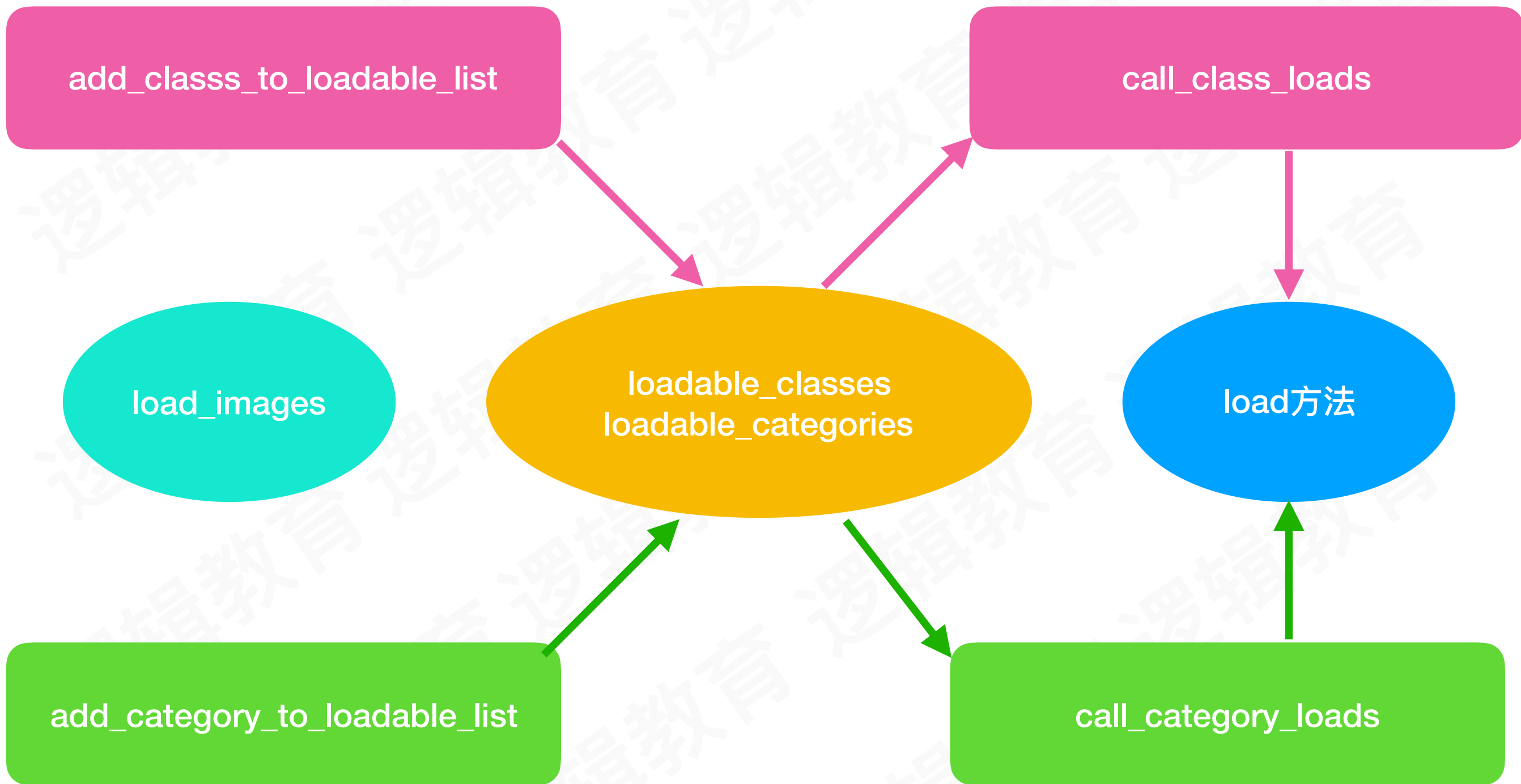
和谐学习，不急不躁

LG_Cooci

课程研发:Cooci老师
课程授课:Cooci老师

转载需注明出处,不得用于商业用途.已申请版权保护

load_images 分析



课程研发:Cooci老师
课程授课:Cooci老师



Runtime是什么

runtime 是由C 和C++ 汇编 实现的一套API，为OC语言加入了面向对象，运行时的功能

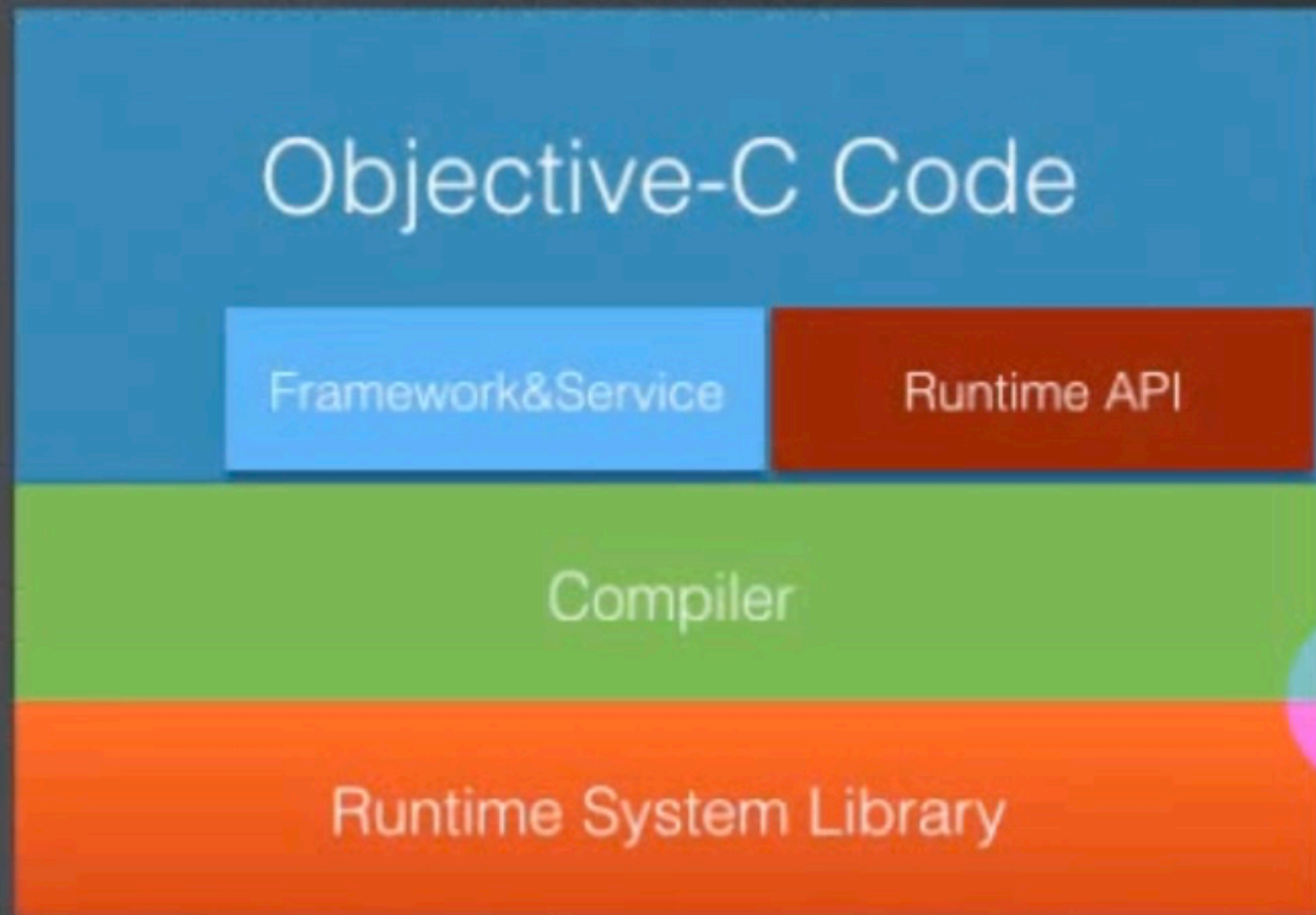
运行时(Runtime)是指将数据类型的确定由编译时推迟到了运行时 -

举例子🍎： extension - category 的区别

平时编写的OC代码，在程序运行过程中，其实最终会转换成Runtime的C语言代码，`Runtime` 是 `Object-C` 的幕后工作者

课程研发:Cooci老师

课程授课:Cooci老师





方法的本质，sel是什么？IMP是什么？两者之间的关系又是什么？

方法的本质：发送消息，消息会有以下几个流程

- 1: 快速查找 (objc_msgSend) ~ cache_t 缓存消息
- 2: 慢速查找~ 递归自己| 父类 ~ lookupImpOrForward
- 3: 查找不到消息: 动态方法解析 ~ resolveInstanceMethod
- 4: 消息快速转发~ forwardingTargetForSelector
- 5: 消息慢速转发~ methodSignatureForSelector & forwardInvocation

sel 是方法编号 ~ 在read_images 期间就编译进入了内存

imp 就是我们函数实现指针，找imp 就是找函数的过程

sel 就相当于书本的目录 tittle

imp 就是书本的页码

查找具体的函数就是想看这本书里面具体篇章的内容

- 1: 我们首先知道想看什么 ~ tittle (sel)
- 2: 根据目录对应的页码 (imp)
- 3: 翻到具体的内容

课程研发:Cooci老师

课程授课:Cooci老师



能否向编译后的得到的类中增加实例变量？能否在运行时创建的类中添加实例变量

答案：

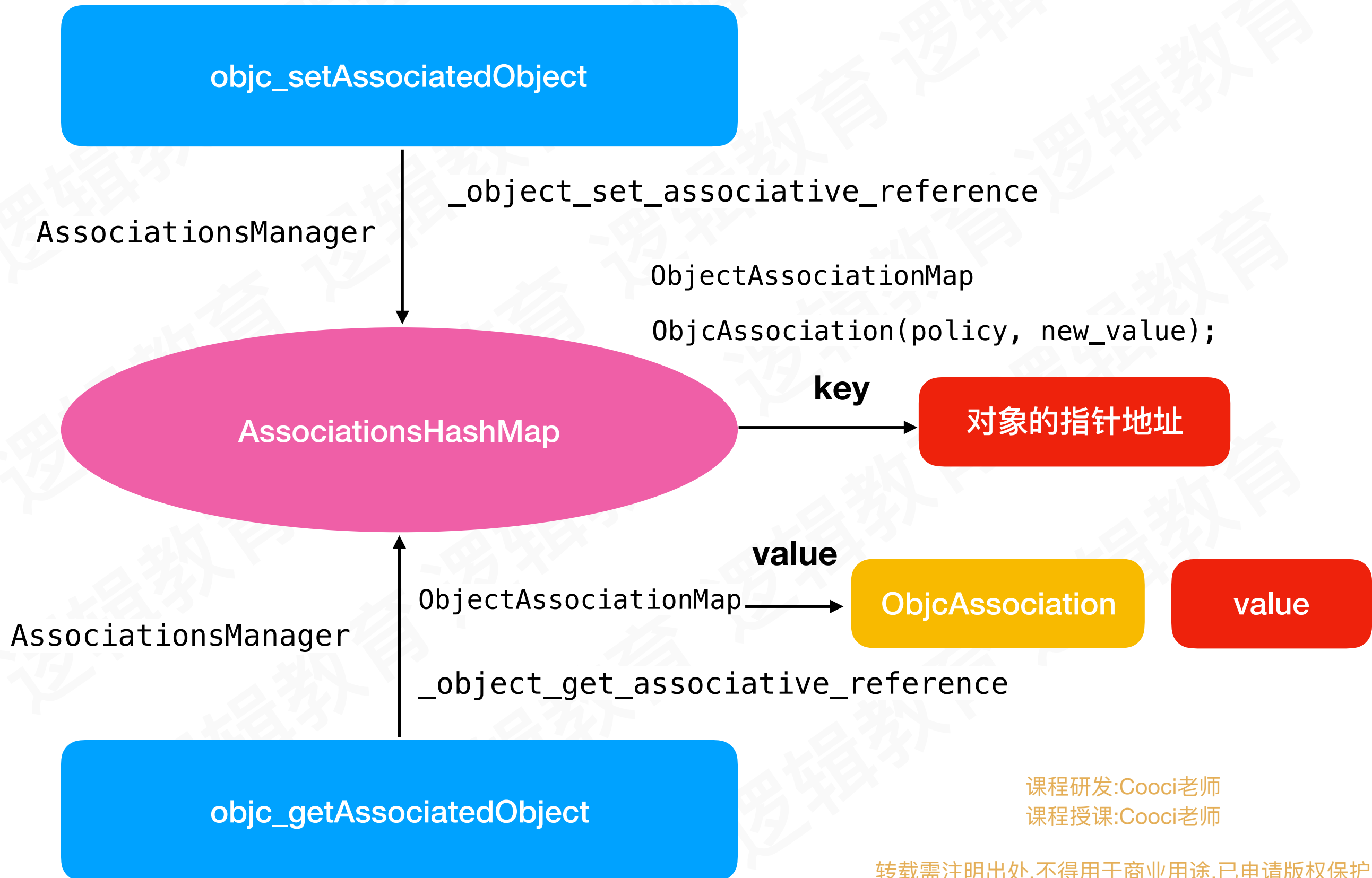
1：不能向编译后的得到的类中增加实例变量

2：只要类没有注册到内存还是可以添加

原因：我们编译好的实例变量存储的位置在 ro，一旦编译完成，内存结构就完全确定就无法修改

可以添加属性 + 方法

Associate方法关联的对象





isKindOfClass 和 isKindOfClass

```
BOOL re1 = [(id)[NSObject class] isKindOfClass:[NSObject class]]; // 1
BOOL re2 = [(id)[NSObject class] isKindOfClass:[NSObject class]]; // 0
BOOL re3 = [(id)[LGPerson class] isKindOfClass:[LGPerson class]]; // 0
BOOL re4 = [(id)[LGPerson class] isKindOfClass:[LGPerson class]]; // 0
NSLog(@" re1 :%hhd\n re2 :%hhd\n re3 :%hhd\n re4 :%hhd\n", re1, re2, re3, re4);
```

```
BOOL re5 = [(id)[NSObject alloc] isKindOfClass:[NSObject class]]; // 1
BOOL re6 = [(id)[NSObject alloc] isKindOfClass:[NSObject class]]; // 1
BOOL re7 = [(id)[LGPerson alloc] isKindOfClass:[LGPerson class]]; // 1
BOOL re8 = [(id)[LGPerson alloc] isKindOfClass:[LGPerson class]]; // 1
NSLog(@" re5 :%hhd\n re6 :%hhd\n re7 :%hhd\n re8 :%hhd\n", re5, re6, re7, re8);
```

isKindOfClass 和 isKindOfClass 主要看清楚查找的方向



[self class]和[super class]的区别以及原理分析

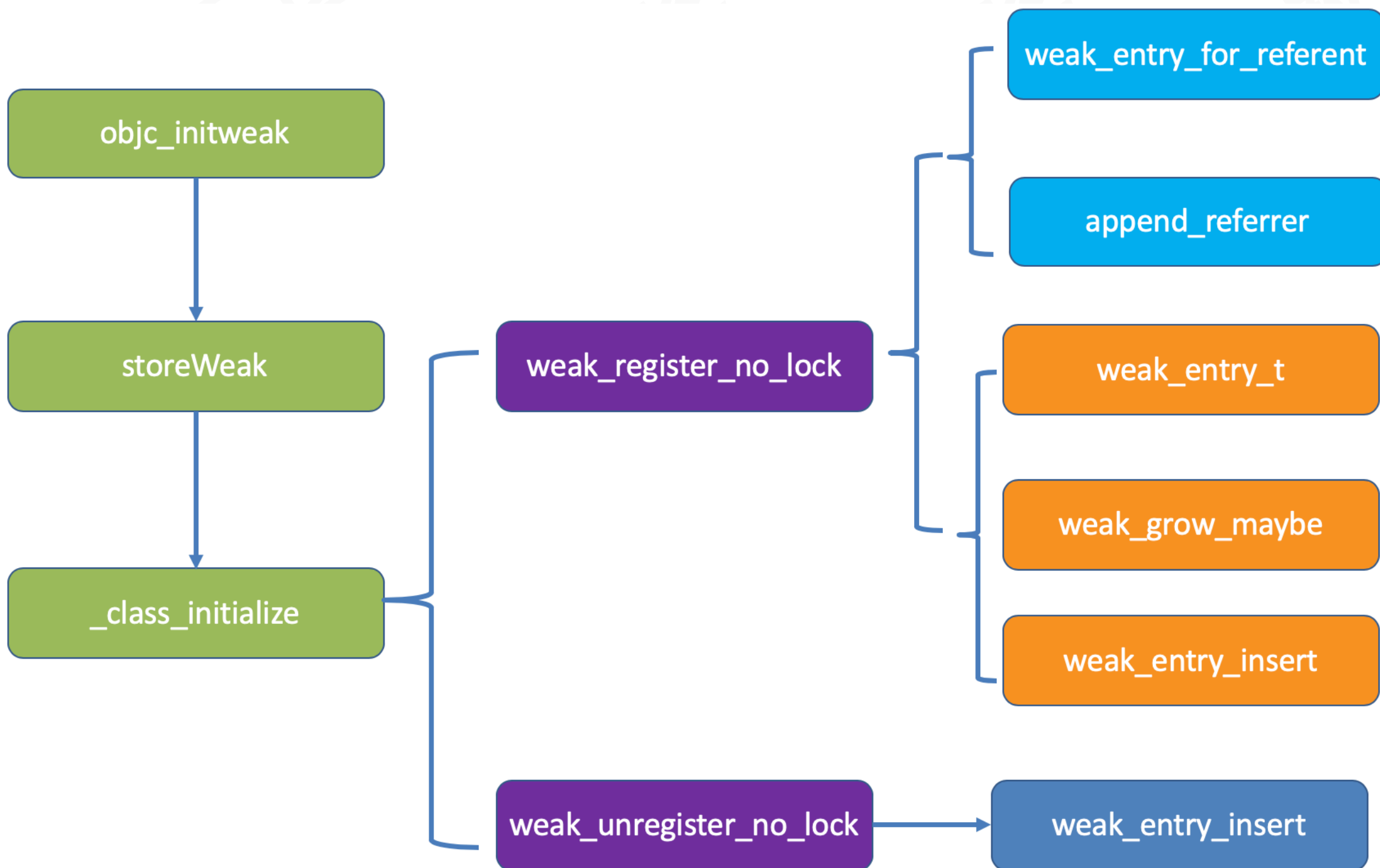
[self class] 就是发送消息objc_msgSend, 消息接受者是 self 方法编号: class

[super class] 本质就是objc_msgSendSuper, 消息的接受者还是 self 方法编号: class

只是objc_msgSendSuper 会更快 直接跳过 self 的查找



Runtime是如何实现weak的，为什么可以自动置nil



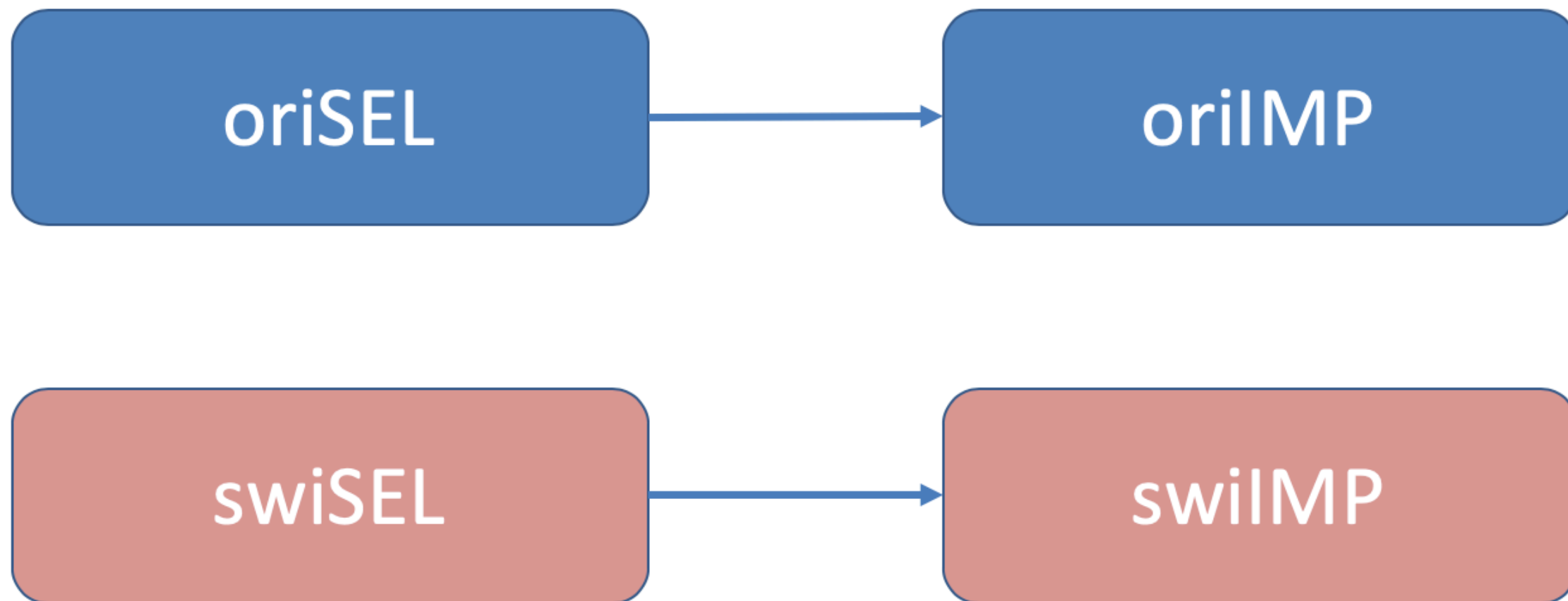
课程授课:Cooci老师



- 1.通过SideTable找到我们的weak_table
- 2.weak_table 根据referent 找到或者创建 weak_entry_t
- 3.然后append_referrer(entry, referrer)将我的新弱引用的对象加进去entry
- 4.最后weak_entry_insert 把entry加入到我们的weak_table

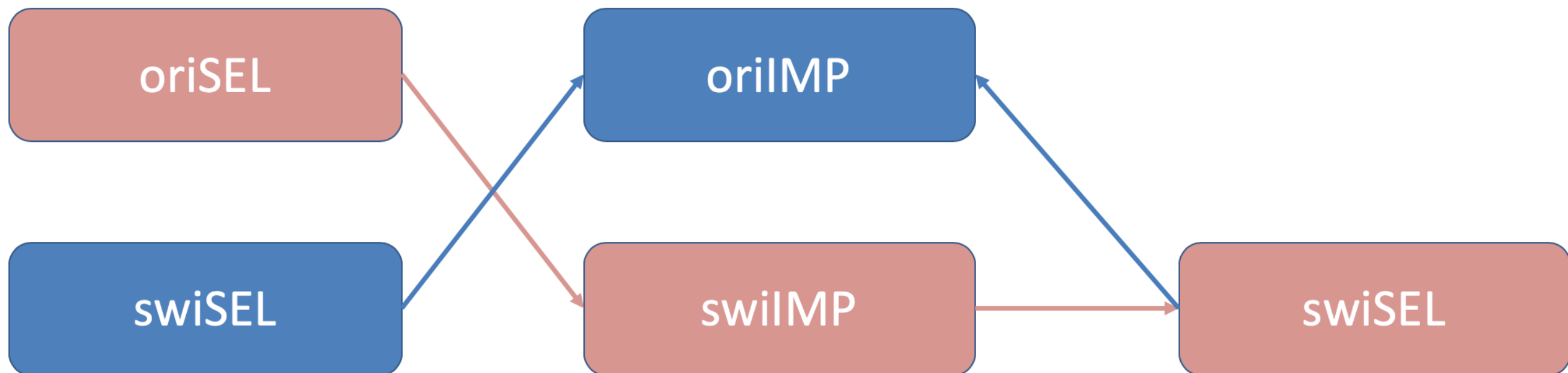


Method Swizzling的坑与应用



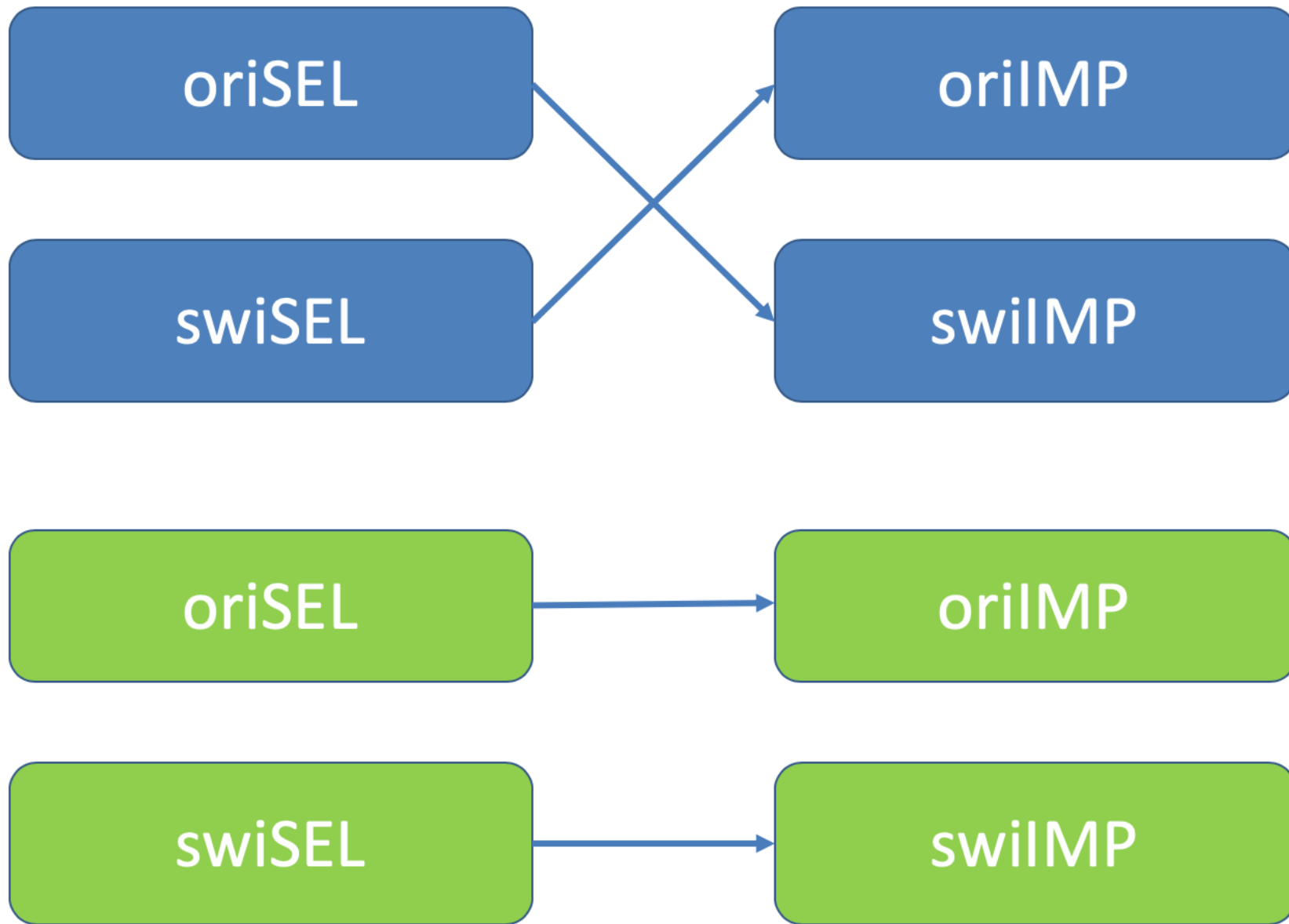


Method Swizzling的坑与应用





Method Swizzling的坑与应用





一：交换类主动调用load

解决办法：单例设计



二：交换没有自己没有实现，父类实现

解决办法：

A:会先尝试给自己添加要交换的方法

B:然后再将父类的IMP给swizzle



逻辑教育
Logic education

Hello Cooci

我就是我，颜色不一样的烟火

课程研发:Cooci老师
课程授课:Cooci老师

转载需注明出处,不得用于商业用途.已申请版权保护