

Objective-C

- 1、分类
- 2、关联对象
- 3、扩展
- 4、代理
- 5、通知
- 6、KVO
- 7、KVC
- 8、属性关键字

分类（category）

- 声明私有方法
- 分解体积庞大的类文件
- 把Framework的私有方法公开

特点

- 运行时进行决议 编译时没有添加到宿主类上，在运行时通过runtime添加到宿主类上
- 可以为系统类添加分类

分类中都可以添加哪些内容？

- 实例方法
- 类方法
- 协议
- 属性

```
1 struct category_t {
2     const char name;    //分类类名
3     classref_t cls;     //具体含义不太清楚 （宿主类）
4     WrappedPtr<method_list_t, PtrauthStrip> instanceMethods; //对象方法列表
5     WrappedPtr<method_list_t, PtrauthStrip> classMethods; //类方法列表
6     struct protocol_list_t protocols; //协议方法列表
7     struct property_list_t instanceProperties; //属性列表
```

```

8      // Fields below this point are not always present on disk.
9      struct property_list_t _classProperties;
10
11     method_list_t methodsForMeta(bool isMeta) {
12         if (isMeta) return classMethods;
13         else return instanceMethods;
14     }
15
16     property_list_t propertiesForMeta(bool isMeta, struct header_info hi);
17
18     protocol_list_t protocolsForMeta(bool isMeta) {
19         if (isMeta) return nullptr;
20         else return protocols;
21     }
22 };

```

关联对象 (Associate)

能否给分类添加“成员变量”？

```

1 id objc_getAssociatedObject(id object, const void *key)
2
3 void objc_setAssociatedObject(id object, const void *key, id value,
   objc_AssociationPolicy policy)
4
5 void objc_removeAssociatedObjects(id object)

```

关联对象没有直接添加到宿主对象中

关联对象由AssociationsManager管理在AssociationHashMap存储

所有对象的的关联内容都在同一个全局容器中。

扩展 (Extension)

一般用扩展做什么？

- 声明私有属性
- 声明私有方法
- 声明私有成员变量



分类和扩展的区别

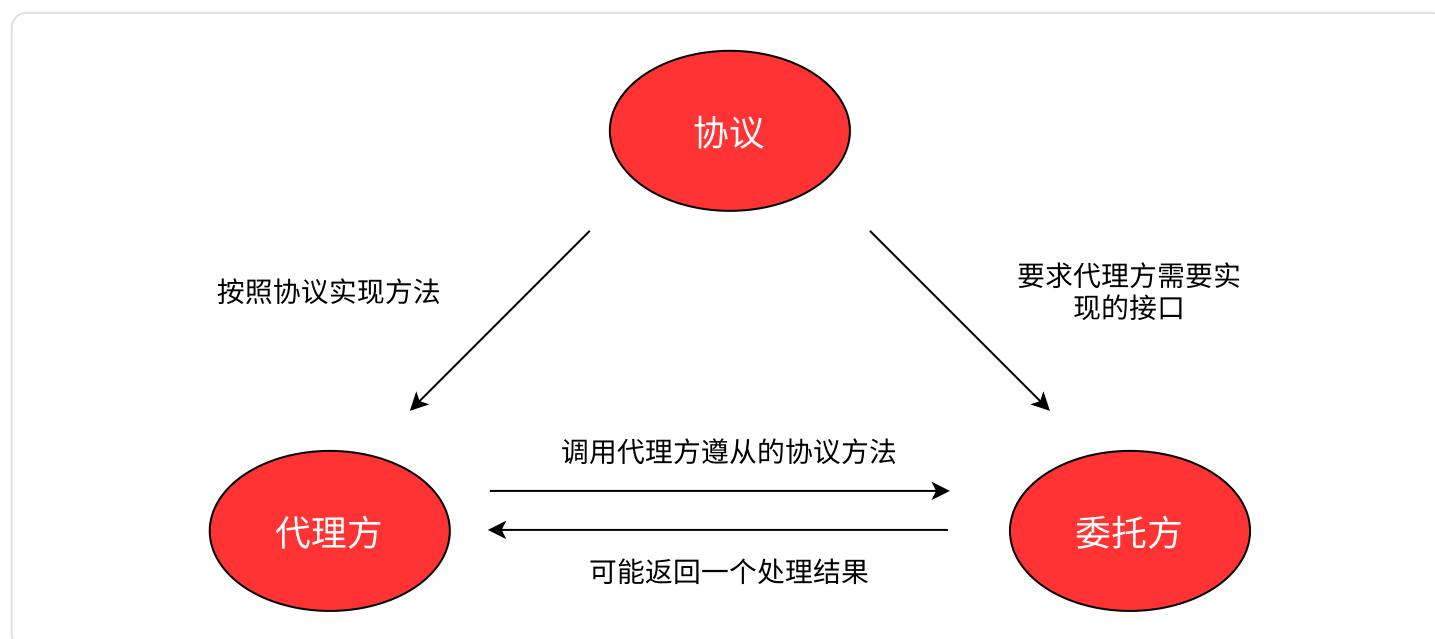
扩展

- 编译时决议
- 只以声明的形式存在，多数情况下寄生于宿主类的.m文件中。
- 不能为系统类添加扩展

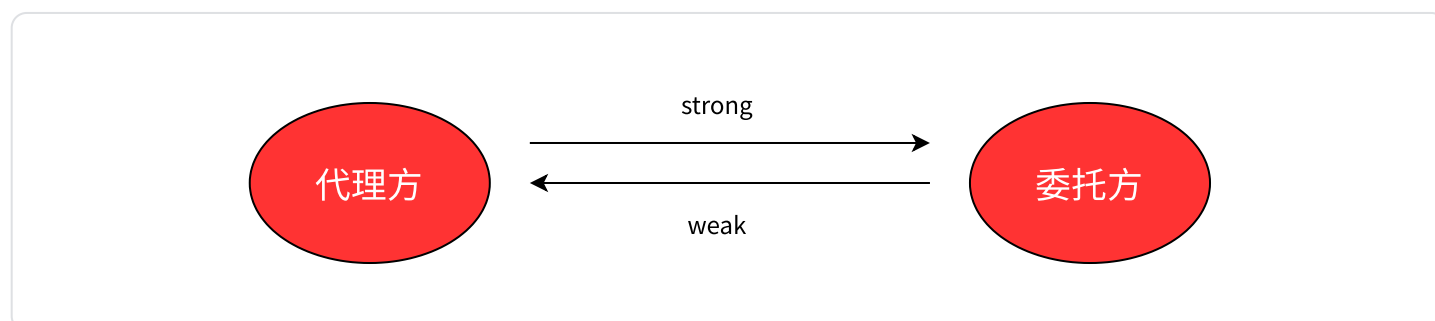
代理 (Delegate)

- 准确的说是一种软件设计模式
- iOS当中以@protocol形式体现
- 传递方式**一对一**

代理的工作流程是什么



一般声明为weak以规避循环引用



通知 (NSNotification)

使用**观察者模式**来实现的用于跨层传递消息的机制

传递方式**一对多**

kVO

什么是KVO

Key-value-observing

是观察者设计的一种实现

Apple使用了isa混写 (isa-swizzling) 来实现KVO

```
1 addObserver:<#(nonnull NSObject )#> forKeyPath:<#(nonnull NSString )#> options:
  <#(NSKeyValueObservingOptions)#> context:<#(nullable void *)#>
```

调用以上方法会在运行时创建被观察者对象的子类NSKVONotifying_Object，并重写setter方法，通知所有观察者

1.通过KVC设置value能否生效？能够生效

2.通过成员变量直接赋值value能否生效

KVC

```
1 (void)setValue:(id)value forKey:(NSString *)key{
2 }
3
4 (void)setValue:(id)value forUndefinedKey:(NSString *)key{
5 }
```

setValue:forKey方法调用时