

day02

ARC 与 MRC 【掌握】

- MRC项目 转 ARC项目
- MRC 与 ARC 共存
- Edit -> Refactor-> Convert to Objective-C ARC
- 选中项目跟节点 -> 选中target -> Build Phases->Compile Sources (1 item) ->在需要设置为不使用ARC文件所在的行与compiler Flags的列 的交汇处双击，写上命令 -fno-objc-arc

分类【掌握】

- 【掌握】作用：扩充原有类的功能
- 【掌握】定义
 - 名称：原有类的名称(分类名称)
 - 文件名称：原有类的名称 + 分配名称
- 【了解】优点
 - 代码模块化
 - 团队协作
 - 可以扩充系统自带的功能
- 【掌握】使用
 - 在分类中可以声明和实现新的方法
 - 在分类的方法中可以访问原有类的成员变量
- 【掌握】练习
 - 给NSString类增加一个计算字符串中有多少数字的功能
 - 给NSObject方法增加一个run方法
 - 给NSObject扩展分类，应该非常谨慎，因为他会作用到所的类上面
- 【掌握】注意点
 - 分类只能扩充功能不能扩充属性
 - 在分类使用@property 只能生成方法的声明，不能生成方法的实现，更不能生成成员变量
 - 分类中不能有与原来类同名的方法，不同分类中也不能有同名的方法
 - 【理解】如果分类中如果存在与原来类中的同名函数，那么分类中的方法会覆盖原有类中的方法(哪怕你没有导入这个分类)
 - 【理解】如果多个分类中有相同的方法，最后将使用最后编译的那个分类中的方法

【掌握】类扩展

- 用在.m文件中，给类增加私有的成员变量和方法
- 类扩展中小括号中没有内容

SEL【掌握】

- 概念【掌握】它是一种代表着方法的签名数据类型，我们也成为它使一个选择器
- 【掌握】定义普通的变量
- 使用
 - 【理解】作为方法实参
 - 【了解】方法形参
 - 【了解】作为方法的返回值
 - 【了解】OC中的方法调用过程
- 与NSObject 相关方法配合使用
 - 【掌握】判断一个对象是否实现了某个方法
 - 【了解】执行对象上某个的方法
 - 【了解】延迟执行某个对象上的方法

【掌握】block

- 【掌握】作用：封装代码
- 【掌握】优点：消除了函数不能定义在方法内部的缺点，使用起来更加灵活
- 【理解】使用
 - 【掌握】定义变量
 - 【掌握】定义功能
 - 【掌握】作为函数参数
 - 【理解】typedef 来定义类型
- 【理解】使用场景
- 【掌握】注意点

【掌握】协议

- 【掌握】概念：两个或两个以上对象之间约定
- 【掌握】定义
 - @protocol 协议名称 <NSObject>
 - //声明方法
 - @end
 - 中间只写方法声明
 - 需要遵守协议NSObject
 - 以@protocol 开头以@end 结尾
- 【掌握】协议的基本使用
 - 一个协议也可以遵守多个协议
 - 第一步：定义协议
 - 第二步：遵守协议
 - 在声明类的时候，在类名后面的通过<协议名称>来遵守协议
 - 在类扩展中遵守协议
 - 第三步：实现协议中的方法
 - 【掌握】注意点
- 【掌握】id 类型与协议
- 【掌握】两个关键字
 - @required
 - @optional

【掌握】代理设计模式

- 使用场景
 - 当A类对象发生了什么事，想通知B类对象的时候(让B类对象成为A类对象的代理)
 - 当A类对象有些事情，自己无法处理，需要B类对象帮助它处理(让B类对象成为A类对象的代理)
 - 当B类对象想要监听A类对象中发生的事情的时候(让B类对象成为A类对象的代理)
- 实现步骤
 - A类中定义协议
 - A类中添加一个类型 id<ADelegate> delegate的属性
 - A类中当发生了什么事的时候,调用delegate相应的方法
 - B类遵守ADelegate的协议
 - B类中实现ADelegate协议中声明的方法
 - 将B类对象赋值给A类对象的delegate属性
- 使用场景
 - 婴儿想吃吃了要通知保姆，让保姆成为婴儿代理
 - 保姆想知道婴儿什么时候想吃吃了，让保姆成为婴儿的代理
 - 婴儿自己不会吃，需要保姆帮忙，让保姆成为婴儿的代理