



Objective-C编程

第七讲：类的扩展



本讲内容

- 一、掌握NSDate的使用
- 二、掌握NSDateFormatter的使用
- 三、掌握Category的定义和使用
- 四、掌握Extension的定义和使用
- 五、掌握Protocol的定义和使用
- 六、掌握delegate的使用



一、NSDate和NSDateFormatter



NSDate的使用

- 在iOS开发中，NSDate的使用场景很多，比如：聊天程序里：几分钟前，昨天，几天前的计算；网络请求中的时间戳（计算当前时间距离1970年1月1日的秒数）



NSDate的使用

- NSDate是Cocoa中用于处理日期和时间的基础类，封装了某一给定的时刻（含日期，时间，时区）。
- 使用+date方法获取当前时间
- 例如：NSDate *nowDate = [NSDate date];
- 注意NSLog(@"%@",nowDate); 无论你是哪个时区的时间，打印时总是打印对应的0时区时间。



NSTimeInterval

- NSTimeInterval (即double类型) 用以表示以秒为单位的时间间隔
- 可以使用-initWithTimeIntervalSinceNow:方法传入一个NSTimeInterval参数, 创建一个NSDate对象
- 例如: NSDate * tomorrowDate = [[NSDate alloc] initWithTimeIntervalSinceNow:24*60*60];
- NSDate * yesterdayDate = [[NSDate alloc] initWithTimeIntervalSinceNow:-1 * 24*60*60];
- 取两个时间对象的间隔: NSTimeInterval = [tomorrowDate
timeIntervalSinceDate:yesterdayDate];



练习1

- 计算当前时间和一个固定时间的差值，如果差值在60秒内，输出“刚刚”，如果在60秒外3600秒内，输出“xx分钟前”，如果3600秒外，3600*24秒内，输出“xx小时前”。



NSDateFormatter的使用

- **NSDateFormatter**是iOS中的日期格式类，功能是实现NSString和NSDate的互转。
- 常见的时间格式化字符串有以下一些：**y** 年、**M** 年中的月份、**d** 月份中的天数、**H** 一天中的小时数(0-23)、**h** am/pm 中的小时数(1-12)、**m** 小时中的分钟数、**s** 分钟中的秒数 等
- 指定日期格式：`NSDateFormatter * formatter = [[NSDateFormatter alloc] init];`
- `[formatter setDateFormat:@"yyyy-MM-dd HH:mm:ss"];`



NSDate转NSString

- 日期转化为字符串: NSDateFormatter*formatter = [[NSDateFormatter alloc] init];
- [formatter setDateFormat:@"yyyy-MM-dd hh:mm:ss"];
- NSString *dateString=[formatter stringFromDate: [NSDate date]];



NSString转NSDate

- 时间字符串转化为相对应的日期: NSDateFormatter * formatter = [[NSDateFormatter alloc] init];
- [formatter setDateFormat:@"yyyy-MM-dd HH:mm:ss"];
- NSString * dateStr = @"2008-08-08 20:08:08";
- NSDate * date = [formatter dateFromString:dateStr];



练习2

- 定义一个NSDateFormatter，设置合适的格式。
- 将字符串@“2014年05月01日 10点23分18秒”转换为NSDate对象。



二、类的扩展



Category

- Category也叫分类或类目
- 主要作用是为 **没有源代码** 的类添加方法。
- 通过Category添加的方法会成为原类的一部分。从而达到扩展一个类的功能。



如何定义 Category

- 新建文件
- 选择Objective-C Category模板
- 填写类名和分类名
- .h文件添加方法声明
- .m添加方法实现



Category的声明

- NSString+SayHi.h文件
- @interface NSString (SayHi)
- - (void)hi;
- @end



Category的实现

- NSString+SayHi.m文件
- #import "NSString+SayHi.h"
- @implementation NSString(SayHi)
- - (void)hi{
 NSLog(@"这是通过category为NSString添加的hi方法");
}
- @end



Category与子类的区别

	Category	Subclass
功能	只能为类添加方法	既能为类添加方法也能添加变量
特点	新添加的方法会成为原始类的一部分，被子类继承	新添加的方法只有子类才有，父类不具有
使用	使用原始类的实例（如果是-方法）或者原始类（如果是+方法）调用方法	子类才能调用。如果在项目开发到尾声的时候，使用子类添加了方法，需要对已写代码做类型替换（将父类替换为子类）



练习3

- 使用Category为NSDate类添加一个类方法（dateWithDateString:），实现将下述字符串转换为NSDate对象。
- 将@“20140402142850”转换为NSDate。



Extension

- Extension的主要作用是管理类的“私有”方法。
- 面向对象编程也叫面向接口编程。
- 在设计一个类的时候，有些方法需要对外公开（我们称为接口），有些方法仅仅内部使用（比如：是某个方法的一个小步骤）。
- Extension的功能是帮我们去管理这些内部使用的方法（“私有”方法）。



定义Extension

- Extension的语法格式和Category很相似
- 相当于把Category的.h文件挪到了原始类的.m文件中
- Extension针对的是自己的类，必须有源代码的类



定义Extension

- xxx.m文件
- @interface xxx (ExtensionName)
- //你的方法列表
- @end
- @implementation xxx
- //方法的实现
- @end



Category和Extension的区别

	Category	Extension
作用	为没有源代码的类添加方法	管理类的私有方法
格式	定义一对.h 和.m	把代码写到原始类的.m 文件中



Protocol

- Protocol（协议），是iOS开发中常用的技术。
- 协议是一套标准（一堆方法的声明），只有.h文件。就像一张任务清单（或便利贴），上面写了一堆需要处理的事。清单交给谁，谁就要去完成清单上规定的任务。
- 接受协议的对象实现协议中定义的方法。即：清单交给谁，谁就要去完成清单上规定的任务。



如何定义Protocol

- @protocol开始，以@end结束，例如:
- @protocol MarryProtocol <NSObject>
- - (void)makeMoney;
- @optional
- - (void)washcloth;
- - (void)cook;
- @end



如何定义Protocol

- @protocol开始，以@end结束，例如:

- @protocol MarryProtocol <NSObject>

- - (void)makeMoney;

- @optional

- - (void)washcloth;

- - (void)cook;

- @end

协议中的方法默认是必须实现的，即@required。
关键字 @optional 修饰的方法是可选的，可实现也可不实现。



遵守协议

- 遵守协议分两步
- 1、在类的.h文件 父类名后写上<协议名>。
- 2、在.m文件中实现协议中的方法。
- 相当于给这个类添加了若干个方法。这个类的实例就可以调用这些方法。



如何遵守协议

- #import "MarryProtocol"
- @interface Boy : NSObject <MarryProtocol>
- ...
- @end
- @implementation Boy
- //协议中方法的实现
- @end



delegate设计模式

- Protocol的核心使用场景是实现delegate设计模式。
- delegate（代理）。通俗的讲就是代理商，主要的任务是帮你完成一些任务。
- 比如：保姆就可以认为是delegate，主要的任务是帮你带孩子、做饭、洗衣服等。



delegate设计模式



delegate设计模式

● 使用场景： 凡是某些任务自己不去实现，想让别人去实现的时候，就可以指定一个代理，让代理帮你去做。你只需要通知代理去做某事。



delegate设计模式



delegate的应用

● 课堂练习：结婚。



练习4

- 建立一个保姆协议，包含带孩子，做饭等方法。
- 建立一个母亲类，母亲有一个id<BaoMuDelegate>delegate
- 分别让医生、学生、老师作为代理，实现协议里的方法



总结

- NSDate是iOS中表示日期的类。可以使用NSDateFormatter控制NSString和NSDate的互转
- Category能为没有源代码的类添加方法
- Extension为类管理私有方法
- Protocol和delegate通常联合使用，是iOS开发中特别常用。



课程预告

● 属性