



无限互联

Infinite Interconnection

INFINITE
INTERCONNECTION

无限互联是国内唯一一家**专注**于iPhone和iPad软件开发培训机构，到目前为止为各大公司输送了一大批优秀的iOS高级软件研发人才。随着iOS6系统的发布，我们也在当天陆续发布**国内首套完整**的iOS开发的视频教程，手把手教您写代码，从入门到熟练再到精通。

高薪就业是检验一家培训机构质量的唯一标准，我们的学员高薪就业是对我们最好的肯定，也是我们前进的最强烈的动力，我们感谢同学们的努力，感谢你们对我们的支持！我们也将免费为你们提供最好的就业后的技术支持！

亲爱的同学们，你们的**高薪就业**才是我们最大的成功！

<http://www.iphonetrain.com>

版权所有：无限互联3G学院



无限互联

Infinite Interconnection

interconnection

第八课 KVC、KVO和谓词

主讲：陈 为

<http://www.iphonetrain.com>

版权所有：无限互联3G学院



- KVC的基本概念和用法
- KVO的基本概念和用法
- 键值观察设计模式的基本概念和用法
- 谓词的基本概念和用法



1、键值编码的基本概念和用法

- 键值编码的基本概念

KVC是Key Value Coding的简称，它是一种可以直接通过字符串的名字(key)来访问类属性的机制。使用该机制不需要调用存取方法和变量实例就可访问对象属性。本质上讲，键-值编码定义了你的程序存取方法需要实现的样式及方法签名。

在应用程序中实现键-值编码兼容性是一项重要的设计原则。存取方法可以加强合适的数据封装，而键-值编码方法在多数情况下可简化程序代码。

键-值 编码方法在Objective-C非标准协议（类目）NSKeyValueCoding中被声明，默认的实现方法由NSObject提供。

键-值编码支持带有对象值的属性，同时也支持纯数值类型和结构。非对象参数和返回类型会被识别并自动封装/解封。

KVC的基本用法

- 设置和访问

键/值编码中的基本调用包括-valueForKey: 和-setValue: forKey: 这两个方法，它们以字符串的形式向对象发送消息，字符串是我们关注属性的关键。

```
Person *jack = [[Person alloc] init];
NSMutableString *name = [[NSMutableString alloc] initWithFormat:@"jack"];
[jack setValue:name forKey:@"name"];
NSLog(@"jack name : %@", [jack valueForKey:@"name"]);
```

是否存在setter、getter方法，如果不存在，它将在内部查找名为_key或key的实例变量。通过KVC，可以获取不存在getter方法的对象值，无需通过对象指针直接访问。

这里我们需要注意，当我们通过setValue: forKey: 设置对象的值，或通过valueForKey来获取对象的值时，如若对象的实例变量为基本数据类型时(char、int、float、BOOL)，我们需要对数据进行封装。

路径与一对多的关系

- 路径

除了通过键设置值外，键/值编码还支持指定路径，像文件系统一样。

用“点”号隔开。

```
[book setValue:@"比尔" forKeyPath:@"author._name"];
```

```
[book valueForKeyPath:@"relativeBooks.price"]
```

- 一对多的关系

如果向NSArray请求一个键值，它实际上会查询数组中的每个对象来查找这个键值，然后将查询结果打包到另一个数组中并返回给你。

```
NSArray *booksArray = [NSArray arrayWithObjects:book1, book2, nil];  
[book1 release];  
[book2 release];  
[book setValue:booksArray forKey:@"relativeBooks"];  
NSLog(@"books 2: %@", [book valueForKeyPath:@"relativeBooks.price"]);
```

实现简单的运算

- KVC的简单运算

此外，还可以应用一些字符做简单运算。sum、min、max、avg、count

```
NSString *count = [book valueForKeyPath:@"relativeBooks.@count"];
NSLog(@"count : %@", count);
NSString *sum = [book valueForKeyPath:@"relativeBooks.@sum._price"];
NSLog(@"sum : %@", sum);
NSString *avg = [book valueForKeyPath:@"relativeBooks.@avg._price"];
NSLog(@"avg : %@", avg);
NSString *min = [book valueForKeyPath:@"relativeBooks.@min._price"];
NSLog(@"min : %@", min);
NSString *max = [book valueForKeyPath:@"relativeBooks.@max._price"];
NSLog(@"max : %@", max);
```

- KVC的缺点，你感觉到了吗？



无限互联

Infinite
Interconnection

INFINITE CONNECTION

课堂练习

<http://www.iphonetrain.com>

1. 定义一个Person对象，他叫乔布斯，他有许多苹果设备：1个macbook，价格8000；1个iPhone，价格5000；1个iPad，价格3800。使用kvc为Person设置名字，使用kvc计算乔布斯所有设备的总价。

版权所有：无限互联3G学院

2、KVO的基本概念

- 基本概念

Key Value Observing, 直译为：基于键值的观察者。它提供一种机制，当指定的对象的属性被修改后，则对象就会接受到通知。简单的说就是每次指定的被观察的对象的属性被修改后，KVO就会自动通知相应的观察者了。

与 NSNotification不同，键-值观察中并没有所谓的中心对象来为所有观察者提供变化通知。取而代之地，当有变化发生时，通知被直接发送至处于观察状态的对象。NSObject提供这种基础的键-值观察实现方法。

你可以观察任意对象属性，包括简单属性，对一或是对多关系。对多关系的观察者将会被告知发生变化的类型-也就是任意发生变化的对象。

键-值观察为所有对象提供自动观察兼容性。你可以通过禁用自动观察通知并实现手动通知来筛选通知。

注册观察者

- 注册观察者

为了正确接收属性的变更通知，观察对象必须首先发送一个 addObserver:forKeyPath:options:context: 消息至被观察对象，用以传送观察对象和需要观察的属性的关键路径，以便与其注册。选项参数指定了发送变更通知时提供给观察者的信息。使用 NSKeyValueObservingOptionOld 选项可以将初始对象值 以变更字典中的一个项的形式 提供给观察者。指定 NSKeyValueObservingOptionNew 选项可以将 新的值以一个项的形式 添加至变更字典。你可以使用逐位 “|” 这两个常量来指定接收上述两种类型的值。

- 示例:

```
[child addObserver:self forKeyPath:@"tired"  
options:NSKeyValueObservingOptionOld|NSKeyValueObservingOptionNew  
context:nil];
```


接收变更通知

- 接受变更通知

当对象的一个被观察属性发生变动时，观察者收到一个 `observeValueForKeyPath:ofObject:change:context:` 消息。所有观察者都必须实现这一方法。触发观察通知的对象和键路径、包含变更细节的字典，以及观察者注册时提交的上下文指针均被提交给观察者，`context` 可以为任意类型参数。

- 示例：

```
- (void)observeValueForKeyPath:(NSString *)keyPath
                        ofObject:(id)object
                        change:(NSDictionary *)change
                        context:(void *)c{
    if ([keyPath isEqualToString:@"key"]) {
        NSLog(@"");
    }
}
```

移除观察者身份

- 移除观察者身份

你可以发送一条指定观察方对象和键路径的removeObserver:forKeyPath:消息至被观察的对象，来移除一个键-值观察者。（当我们达到目的时）

- 示例：

```
[child removeObserver:self forKeyPath:@"key"];
```




课堂实例1：

- 将保姆与孩子的例子修改成键值观察模式
 - 对象：孩子、保姆
 - 保姆对孩子的状态进行观察
 - 注意代理设计模式与键值观察模式的区别（前者一对一、后者一对多）

谓词的基本概念

- 概念

cocoa中提供了NSPredicate类，指定过滤器的条件。将符合条件的对象保留下来。

- 创建谓词

```
// 设置谓词条件
NSPredicate *predicate = [NSPredicate predicateWithFormat:@"age <= 28"];
for (Person *person in array) {
    // 表示指定的对象是否满足谓词条件
    if ([predicate evaluateWithObject:person]) {
        //NSLog(@"person name : %@", person.name);
    }
}
```

```
// 返回一个符合谓词条件的数组
NSArray *newArray = [array filteredArrayUsingPredicate:predicate];

for (Person *person in newArray) {
    //NSLog(@"person name : %@", [person valueForKey:@"_name"]);
}
```




- 格式占位符

```
// 格式占位符号
NSPredicate *pre = [NSPredicate predicateWithFormat:@" age <= %d", 30];
NSArray *array2 = [array filteredArrayUsingPredicate:pre];

for (Person *person in array2) {
    //NSLog(@"person name 2 %@", [person valueForKey:@"_name"]);
}
}
```



- 逻辑运算符

```
// 运算符 的加入 谓词不区分大小 && AND || OR
NSPredicate *pre3 = [NSPredicate predicateWithFormat:@"name > 'bruse' &&
age < %d", 30];
NSArray *array4 = [array filteredArrayUsingPredicate:pre3];
```

- IN

```
// 关键字 注意字符串一定要添加''
NSPredicate *pre4 = [NSPredicate predicateWithFormat:@"self.name IN
{'rose', 'bruse'}"];
NSArray *array5 = [array filteredArrayUsingPredicate:pre4];
NSLog(@"person name : %@", [array5 valueForKey:@"_name"]);
```


关键字

- 以**开始——BEGINSWITH

```
// BEGINSWITH 检查某个字是否以**开头
NSPredicate *pre5 = [NSPredicate predicateWithFormat:@"self.name
BEGINSWITH 'J'"];
NSArray *array6 = [array filteredArrayUsingPredicate:pre5];
NSLog(@"person name : %@", [array6 valueForKey:@"name"]);
```

- 以**结束——ENDSWITH

```
// ENDSWITH 检查某个字符是以**结尾
NSPredicate *pre6 = [NSPredicate predicateWithFormat:@"self.name endswith
'e'"];
NSArray *array7 = [array filteredArrayUsingPredicate:pre6];
NSLog(@"array7 : %@", [array7 valueForKey:@"name"]);
```



- 包含——CONTAINS

```
// CONTAINS 检查包含某个字符
NSPredicate *pre8 = [NSPredicate predicateWithFormat:@"self.name CONTAINS '小'"];
NSArray *array8 = [array filteredArrayUsingPredicate:pre8];
NSLog(@"array8 : %@", [array8 valueForKey:@"name"]);
```




无限互联

Infinite
Interconnection

INFINITE CONNECTION

今日习题

<http://www.iphonetrain.com>

1. 熟练掌握谓词的基本用法
2. 中介找房修改成键值观察模式

版权所有：无限互联3G学院



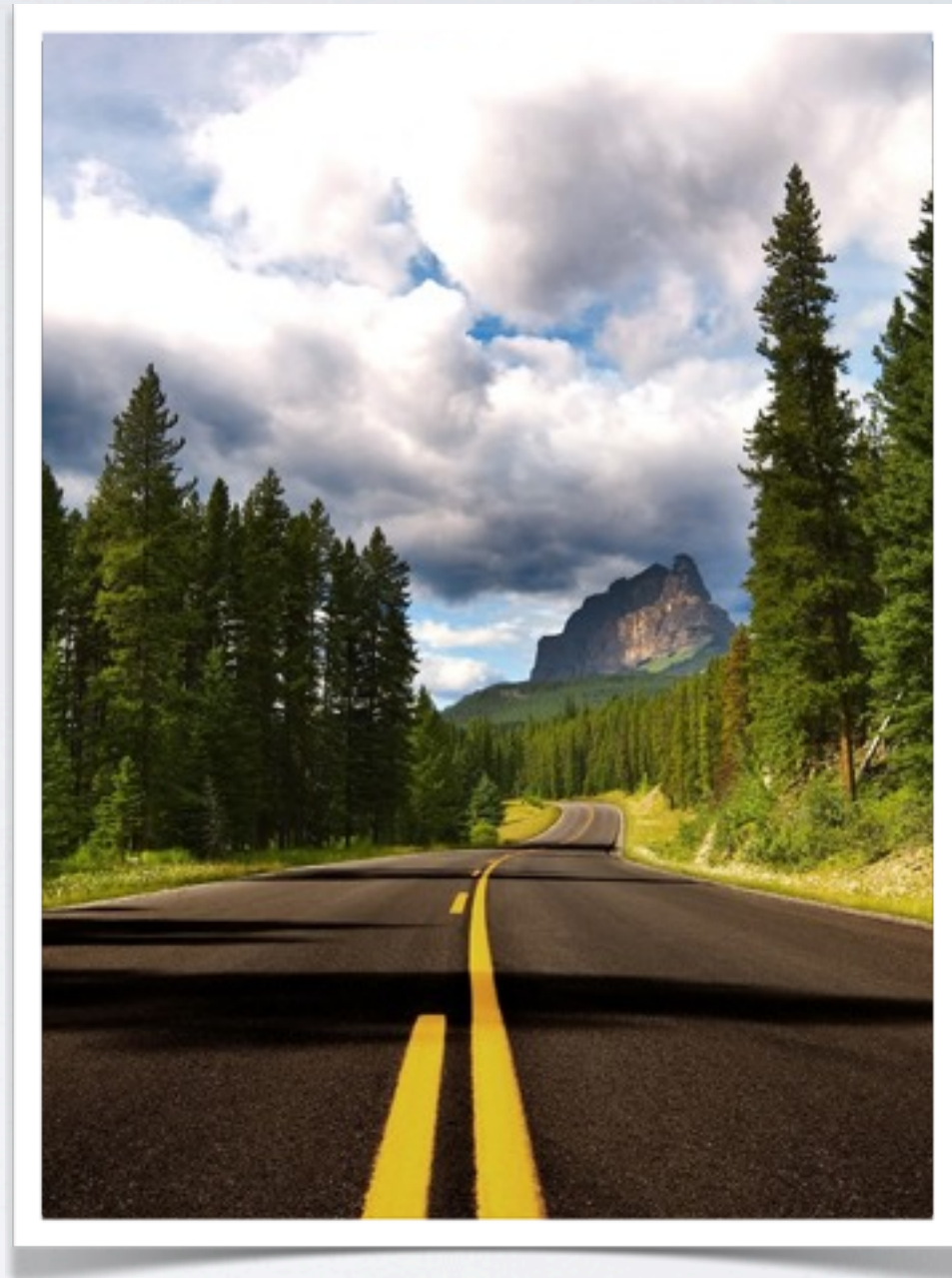
无限互联
Infinite
Interconnection

INFINITE CONNECTION

下节预告

<http://www.iphonetrain.com>

- 下一节课将会进行“OC、C”的笔试、面试和上机考试，请大家提前做好准备



版权所有：无限互联3G学院

未经允许不得将视频用于商业用途，否则将追究其法律责任！

无限互联网站：<http://www.iphonetrain.com>

博客：<http://blog.csdn.net/xbiii3s/>

公司E-mail：wxhl2805@gmail.com

老师E-mail：xbiii3s@gmail.com

视频讲解过程中如有不妥之处，欢迎大家将信息反馈到我的Email中，我们会努力完善！谢谢各位的支持。

视频持续更新中... 敬请期待！