1、指针分类

（1）强指针：默认的情况下，所有的指针都是强指针，关键字strong

（2）弱指针：__weak关键字修饰的指针

声明一个弱指针如下：

__weak Person *p;

- 问题：

  1,将指针声明为强指针和弱指针的修饰符是什么？

```
//定义另外一个强指针指向
//__strong 修饰符,修饰这个指针是一个强指针,也可以不写
//__weak       修饰的这个指针是若指针
```

weak前面有两个__,中间没有空格

1，ARC的判断对象释放标准是什么？

2，ARC是iOS几出来的特性。

3，ARC与其他语言的”垃圾回收“是否一样?

1.是否有强指针指向

2.ios5

Xcode编译器的自己特性,其实底层实现还是MRC

2、什么是ARC?

Automatic Reference Counting，自动引用计数，即ARC，可以说是WWDC2011和iOS5所引入的最大的变革和最激动人心的变化。ARC是新的LLVM 3.0编译器的一项特性，使用ARC，可以说一举解决了广大iOS开发者所憎恨的手动内存管理的麻烦。

在工程中使用ARC非常简单：只需要像往常那样编写代码，只不过永远不写retain,release和autorelease三个关键字就好~这是ARC的基本原则。

当ARC开启时，编译器将自动在代码合适的地方插入retain, release和autorelease，而作为开发者，完全不需要担心编译器会做错（除非开发者自己错用ARC了）。

手动管理内存，可以简称MRC (Manual Reference Counting)

ARC与其他语言的”垃圾回收”机制不同。ARC：编译器特性；“垃圾回收”运行时特性

3.不一样

3、ARC工作原理及判断准则

ARC是Objective-C编译器的特性，而不是运行时特性或者垃圾回收机制，ARC所做的只不过是在代码编译时为你自动在合适的位置插入release或autorelease，

ARC的判断准则：

只要没有强指针指向对象，对象就会被释放。

注意：当使用ARC的时候，暂时忘记“引用计数器”，因为判断标准变了。

4、ARC机制图解

NSString *firstName = @"oneV";

这个时候firstName持有了@"OneV"。

强指针指向直接释放

弱指针会先释放空间,再把弱空间指向nil变为空指针

6-【理解】ARC快速入门

本小节知识点：

1、【了解】ARC机制判断

2、【了解】ARC快速使用

1、ARC机制判断

iOS5以后，创建项目默认的都是ARC

ARC机制下有几个明显的标志：

1）不允许调用对象的release方法

2）不允许调用autorelease方法

3）再重写父类的dealloc方法时，不能再调用 [super dealloc];

2）使用

正常创建对象，不用手动释放对象

```
port <Foundation/Foundation.h>
port "Dog.h"
main(int argc, const char * ar
@autoreleasepool {

    Dog *jd = [Dog new];
    [jd run];
```

ARC单个对象内存管理

本小节知识点：
1、【掌握】ARC下单对象内存管理
2、【掌握】强弱指针

在ARC机制下，对象没有强指针指向，会被立即释放

注意释放池的存在

```
    __weak Car *bwm2 = bigBen;
    NSLog(@"bigBen = %@,bwm2 = %@",bigBen,bwm2);
    //bigBen的指向发生改变，对于Car对象来说没有强指针指向了，所以要释放对象

    bigBen = nil;  //
    // 要   1) bigBen 是强指针，重新指向其他内容了，对于对象来说没有强指针了
    //      2) 弱指针 赋值为nil

    NSLog(@"bigBen = %@,bwm2 = %@",bigBen,bwm2);
    [bwm2 run];  //[nil run];

    [bigBen run]; //nil run];

    NSLog(@"xxxxxx");

}
```

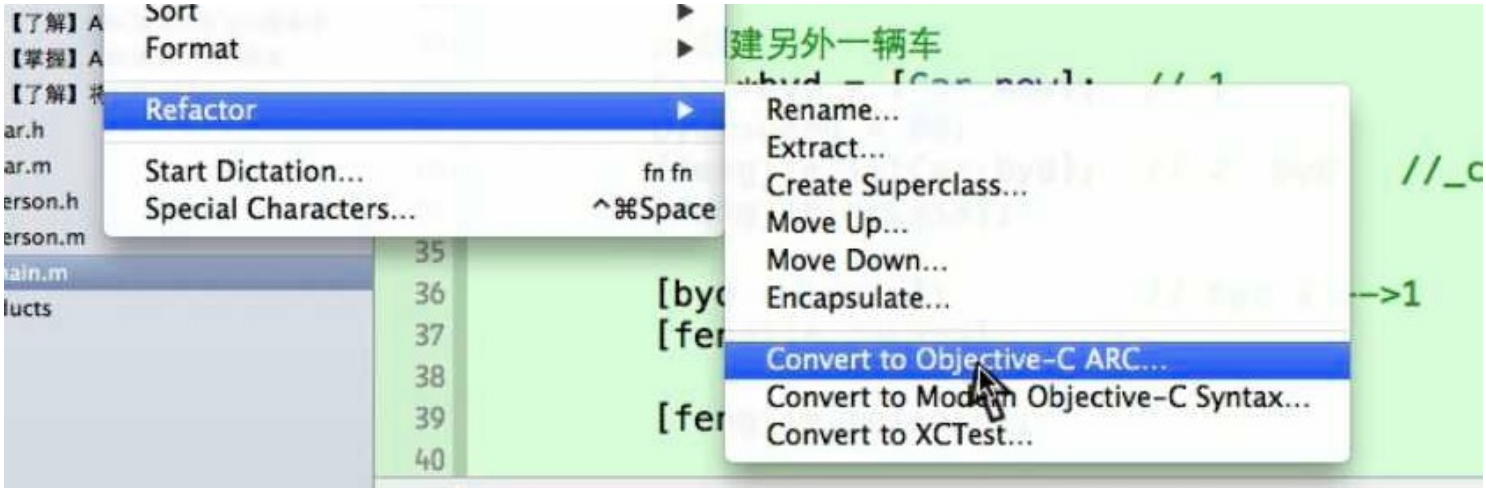
虽然对象已经释放，但两个指针都指向了nil
所以不会直接报错

11-【掌握】ARC的兼容和转换

本小节知识点：

1、【了解】ARC模式下如何兼容非ARC的类

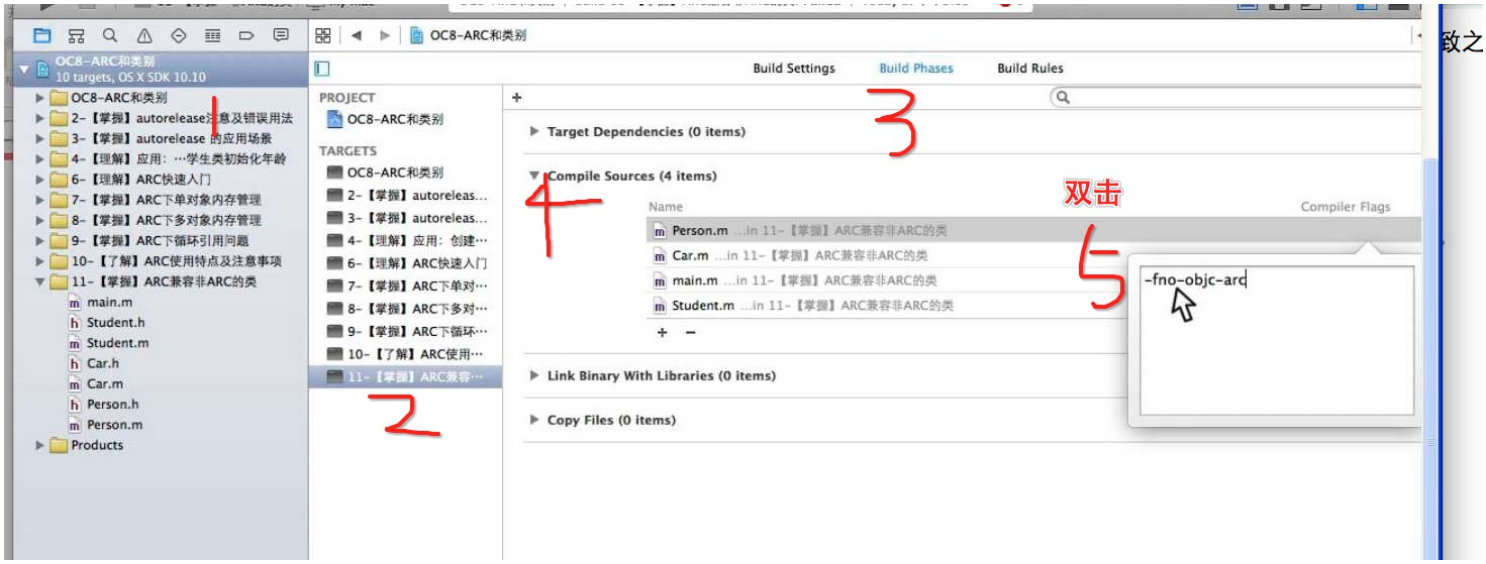
2、【了解】将MRC转换为ARC



不代表所有的项目都能很好的转换好,可能有的转换完后就会导致之前的代码失效,再转换之前需要先提前备份好

1、ARC模式下如何兼容非ARC的类

让程序兼容ARC和非ARC部分。转变为非ARC -fno-objc-arc 转变为ARC的， -f-objc-arc 。



MRC使用转变为非ARC,意思就是再编译时,不按照ARC编译

- 问题：

  1.将arc项目中的mrc文件设置成兼容的指令是？

非ARC _fno_objc_arc

ARC _f_objc_arc