kvo 观察者注意事项

1. 概述：

   ios 中页面间传值的方式有很多种，如：

   直接传值、单例传值、代理传值、block 传值、通知传值、kvo 传值等。

   要说做到真正轻耦合，那么非 kov 监听莫属了。为啥，我可以偷偷的观察某个对象某个值的变化后进行事件处理而不需要动该对象内任何代码。

   But...

   我想说的是，kvo 不是可以随便乱用滴～

2. 分析

   kvo 如果使用不当的话会造成一些意想不到的后果，甚至导致应用闪退。以下作为问题收集区，对使用 kvo 的错误方式进行纠正

   (1) 写了观察者、移除观察者，却没实现以下方法照成崩溃

   observeValueForKeyPath:ofObject:change:context:

   (2) 重复移除了两次观察者,导致崩溃-_-!!!!

   Ok,以上方法的结论如下：

   1) A 对象要通知 B 对象,B 对象必须实现监听的方法,否则一旦有消息发送就会导致崩溃.

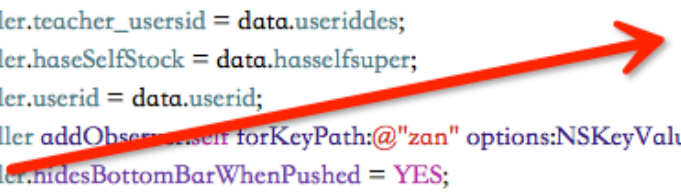
   2) A 对象不想通知 B 对象了,需要从 B 对象身上移除掉通知.

   (3) 当要通知的对象已经 nil 了,这个通知不会自动移除，可能导致崩溃

   很重要：观察者不会自动移除，通知中心并不会 retain 他的观察者，因此，不管是 kvo 还是通知 ，你都必须确保你那些对象销毁之前注销掉观察者,否则就会出现错误

3. 从分析看来，kvo 的移除很重要，那么移除的时机呢？下面举个例子

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath{
    controller = [[LaoShiZiLiaoViewController alloc] initWithNibName:@"LaoShiZiLiaoViewController" bundle:nil];
    ZhaoLaoShiDataModel *data = (ZhaoLaoShiDataModel *)[self.dataModel.data objectAtIndex:indexPath.section];
    self.select_data =  data;
    controller.teacherId = data.Teacher_Id;
    controller.imageUrl = data.AppImgUrl;
    controller.name = data.Name;
    controller.shotInfo = data.ShortInfo;
    controller.info = data.Info;
    controller.zan = data.Zan;
    controller.isYuyue = data.isYuyue;
    controller.zhiYeBianHao = data.codenumber;
    controller.teacher_usersid = data.useriddes;
    controller.haseSelfStock = data.hasselfsuper;
    controller.userid = data.userid;
    [controller addObserver:self forKeyPath:@"zan" options:NSKeyValueObservingOptionNew context:nil];
    controller.hidesBottomBarWhenPushed = YES;
```



点击A页面，我们跳转B页面，并需要观察B页面的一个属性值的变化

移除观察者的时机如果是位于以下两个位置，那么可能出现问题

位置 1：observeValueForKeyPath:ofObject:change:context:里

原因：接收到变化后移除的话，那么就不能监听 B 页面该属性的第二次变化了。

位置 2：dealloc 里

原因 dealloc 是在 B 页面为全局变量的情况下没问题，但此例为局部变量，所以 dealloc 只移除一个观察者，其他的还在，肯定不行

正确的位置：

A 页面 viewDidAppear 方法里，原因很简单，一旦走了这个方法，说明已经又 B 反回 A 页面，B 页面已注销，可以不用监听了

4. 附：kvo 使用正确示例

```
[controller addObserver:self forKeyPath:@"zan" options:NSKeyValueObservingOptionNew context:nil];
controller.hidesBottomBarWhenPushed = YES;


if(controller.haseSelfStock == YES){
    __weak typeof(self) weakSelf = self;
    [FourQuanXianData fourGaoShouZiXuanQuanXianDataRequestUserid:[[NSUserDefaults standardUserDefaults] objectForKey:@"userid"] selfstocksuperid:data.selfstocksuperid
        withArry:^(FourQuanXianModle *model) {
        FourQuanXianModle *quanXianModle=model;
        if ([quanXianModle.type isEqualToString:@"1"]) {
            controller.canSeeTeacherStocks = YES;
        }else{
            controller.canSeeTeacherStocks = NO;
        }
        [weakSelf.navigationController pushViewController:controller animated:YES];
    }];
}else{
    [self.navigationController pushViewController:controller animated:YES];
}


- (void)viewDidAppear:(BOOL)animated{
    [super viewDidAppear:animated];
    [controller removeObserver:self forKeyPath:@"zan"];
}


- (void)observeValueForKeyPath:(NSString *)keyPath ofObject:(id)object change:(NSDictionary<NSKeyValueChangeKey,id> *)change context:(void *)context{
    if([keyPath isEqualToString:@"zan"]){
        self.select_data.Zan = [[change objectForKey:@"new"] integerValue];
    }else
        [super observeValueForKeyPath:keyPath ofObject:object change:change context:context];
}
```