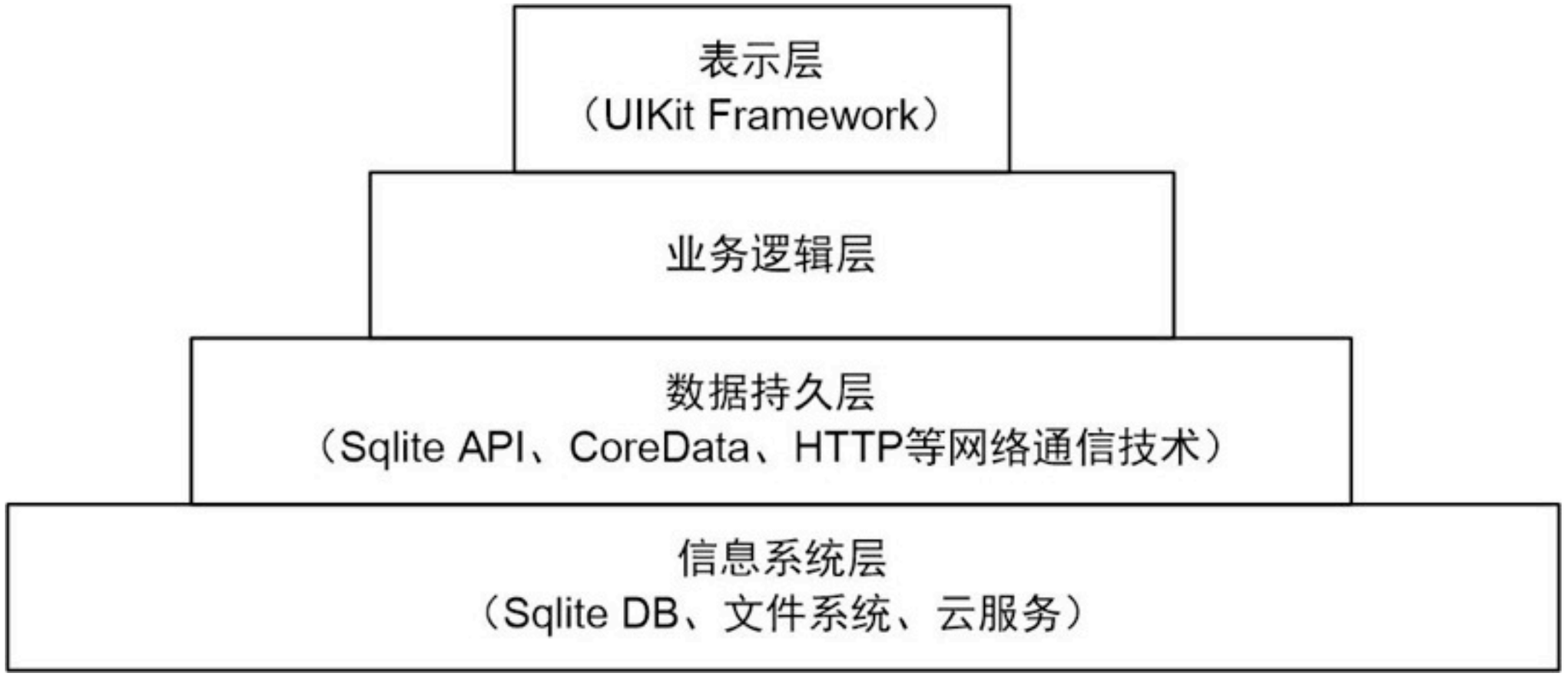
51work6.com

智捷iOS课堂

iOS单元测试最佳实践

iOS单元测试策略

iOS平台信息处理应用分层架构设计


表示层
（UIKit Framework）
业务逻辑层
数据持久层
（Sqlite API、CoreData、HTTP等网络通信技术）
信息系统层
（Sqlite DB、文件系统、云服务）

测试框架与应用分层架构

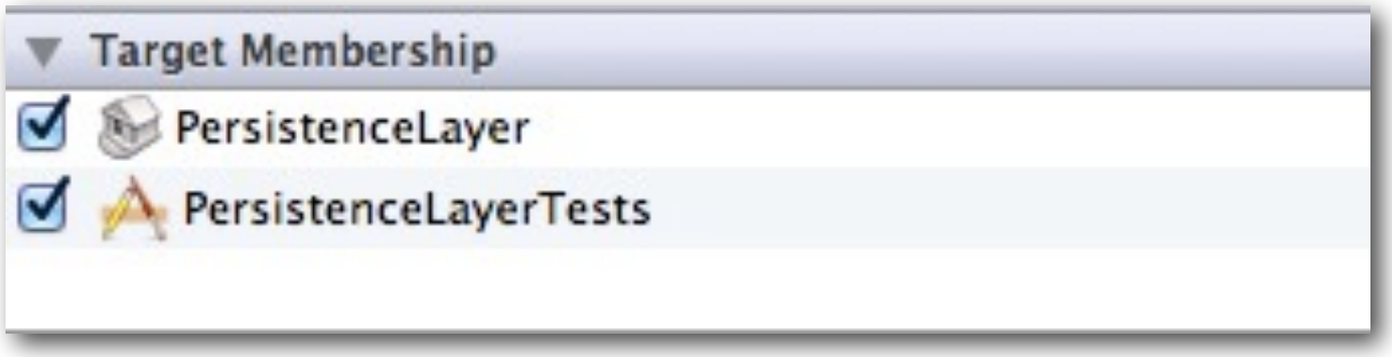
测试框架	表示层	业务逻辑层	数据持久层
OCUnit	逻辑测试、应用测试	逻辑测试	逻辑测试
GHUnit	逻辑测试	逻辑测试	逻辑测试
OCMock	逻辑测试提供伪对象	逻辑测试提供伪对象	逻辑测试提供伪对象

复习：分层架构案例MyNotes

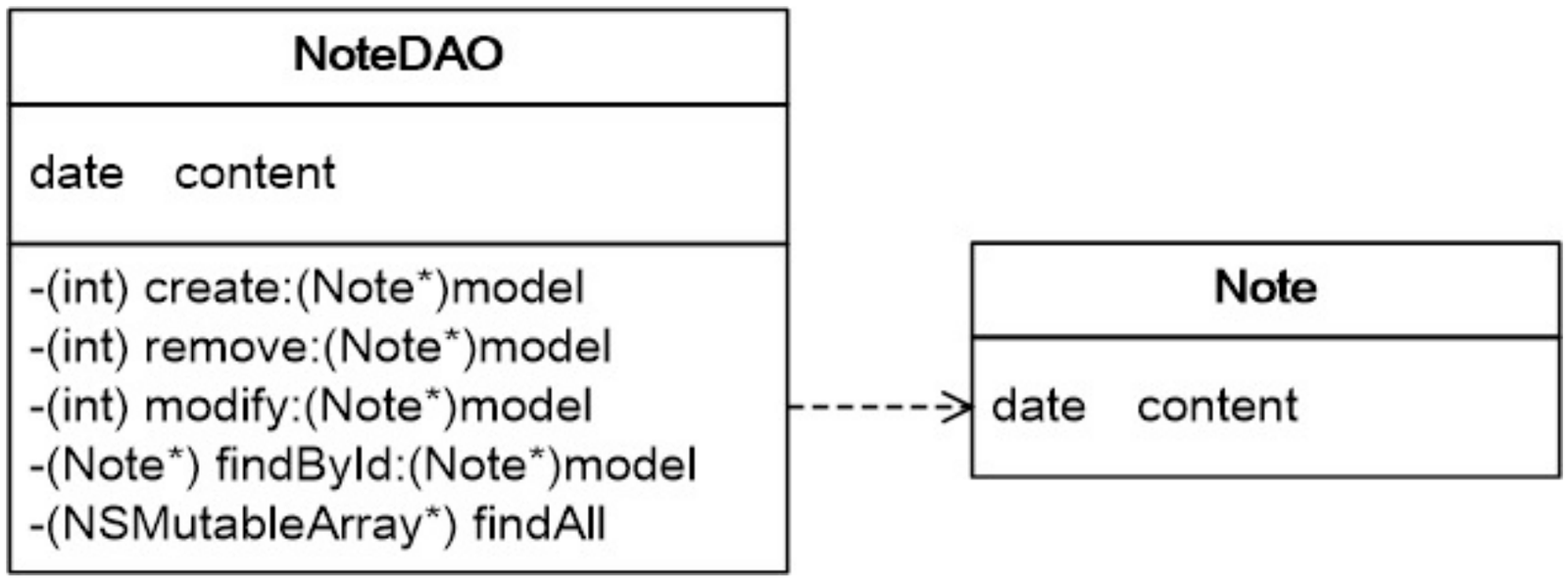
测试数据持久层

创建的测试工程

PersistenceLayerTests



数据持久层类图


NoteDAO
date content
-(int) create:(Note*)model
-(int) remove:(Note*)model
-(int) modify:(Note*)model
-(Note*) findById:(Note*)model
-(NSMutableArray*) findAll
Note
date content

数据持久层测试用例

测试用例	测试方法	输入条件	输出结果
1	`createNote`	`cdate=2000-06-0308:20:38 content= welcome www.51work6.com`	查询记录数为1
2	`findAll`	无	查询记录数为1，第1条记录的`cdate=2000-06-0308:20:38 content= welcome www.51work6.com`
3	`remove`	`cdate=2000-06-0308:20:38`	查询记录数为0

测试类NoteDAOTests

```
#import <GHUnitIOS/GHUnit.h>
#import "NoteDAO.h"
#import "Note.h"

@interface NoteDAOTests: GHTestCase {

}

@property (nonatomic,strong) NSDateFormatter * dateFormatter;
@property (nonatomic,strong) NoteDAO * dao;

@property (nonatomic,strong) NSString* theContent;
@property (nonatomic,strong) NSDate* theDate;

@end
```

```
- (void)setUpClass {
    self.dateFormatter = [[NSDateFormatter alloc] init];
    [self.dateFormatter setDateFormat:@"yyyy-MM-dd HH:mm:ss"];

    self.dao = [NoteDAO sharedManager];

    self.theContent = @"welcome www.51work6.com";
    self.theDate = [self.dateFormatter dateFromString:@"2000-06-03
08:20:38"];
}
```

```
- (void)tearDownClass {
    self.dateFormatter = nil;
    self.dao = nil;
}

- (void)setUp {}

- (void)tearDown { }
```

测试create方法

```
//测试 插入Note方法
-(void) testCreate
{
    //创建Note对象
    Note *note = [[Note alloc] init];
    note.date  = self.theDate;
    note.content = self.theContent;

    int res = [self.dao create:note];
    //断言 无异常，返回值为0，
    GHAssertTrueNoThrow(res == 0, @"数据插入失败");
}
```

根据测试用例要求断言无异常，返回值为0，GHAssertTrueNoThrow(res == 0，@"数据插入失败")语句实现了这个断言。

测试业务逻辑层

创建的测试工程

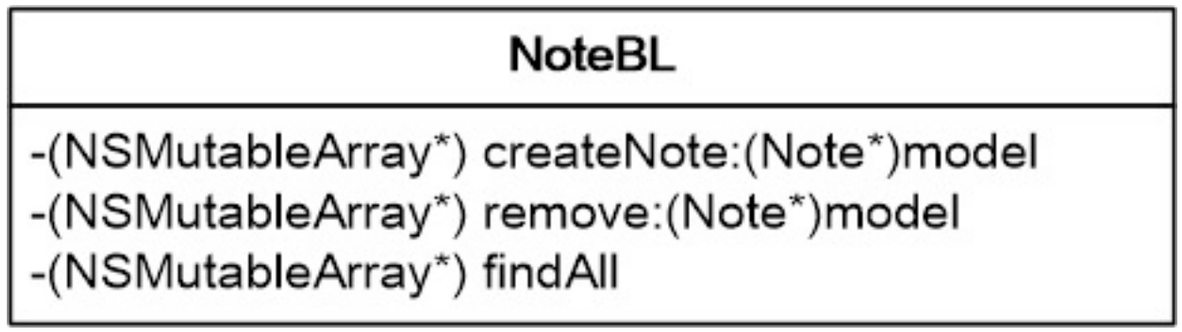
User Header Search Paths

User Header Search Paths是指定用户提供类库的头文件搜索路径，设置编译参数User Header Search Paths，选择TARGETS中的BusinessLogicLayerTests→Build Settings→User Header Search Paths，改为$(BUILT_PRODUCTS_DIR)。

Library Search Paths

Library Search Paths是设定类库搜索路径，这里我们需要清除原来的内容，因为它依赖的类库文件libPersistenceLayer.a在编译的时候会拷贝到BusinessLogicLayerTests编译目录下。清除Library Search Paths内容，选择TARGETS中的BusinessLogicLayerTests→Build Settings→Library Search Paths，删除内容。

业务逻辑层类图


NoteBL
-(NSMutableArray*) createNote:(Note*)model
-(NSMutableArray*) remove:(Note*)model
-(NSMutableArray*) findAll

业务逻辑层测试用例

测试用例	测试方法	输入条件	输出结果
1	`createNote`	`cdate=2000-06-0308:20:38 content= welcome www.51work6.com`	查询记录数为1
2	`findAll`	无	查询记录数为1，第1条记录的`cdate=2000-06-0308:20:38 content= welcome www.51work6.com`
3	`remove`	`cdate=2000-06-0308:20:38`	查询记录数为0

测试类NoteBLTests

```
#import <GHUnitIOS/GHUnit.h>
#import "NoteDAO.h"
#import "Note.h"
#import "NoteBL.h"

@interface NoteBLTests : GHTestCase {

}
@property (nonatomic,strong) NSDateFormatter * dateFormatter;
@property (nonatomic,strong) NoteBL * bl;

@property (nonatomic,strong) NSString* theContent;
@property (nonatomic,strong) NSDate* theDate;

@end
```

```
@implementation NoteBLTests

- (void)setUpClass {
    self.dateFormatter = [[NSDateFormatter alloc] init];
    [self.dateFormatter setDateFormat:@"yyyy-MM-dd HH:mm:ss"];

    self.bl = [[NoteBL alloc] init];

    self.theContent = @"welcome www.51work6.com";
    self.theDate = [self.dateFormatter dateFromString:@"2000-06-03
08:20:38"];
}

- (void)tearDownClass {
    self.dateFormatter = nil;
    self.bl = nil;
}

- (void)setUp { }

- (void)tearDown { }
```

测试create方法

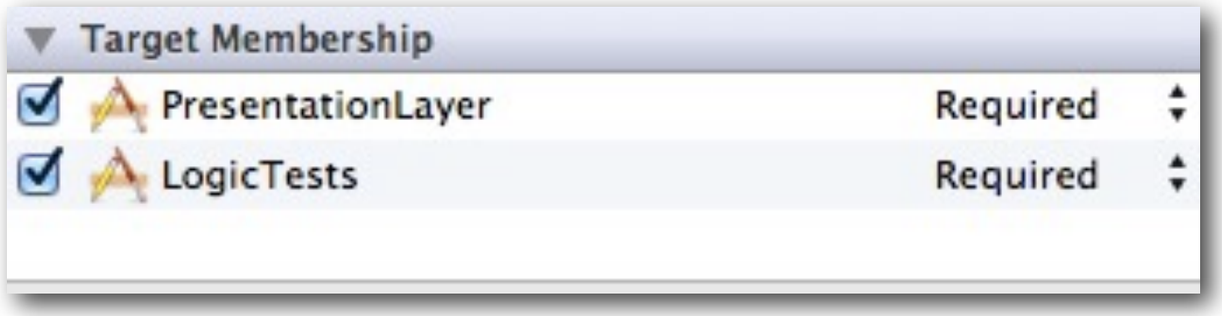
```
//测试 插入Note方法
-(void) testCreateNote
{
    //创建Note对象
    Note *note = [[Note alloc] init];
    note.date  = self.theDate;
    note.content = self.theContent;

    NSArray* list = [self.bl createNote:note];
    //断言 查询记录数为1
    GHAssertTrue([list count] == 1, @"查询记录数期望值为: 1 实际值为: %i", [list
count]);
}
```

测试表示层

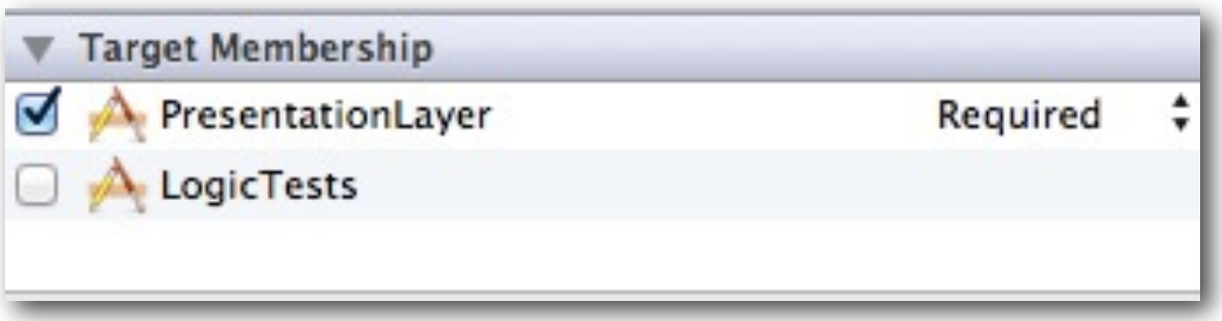
环境的配置

逻辑测试的Target

逻辑测试的Target会用于GHUnit和OCMock



libBusinessLogicLayer.a设置Target Membership

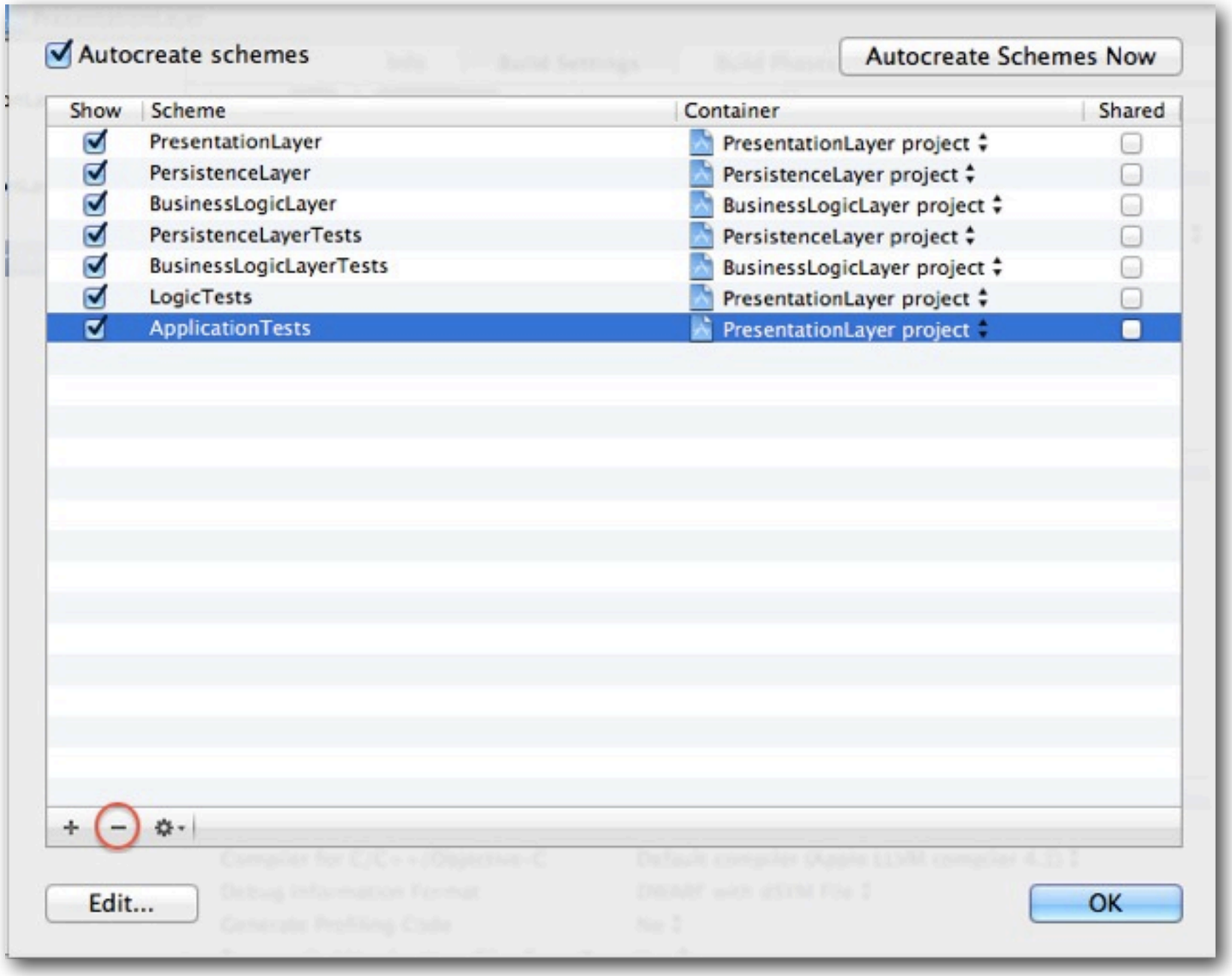


libPersistenceLayer.a设置Target Membership

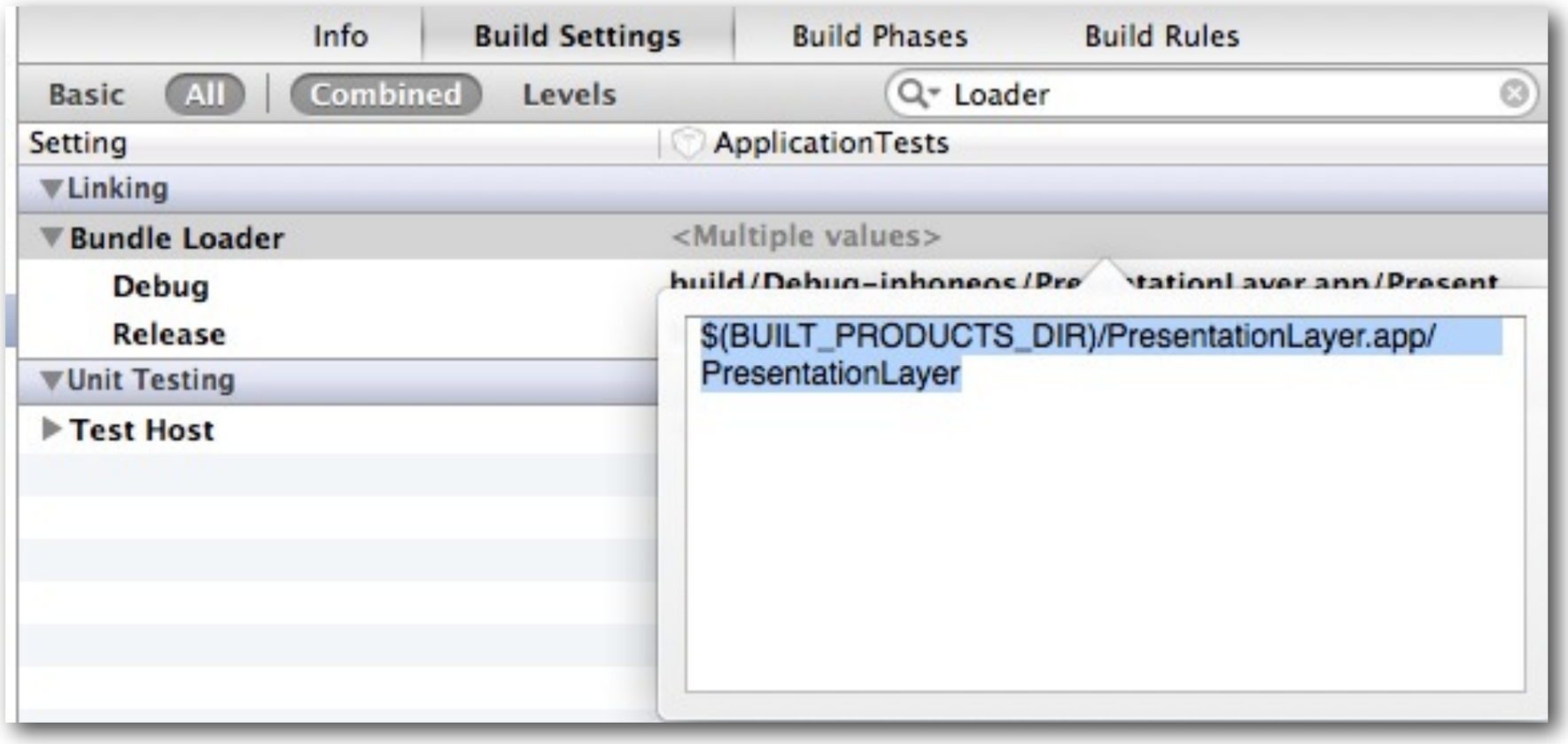
应用测试的Target

应用测试不能使用GHUnit，所以我们只需要添加OCMock到MyNotesWorkspace的工程PresentationLayer中。

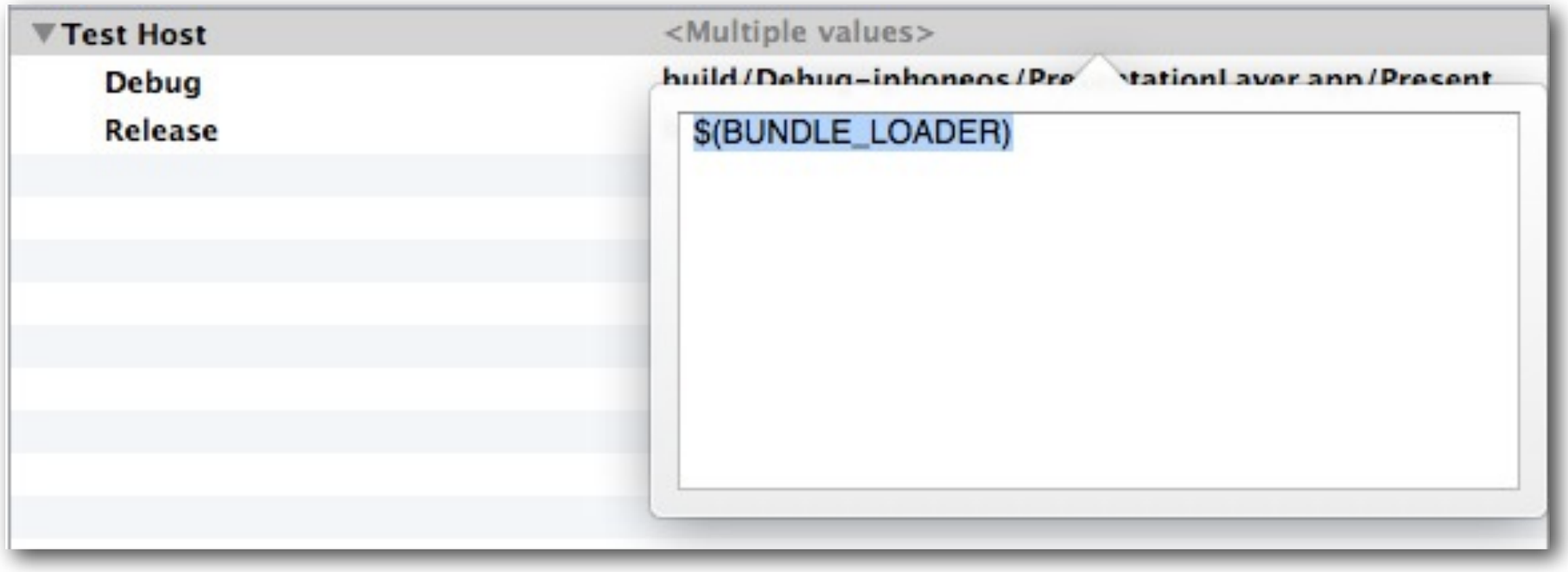
逻辑测试升级为应用测试



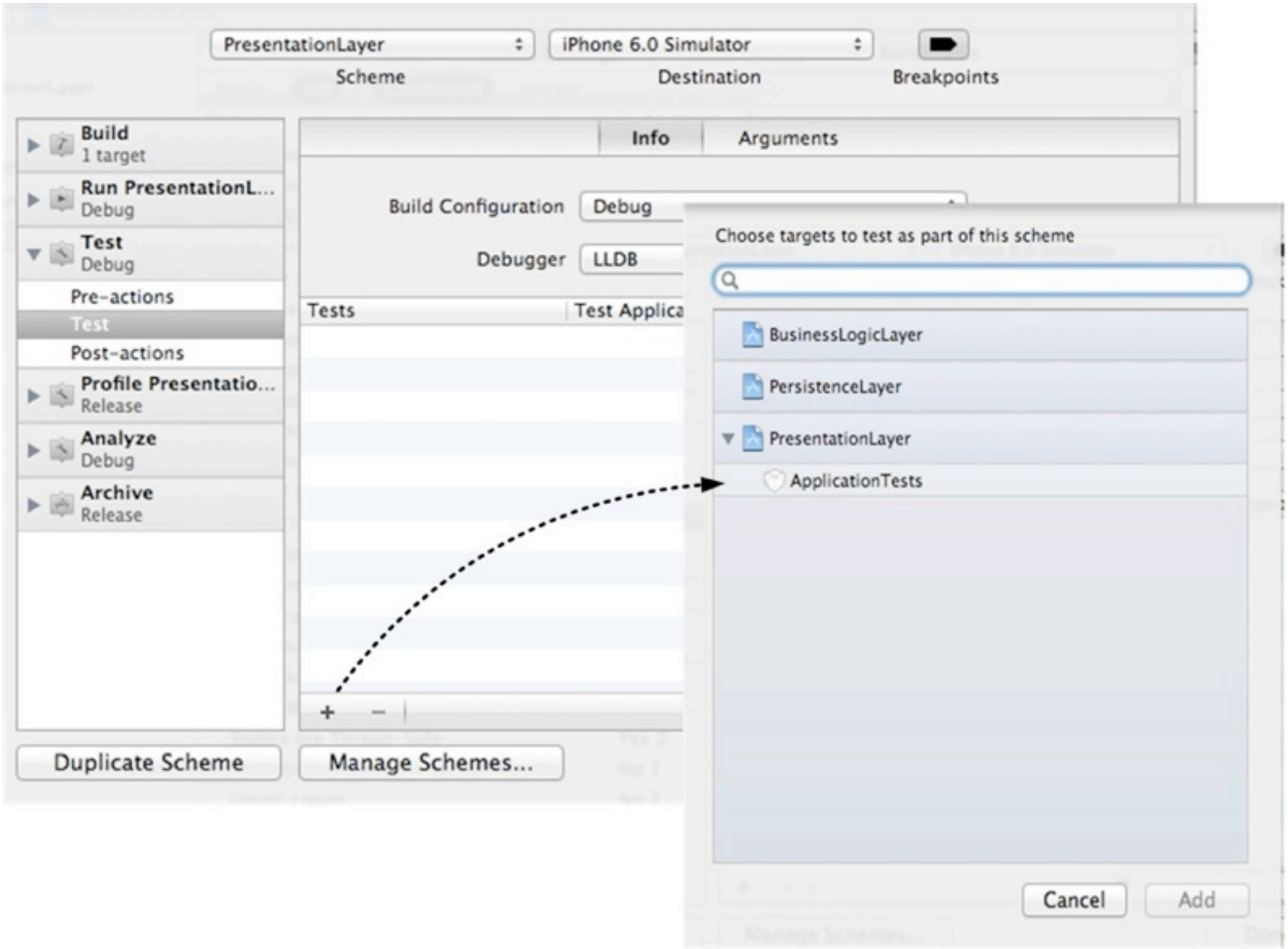
Scheme管理对话框



修改编译参数Bundle Loader



修改编译参数Test Host



修改PresentationLayer Scheme

应用测试

ApplicationTests

```
#import <SenTestingKit/SenTestingKit.h>
#import <OCMock/OCMock.h>
#import "MasterViewController.h"
#import "DetailViewController.h"
#import "Note.h"
#import "AppDelegate.h"

@interface ApplicationTests : SenTestCase

@property (nonatomic,strong) MasterViewController *masterViewController;


@end
```

```
- (void)setUp {
    [super setUp];
    AppDelegate *appDelegate = [[UIApplication sharedApplication] delegate];
    UIWindow *window = [appDelegate window];

    UINavigationController *navController
                    = (UINavigationController*)window.rootViewController;
    self.masterViewController
                    = (MasterViewController*)navController.topViewController;
    //准备测试数据
    NSDateFormatter* dateFormatter = [[NSDateFormatter alloc] init];
    [dateFormatter setDateFormat:@"yyyy-MM-dd HH:mm:ss"];
    //插入测试数据 cdate=2000-06-03 08:20:38 , content=welcome www.51work6.com
    NSString *theContent = @"welcome www.51work6.com";
    NSDate* theDate = [dateFormatter dateFromString:@"2000-06-03 08:20:38"];
    //创建Note对象
    Note *note = [[Note alloc] init];
    note.date  = theDate;
    note.content = theContent;
    [[NoteDAO sharedManager] create:note];
    //插入测试数据 cdate=2001-06-03 08:20:38 , content=www.51work6.com
    theContent = @"www.51work6.com";
    theDate = [dateFormatter dateFromString:@"2001-06-03 08:20:38"];
    //创建Note对象
    note = [[Note alloc] init];
    note.date  = theDate;
    note.content = theContent;
    [[NoteDAO sharedManager] create:note];
}
```

```
- (void)tearDown
{
    [super tearDown];

    //删除测试数据
    NSArray* list =  [[NoteDAO sharedManager]  findAll];
    for (Note* note in list) {
        [[NoteDAO sharedManager]  remove:note];
    }
}
```

测试方法testInitViewController

```
- (void)testInitViewController
{
    //断言 MasterViewController非空
    STAssertNotNil(self.masterViewController, @"MasterViewController非空");
    //断言 MasterViewController中listData属性非空
    STAssertNotNil(self.masterViewController.listData,
@"MasterViewController中listData属性非空");
    //断言 MasterViewController中bl属性非空
    STAssertNotNil(self.masterViewController.bl, @"MasterViewController中bl属
性非空");
}
```

测试方法testMasterViewControllerReturnsCorrectNumberOfRows

```
- (void)testMasterViewControllerReturnsCorrectNumberOfRows
{
    [self.masterViewController.tableView reloadData];
    //断言 表视图返回行数为2
   STAssertEquals(2,
        [self.masterViewController tableView:nil numberOfRowsInSection:0],
        @"应该返回行数为2");
}
```

测试方法testMasterViewControllerTableViewCanEditRowAtIndexPath

```
- (void)testMasterViewControllerTableViewCanEditRowAtIndexPath
{
    //断言 表视图返回行数为2
   STAssertTrue([self.masterViewController tableView:nil
                         canEditRowAtIndexPath:nil], @"应该返回True");
}
```

测试方法testMasterViewControllerReturnsTableViewCell

```
- (void)testMasterViewControllerReturnsTableViewCell
{
    NSIndexPath *indexPath = [NSIndexPath indexPathForRow:0 inSection:0];

  UITableViewCell *cell = [self.masterViewController
tableView:self.masterViewController.tableView
                            cellForRowAtIndexPath:indexPath];
  //断言 单元格应该返回非空
  STAssertNotNil(cell, @"单元格应该返回非空");
    //断言 标签为非空
  STAssertEqualObjects(@"welcome www.51work6.com", cell.textLabel.text, @"标
签为非空");

}
```

测试方法testPrepareForSegue

```
- (void)testPrepareForSegue {
    id mockDetailViewController
                = [OCMockObject mockForClass:[DetailViewController class]];
    id mockStoryboardSegue
                = [OCMockObject mockForClass:[UIStoryboardSegue class]];

    [[[mockStoryboardSegue stub] andReturn:@"showDetail"] identifier];
    [[[mockStoryboardSegue stub] andReturn:mockDetailViewController]
                  destinationViewController];
    NSIndexPath *indexPath = [NSIndexPath indexPathForRow:0 inSection:0];
    Note  *note = self.masterViewController.listData[indexPath.row];
    [[mockDetailViewController expect] setDetailItem:note];
    [self.masterViewController.tableView
                          selectRowAtIndexPath:indexPath animated:NO
                                scrollPosition:UITableViewScrollPositionNone];
    [self.masterViewController prepareForSegue:mockStoryboardSegue
                                                sender:OCMOCK_ANY];
    //验证
    [mockStoryboardSegue verify];
    [mockDetailViewController verify];
}
```

逻辑测试

DetailViewControllerTests

```
#import <GHUnitIOS/GHUnit.h>
#import <OCMock/OCMock.h>
#import "DetailViewController.h"
#import "NoteDAO.h"
#import "Note.h"

@interface DetailViewControllerTests : GHTestCase {
}

@property (nonatomic,strong) NSDateFormatter * dateFormatter;

@property (nonatomic,strong) NSString* theContent;
@property (nonatomic,strong) NSDate* theDate;

@end
```

```
@implementation DetailViewControllerTests

- (void)setUpClass {
    self.dateFormatter = [[NSDateFormatter alloc] init];
    [self.dateFormatter setDateFormat:@"yyyy-MM-dd HH:mm:ss"];

    self.theContent = @"welcome www.51work6.com";
    self.theDate = [self.dateFormatter dateFromString:@"2000-06-03
08:20:38"];
}
- (void)tearDownClass {
    self.dateFormatter = nil;
    self.theContent= nil;
    self.theDate= nil;
}
- (void)setUp { }
- (void)tearDown { }
```

```
//测试 setDetailItem方法
-(void) testSetDetailItem
{

    DetailViewController *detailViewController = [[DetailViewController
alloc] init];

    id mockDetailDescriptionLabel = [OCMockObject mockForClass:[UILabel
class]];

    detailViewController.detailDescriptionLabel =
mockDetailDescriptionLabel;

    [[[mockDetailDescriptionLabel stub] andReturn:self.theContent] text];

    //创建Note对象
    Note *note = [[Note alloc] init];
    note.date  = self.theDate;
    note.content = self.theContent;

    [[mockDetailDescriptionLabel expect] setText:self.theContent];

    [detailViewController setDetailItem:note];
    //验证
    [mockDetailDescriptionLabel verify];
}
@end
```

AddViewControllerTests

```
#import <GHUnitIOS/GHUnit.h>
#import <OCMock/OCMock.h>
#import "AddViewController.h"
#import "NoteDAO.h"
#import "Note.h"

@interface AddViewControllerTests : GHTestCase {
}

@property (nonatomic,strong) NSDateFormatter * dateFormatter;

@property (nonatomic,strong) NSString* theContent;
@property (nonatomic,strong) NSDate* theDate;

@end
```

```
//测试 onclickSave方法
-(void) testOnclickSave
{
    AddViewController *addViewController = [[AddViewController alloc] init];

    id mockTextView = [OCMockObject mockForClass:[UITextView class]];
    addViewController.txtView = mockTextView;
    [[[mockTextView stub] andReturn:@"welcome www.51work6.com"] text];
    [[mockTextView expect] resignFirstResponder];

    //测试 通知
    id mock = [OCMockObject observerMock];
    [[NSNotificationCenter defaultCenter] addMockObserver:mock
                    name:@"reloadViewNotification" object:nil];
    [[mock expect] notificationWithName:@"reloadViewNotification"
                                 object:OCMOCK_ANY userInfo:OCMOCK_ANY];

    [addViewController onclickSave:OCMOCK_ANY];

    //验证
    [mockTextView verify];
    [mock verify];

    [[NSNotificationCenter defaultCenter] removeObserver:mock];
}
```

```
-(void)testTextView
{
    AddViewController *addViewController = [[AddViewController alloc] init];
    id mockTextView = [OCMockObject mockForClass:[UITextView class]];

    [[mockTextView expect] resignFirstResponder];
    BOOL res = [addViewController textView:mockTextView
          shouldChangeTextInRange:NSMakeRange(0,0) replacementText:@"\n"];

    GHAssertFalse(res,@"输入回车符号");

    [mockTextView verify];
}
@end
```

谢谢