单元测试

1. 简述：
2. 在iOS开发中，程序员通常用单元测试来保证代码的可靠性，什么是单元测试？在计算机编程语言中，单元测试又称为模块测试，是针对程序模块的最小单位进行正确性检验的测试工程。程序单元是应用的最小可测试部件。在过程化编程中，一个单元就是单个程序、函数、过程等；对于面向对象编程，最小单元就是方法，包括基类(超类)、抽象类或者派生类(子类)中的方法。有了单元测试，就没必要为了测试某个小模块去编译整个程序，运行后点到相应模块去了。一般来说，写完代码或修改完bug后，需要写单元测试来验证代码是否有问题。
3. 在xcode4时代，集成的是OCUnit，到了xcode5时代就升级为XCTest，并且到了xcode7时代还有了UI测试的能力。新建项目的时候可以为工程选择是否带上单元测试，新建项目多两个目录：TestDemoTests、TestDemoUITests，如果没有，可通过File->New->Targe->Test->iOS Unit Testing Bundle／Ios UI Testing Bundle
4. Tests目录
5. 包含一个.m文件和一个plist文件。.m文件中包含如下方法：
6. setup:每个test方法执行前调用，在这个测试用例里进行一些初始化工作
7. teardown:每个test方法执行后调用
8. testExample:测试方法用例，记得以test开头来命名
9. testPerformanceExample:主要做性能测试，评估一段代码的运行时间，在回调中放置需测试效率的代码
10. 只要点击测试方法左边的菱形按钮，即可运行该测试方法。如果要全部运行，按command＋u即可，或者Product->Test。在测试方法里主要通过断言判断，不适预期的正确结果则该测试方法失败，成功显示绿色对勾，失败显示红色叉叉。断言相关介绍如下：
11. XCTFaile(format...)：生成一个失败的测试
12. XCTAssertNil(a1,format...)：为空判断，a1为空通过，否者不通过
13. XCTAssertNotNil(a1,format...)：不为空判断，a1不为空通过，否则通过
14. XCTAssert(expression,format...)：当表达式为true时通过
15. XCTAssertTrue(expression,format...)：当表达式为true时通过
16. XCTAssertFalse(expression,format...)：当表达式为false时通过
17. XCTAssertEqualObjects(a1,a2,format...)：判断相等，[a1 isEqualTo:a2]为true通过
18. XCTAssertNotEqualObjects(a1,a2,format...)：判断不等，[a1 isEqualTo:a2]为false时通过
19. XCTAssertEqual(a1,a2,format...)：判断相等，当a1,a2为C 语言标量、结构体时
20. XCTAssertNotEqual(a1,a2,format...)：判断不等，当a1,a2为C 语言标量、结构体时通过
21. XCTAssertEqualWithAccuracy(a1,a2,accuracy,format...)：判断相等，当a1,a2位double或float，误差在绝对值accuracy之间时通过
22. XCTAssertNotEqualWithAccuracy(a1,a2,accuracy,format...)：判断不等，当a1,a2为double或float，误差范围在绝对值accuracy之间时通过
23. XCTAssertThrows(expression,format...)：异常测试，当expression发生异常时通过
24. XCTAssertThrowsSpecific(expression,specificException,format...)：异常测试，仅当expression发生specific异常时通过
25. XCTAssertThrowsSpecificNamed(expression,specificException,exception\_name,format...)：异常测试，异常名一致时通过
26. XCTAssertNotThrows(expression,format...)：异常测试，当expression没发生异常时通过
27. XCTAssertNotThrowsSpecific(expression,specificException,format...)：异常测试，当expression没发生specificException异常时通过
28. XCTAssertNotThrowsSpecificNamed(expression,specificException,exception\_name,format...)：异常测试，当expression没有发生指定异常名的异常时通过
29. 测试用例文件名约定：以类名起始，添加Tests结束
30. 单元测试的目标是模型文件，用处：不基于界面UI的情况下，保证模型类的逻辑正确
31. 测试用例：

-(void)testAge{//测试年龄范围在18-60岁的人，失败说明程序没做好范围判断

Person \*p = [[Person alloc] init];

//在单元测试中一定要有边界的测试

//测试情况1:小于18岁的人不能使用某些功能->person.age>=18

p.age = 18;

XCTAssertTrue(p.age>=18,@”年龄应该大于或等于18岁”);

//测试情况2:大于60岁的人不能使用某些功能->person.age<=60

p.age = 61;

XCTAssertTrue(p.age<=60,@”年龄应该不大于60岁”);  
}

-(void)testName{

Person \*p = [[Person alloc] init];

//姓名不能为空

p.name = @”a”;

XCTAssertFalse(p.name == nil,@”用户姓名不能为空”);

XCTAssertTrue(p.name.length>2,@”用户姓名长度至少三个字符”);  
}

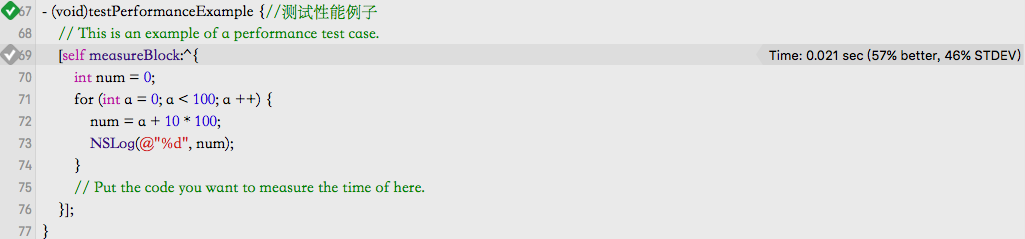
-(void)testClassMethod{

Person \*p = [Person personWithName:@”zhangsan” age:18];

XCTAssertTrue([p.name isEqualToString:@”zhangsan”,@”姓名正确赋值”]);

XCTAssertEqual(p.age,18,@”年龄正确赋值”);  
}

1. 测试方法性能



能够非常直观的看出其调用时间，可以用来对比性能的优劣

1. 异步单元测试

