

2012년 자바카페 OPEN세미나

안드로이드 애플리케이션이 Unit Test를 만날때

2012. 6. 16

유 동 환

koreacio@gmail.com

Java Developer's Forum

JavaCafe
community

Today Story ...



1. SW개발 vs 테스트
2. Unit Test는 무엇인가?
3. Android Unit Test 실습
4. Android Unit Test 알아보기
5. 간단한 Logic 테스트하기
6. 자동화 UI 테스트 소개(Robotium)
7. 기타 고급 주제

Java Developer's Forum

JavaCafe
Community

I. SW개발 vs 테스트

- 요구사항 분석 → 설계 → 프로그래밍 → 테스트 → 유지보수 → 유지보수 → 유지보수
- 그런데 테스트 많이 하시나요?
 - 당장 오늘까지 해결해야 할 Bug가 산더미 같은데?
 - 당장 다음주까지 구현해야 할 Function이 많은데?
 - 예전에 문제없던 Regression Bug가 부메랑이 된다면?
 - 오늘 내가 구현한 것은 향후에는 문제가 없을까?
 - 오늘 내가 만든 코드는 다른 사람이 잘 이해할 수 있나?



- 내가 만든 Class / Method를 테스트 해봅시다.
- 내가 만든 Activity를 테스트해봅시다.
- Test를 위한 Test Project를 만들어 봅시다.
- Test가 잘 돌아가는지 확인해봅시다.

*실전에서
테스트를 생활화하기는
쉽지 않습니다!!*

Java Developer's Forum

JavaCafe
Community

II. Unit Test는 무엇인가?



- 유닛 테스트(unit test)는 컴퓨터 프로그래밍에서 소스 코드의 특정 모듈이 의도된 대로 정확히 작동하는지 검증하는 절차이다. 즉, 모든 함수와 메소드에 대한 테스트 케이스 (Test Case)를 작성하는 절차를 말한다. 이를 통해서 언제라도 코드 변경으로 인해 문제가 발생할 경우 단시간 내에 이를 파악하고 바로 잡을 수 있도록 해준다.
- 유닛 테스트의 목적은 프로그램의 각 부분을 고립 시켜서 각각의 부분이 정확하게 동작하는지 확인하는 것이다. 출처: 위키백과



- 내가 의도한 대로 정확히 검증하기는 어렵다.
- 모든 함수와 메소드에 대해 Test Case를 작성하기는 불가능하다.
- 코드가 변경되면 Test Case가 깨지는 경우도 빈번하다.
- 코드의 각 부분은 서로 꽤나 깊은 상호 의존성을 가진다.

*실전에서는
이런 것 같습니다.*

Java Developer's Forum

JavaCafe
Community

III. Android Unit Test 실습

- 구글에서는 좋은 문서를 제공합니다.

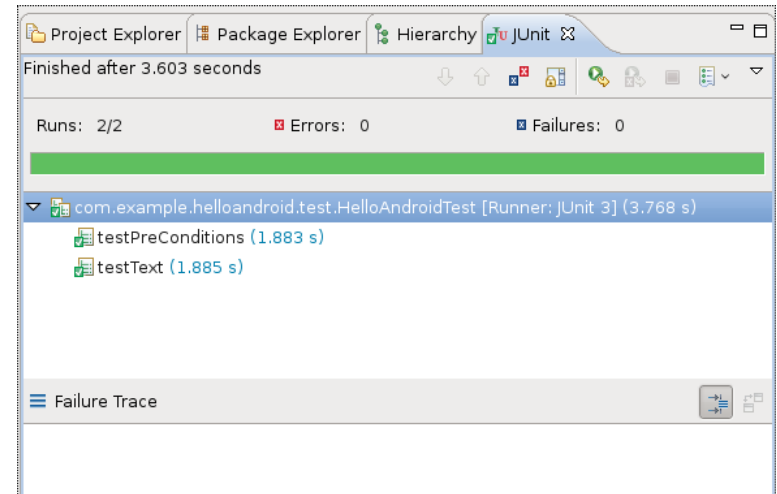
Hello Testing: http://developer.android.com/resources/tutorials/testing/helloandroid_test.html

Activity Testing: http://developer.android.com/resources/tutorials/testing/activity_test.html

- 유닛 테스트 따라하기

1. Test Project 만들기
2. Test Case Class 만들기
 1. 생성자
 2. setUp() 메소드
 3. testPreCondition() 메소드
 4. 실제 test 메소드
3. Test Case 실행
4. 결과 확인

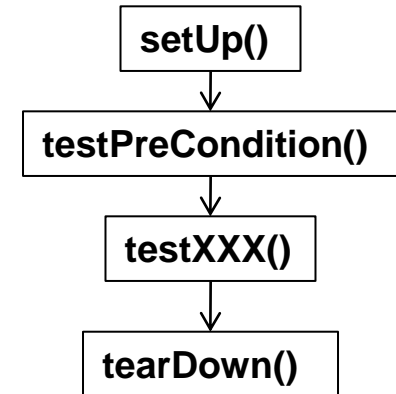
성공!!



IV. Android Unit Test 알아보기



- JUnit 3의 기본 구조를 따릅니다.
- Assert를 사용하여 검증합니다.
 1. assertEquals(expected, actual)
 2. assertEquals(expected, actual)
 3. assertEquals(expected, actual)
 4. assertEquals(actual)
 5. assertEquals(actual)
 6. assertEquals(actual)
 7. assertEquals(actual)
 8. fail()
- ActivityInstrumentationTestCase2<T>를 상속합니다.
 - Activity에 대한 테스트 케이스 만드는 방법



IV. Android Unit Test 알아보기



```
package com.example.helloandroid.test;

import com.example.helloandroid.HelloAndroid;
import android.test.ActivityInstrumentationTestCase2;
import android.widget.TextView;

public class HelloAndroidTest extends ActivityInstrumentationTestCase2<HelloAndroid> {
    private HelloAndroid mActivity; // the activity under test
    private TextView mView; // the activity's TextView (the only view)
    private String resourceString;

    public HelloAndroidTest() {
        super("com.example.helloandroid", HelloAndroid.class);
    }
    @Override
    protected void setUp() throws Exception {
        super.setUp();
        mActivity = this.getActivity();
        mView = (TextView) mActivity.findViewById(com.example.helloandroid.R.id.textview);
        resourceString = mActivity.getString(com.example.helloandroid.R.string.hello);
    }
    public void testPreconditions() {
        assertNotNull(mView);
    }
    public void testText() {
        assertEquals(resourceString, (String)mView.getText());
    }
}
```

상속구조

Set Up

**Pre
Condition**

Assertion

※ 생각보다 복잡하다고 느끼실 것 같습니다.

Java Developer's Forum

JavaCafe
Community

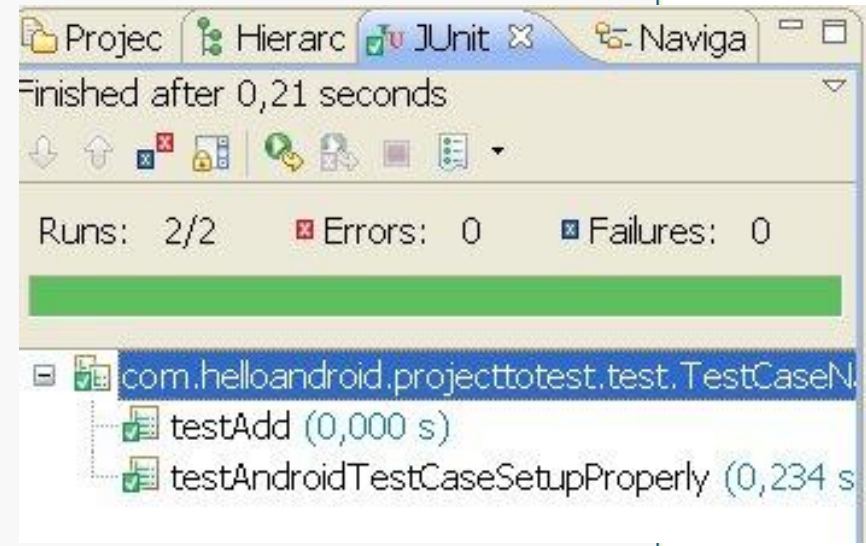
V. 간단한 Logic 테스트하기



```
1 public class ClassToTest{
2     public int add(int arg1,int arg2){
3         return arg1+arg2;
4     }
5 }
```

• Test Case

```
1 public class TestCaseName extends AndroidTestCase {
2     ClassToTest mClassToTest;
3     int mArg1;
4     int mArg2;
5
6     protected void setUp() throws Exception {
7         mClassToTest=new ClassToTest();
8         mArg1=6;
9         mArg2=3;
10        super.setUp();
11    }
12
13    protected void tearDown() throws Exception {
14        super.tearDown();
15    }
16
17    public void testAdd(){
18        assertEquals(9,mClassToTest.add(mArg1,mArg2));
19    }
20 }
```



출처: <http://android10.org/index.php/articlesother/124-unit-testing-example>

Java Developer's Forum

JavaCafe
Community

VI. 자동화UI 테스트 소개(Robotium)



```
public class EditorTest extends
    ActivityInstrumentationTestCase2<EditorActivity> {

    private Solo solo;

    public EditorTest() {
        super("com.test.editor",
            EditorActivity.class);
    }

    public void setUp() throws Exception {
        solo = new Solo(getInstrumentation(), getActivity());
    }

    public void testPreferenceIsSaved() throws Exception {

        solo.sendKey(Solo.MENU);
        solo.clickOnText("More");
        solo.clickOnText("Preferences");
        solo.clickOnText("Edit File Extensions");
        Assert.assertTrue(solo.searchText("rtf"));

        solo.clickOnText("txt");
        solo.clearEditText(2);
        solo.enterText(2, "robotium");
        solo.clickOnButton("Save");
        solo.goBack();
        solo.clickOnText("Edit File Extensions");
        Assert.assertTrue(solo.searchText("application/robotium"));

    }

    @Override
    public void tearDown() throws Exception {
        solo.finishOpenedActivities();
    }
}
```

*마치 사용자가 동작하
는 것처럼 자동화
테스트할 수 있습니다.*

출처: http://code.google.com/p/robotium/wiki/Getting_Started

Java Developer's Forum

JavaCafe
Community

VII. 기타 고급 주제



- 유닛 테스트의 한계 (출처: TDD실천법과 도구)
 1. Thread와 같은 동시성 테스트
 2. Private / Protected 메소드
 3. GUI가 복잡한 경우
 4. 모듈간 의존성 이슈
- 상태기반테스트 vs 행위기반테스트 (출처: TDD실천법과 도구)
 - 상태기반 테스트: 속성 기반
 - 행위기반 테스트: void 메소드는 어떻게 테스트?
- Mock 객체와 Mock 프레임워크
- Android 나머지 4대 컴포넌트 테스트
 - Content Provider
 - Broadcast Receiver
 - Service

극악 난이도:
어떻게 *Unit Test*의
가치를 관리자에게
정당화할 것인가?



- 구글 문서
 - Hello Testing: http://developer.android.com/resources/tutorials/testing/helloandroid_test.html
 - Activity Testing: http://developer.android.com/resources/tutorials/testing/activity_test.html
 - Testing Fundamentals: http://developer.android.com/guide/topics/testing/testing_android.html
- 서적
 - 실용주의 프로그래머를 위한 단위 테스트(인사이트)
 - 테스트 주도 개발 TDD 실천법과 도구(한빛미디어)
- Robotium
 - 주소: <http://code.google.com/p/robotium/>



Thank You!!

koreacio@gmail.com