

# Android Activity のテスト

[Android]

- ・ [http://developer.android.com/guide/topics/testing/activity\\_testing.html](http://developer.android.com/guide/topics/testing/activity_testing.html)

## 概要

- ・ Activity のテストはかなり Android instrumentation framework に依存する
- ・ Activity は他のコンポーネントとは異なり、コールバックメソッドに基づいた、複雑なライフサイクルを持っているため
- ・ これらは、instrumentation 以外から直接呼び出すことができない
- ・ プログラムからイベントをユーザーインターフェースへ送る唯一の方法は、instrumentation を通して

## Activity テスト API

- ・ Activity テスト API の基底クラスは、InstrumentationTestCase
- ・ テストケース派生クラスへ Activity のための操作方法を提供する

### InstrumentationTestCase が提供する機能

#### ライフサイクル制御

- ・ テストケースクラスが提供するメソッドから、Activity を開始でき、一時停止でき、破棄できる

#### 依存性の注入

- ・ Instrumentation は、Contexts や Application のようなシステムオブジェクトのモックを作成することができ、テストで利用できる
- ・ れこは、テスト環境の制御を助け、製品のシステムから隔離する。
- ・ カスタムIntentを用意し、それにより Activity を開始することもできる

#### ユーザーインターフェースインタラクション

- ・ Instrumentation を キーストロークおよびタッチイベントを直接 Activity のユーザーインターフェースに送るために利用できる

## JUnit

Activity テストクラスは、TestCase と Assert クラスを継承した、JUnit フレームワークも提供している

### 2つの主のテスト用サブクラス

- ・ ActivityInstrumentationTestCase2 および、ActivityUnitTestCase
- ・ Activity を通常ではないモードで起動した場合には、SingleLaunchActivityTestCase を利用する

## ActivityInstrumentationTestCase2

- ・ ActivityInstrumentationTestCase2 は、アプリケーションの一つ以上の Activity の機能を通常

- ・のシステム基盤を用いて、テストするためにデザインされている
- ・通常のシステムコンテキストを利用してアプリケーションの通常のインスタンスで Activity を起動する。
- ・モックのインテントを Activity に送信することができるので、
- ・複数のタイプのインテントに対する応答をテストできる
- ・インテントに含まれるデータタイプの確かさを期待する Activity をテストできる

製品のシステムの残りの部分からテストが隔離されていないため、モックの Context および Application は利用できない

## ActivityUnitTestCase

- ・ ActivityUnitTestCase は、隔離された、ただ一つの Activity をテストする。
- ・ Activity を開始する前に、モックの Context または Application ( もしくは両方 ) を注入することができる
- ・そして、 Android と影響し合うことなしに、メソッドのユニットテストを行うことができる
- ・テスト時にモックのインテントを Activity に送ることはできない
- ・しかしながら、 Activity.startActivity(Intent) を呼び出すことはできるし、受け取った引数を確認することが可能

## SingleLaunchActivityTestCase

- ・ SingleLaunchActivityTestCase は、複数のテストの間、変化しない環境の中のただ一つの Activity をテストするのに便利なクラス
- ・ setUp() と tearDown() をメソッドの呼び出ごとに呼び出す代わりに、たった一度だけ呼び出す。
- ・いかなるモックオブジェクトの注入も許可しない

## モックオブジェクトと Activity のテスト

- ・ android.test.mock で定義される モックオブジェクトを Activity のテストに利用する

### MockApplication

- ・ MockApplication は、 ActivityUnitTestCase を使う場合、 Activity のテストに唯一有効。
- ・デフォルトでは、 ActivityUnitTestCase は隠された MockApplication オブジェクトを生成しテストで使用する
- ・独自のオブジェクトを、 setApplication で利用できる

## Activity テストでのアサーション

- ・ ViewAsserts は View のためのアサーションを定義する
- ・ Viewno アイメントや位置、 ViewGroup のステートを確認するのに利用できる

## Eclipse ADT でのテスト

- ・ [http://developer.android.com/guide/developing/testing/testing\\_eclipse.html](http://developer.android.com/guide/developing/testing/testing_eclipse.html)

### テストプロジェクトの作成

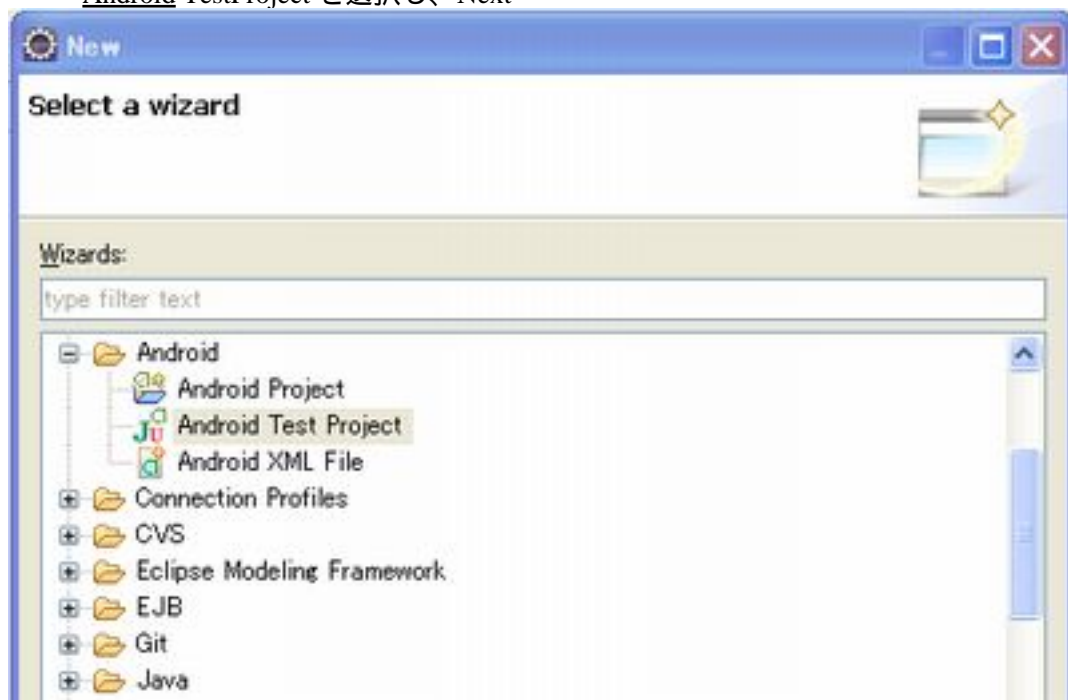
- ・テスト環境を Android アプリケーションにセットアップするには、最初にテストコードを含む分離されたプロジェクトを作成する必要がある

- ・新しいプロジェクトは、Android アプリケーションのディレクトリ構造に従う
- ・同じタイプのコンテンツとファイル、ソースコード、リソース、マニフェストファイルなどを含む
- ・作成したテストパッケージはテスト時にマニフェストファイルの <instrumentation> 要素に従ってアプリケーションに接続する
- ・新規 Android テストプロジェクトダイアログは、それらを簡単に作成するし、いつでもできる
- ・ダイアログは、新しい Android アプリケーション作成直後に表示されるが、事前に作成しておくこともできる

テストプロジェクトを作成するには

File - New - Other

- ・Android TestProject を選択し、Next



### Android Test Project ウィザード

- ・Test Project Name: 任意の名前がつけられるが、対象のプロジェクトに "Test" を付加するのが関連を表す一つの方法
- ・既存の Android プロジェクトを選択すると、必要な項目が自動で設定される

**New Android Test Project**  
Creates a new Android Test Project resource.

Test Project Name:

**Content**  
☒ Use default location  
 Location:

**Test Target**  
 Select the project to test:  
☐ This project  
☒ An existing Android project    
 Test Target Package:

**Build Target**

Target Name	Vendor	Platform	AP...
<input type="checkbox"/> Android 2.1-update1	Android Open Source Project	2.1-updat...	7
<input type="checkbox"/> Google APIs	Google Inc.	2.1-updat...	7
<input checked="" type="checkbox"/> Android 2.2	Android Open Source Project	2.2	8
<input type="checkbox"/> Google APIs	Google Inc.	2.2	8
<input type="checkbox"/> Android 2.3	Android Open Source Project	2.3	9

**Properties**  
 Application name:   
 Package name:   
 Min SDK Version:

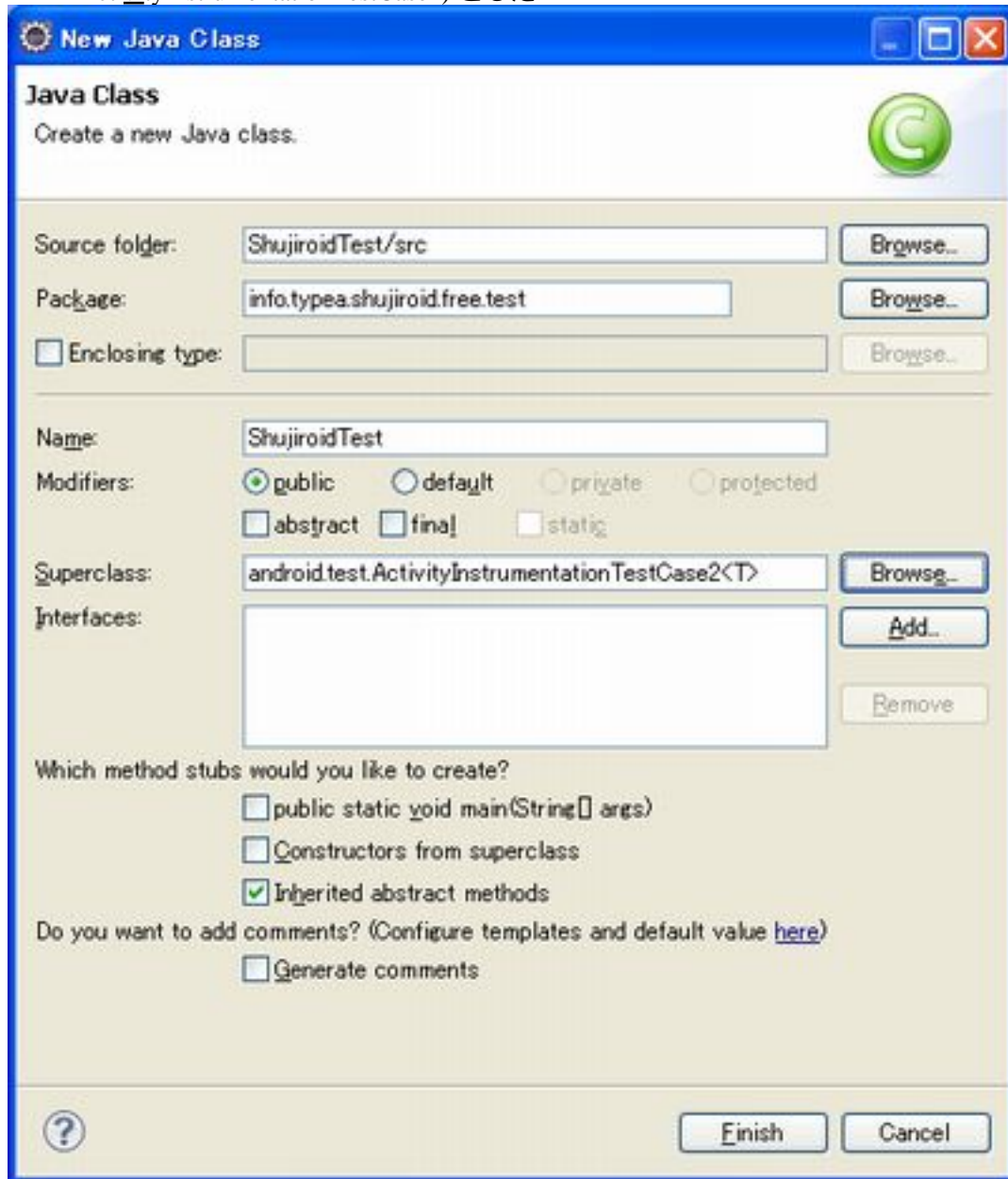
## テストパッケージの作成

- ・テストプロジェクトを作成したら、テストパッケージを作成する
- ・パッケージは、Activity を必須とはしないが、望めば、定義することもできる
- ・テストパッケージは、Activity、テストケースクラスと通常のクラスと一体化可能ではあるが、メインとなるテストケースは Android テストケースクラスの一つ、もしくは JUnit クラスから継承すべきである。なぜなら、ベストなテスト機能を提供するから
- ・テストパッケージは、Android GUI は必要としない
- ・Eclipse ADT から起動する場合、JUnit ビューに結果が表示される
- ・テストを実行し、結果を確認する詳細については、Running Tests を参照

- ・ android.test に Android のテストケースクラスが定義されている
- ・ それらは、JUnit の TestCase クラスを継承している

作成されたテストパッケージのコンテキストメニューから、New - Class

- ・ クラス名は、テストしたい対象のクラス名 + "Test" としておくのが一つの方法
- ・ 継承元は、上記で上げた基底クラスを指定する。(今回は、ActivityInstrumentationTestCase2) とした



- ・ クラスが作成されたら、ActivityInstrumentationTestCase2 のジェネリック型に対象の Activity を指定する。
- ・ テスト環境を制御するためには、setUp() と tearDown() メソッドをオーバーライドする

setUp()

- ・ どのテストメソッドが呼ばれる前に呼ばれる
- ・ テスト用の環境をセットするのに使用する

- ・ ACTION\_MAIN を伴った新しいインテントを生成するのに利用できる

tearDown()

- ・ すべてのテストメソッドの後に呼び出される
- ・ ガベージコレクションおよびテスト用の設定のリセットに利用する

例

```
package info.typea.shujiroid.free.test;

import info.typea.shujiroid.core.ShujiView;
import info.typea.shujiroid.free.R;
import info.typea.shujiroid.free.ShujiActivity;
import android.test.ActivityInstrumentationTestCase2;
import android.test.TouchUtils;
import android.view.Gravity;

public class ShujiroidTest extends
    ActivityInstrumentationTestCase2<ShujiActivity> {
    private ShujiView shujiView;

    public ShujiroidTest() {
        super("info.typea.shujiroid.free", ShujiActivity.class);
    }

    @Override
    protected void setUp() throws Exception {
        super.setUp();
        shujiView = (ShujiView) getActivity().findViewById(R.id.shujiview);
    }

    public void testDraw() {
        getActivity().runOnUiThread(new Runnable() {
            @Override
            public void run() {
                shujiView.requestFocus();
            }
        });
        TouchUtils.dragViewBy(this, shujiView, Gravity.CENTER, 500, 200);
    }
}
```

## ユーティリティ

- ・ Assert: JUnit のアサートクラス
- ・ MoreAsserts: Android の追加アサートクラス
- ・ ViewAsserts: View テスト用に便利なアサートクラス
- ・ TouchUtils: タッチイベントシミュレーション用

## テストの実行

- ・ テストを Eclipse から実行するには、2 つの方法がある
  - ・ Run As ... - Android JUnit Test をプロジェクトのコンテキストメニューから選択
  - ・ Eclipse の run configuration を作成する



- ・自動で、画面に描画された