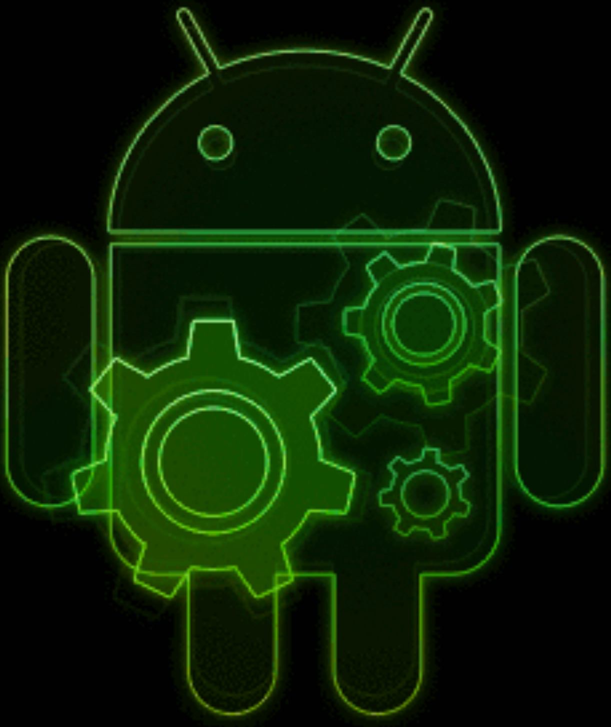


Android Automated Testing

Chuck Greb (@ecgreb)
Senior Mobile Developer
HowAboutWe.com

Why Test?

Validate Requirements

Ensure Quality

Reduce Cost

Why Unit Test?

Test smallest possible units of code
(in isolation)

Makes refactoring easier
(regression suite)

Self-documenting code

Fakes

Mocks

Stubs

Why TDD?

(Test-Driven Development)

Improves Architecture

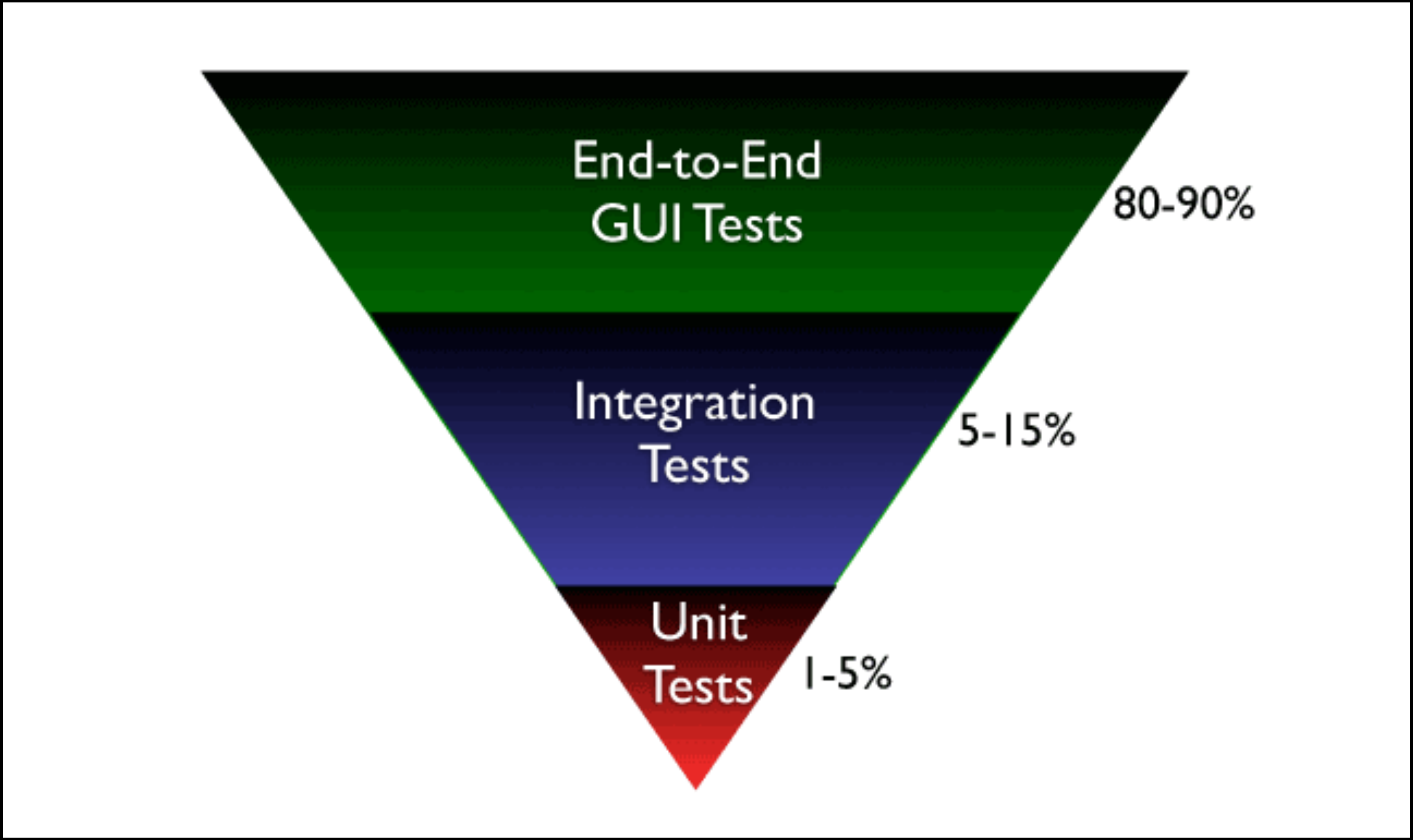
Reduces debugging time

Red -> Green -> Refactor

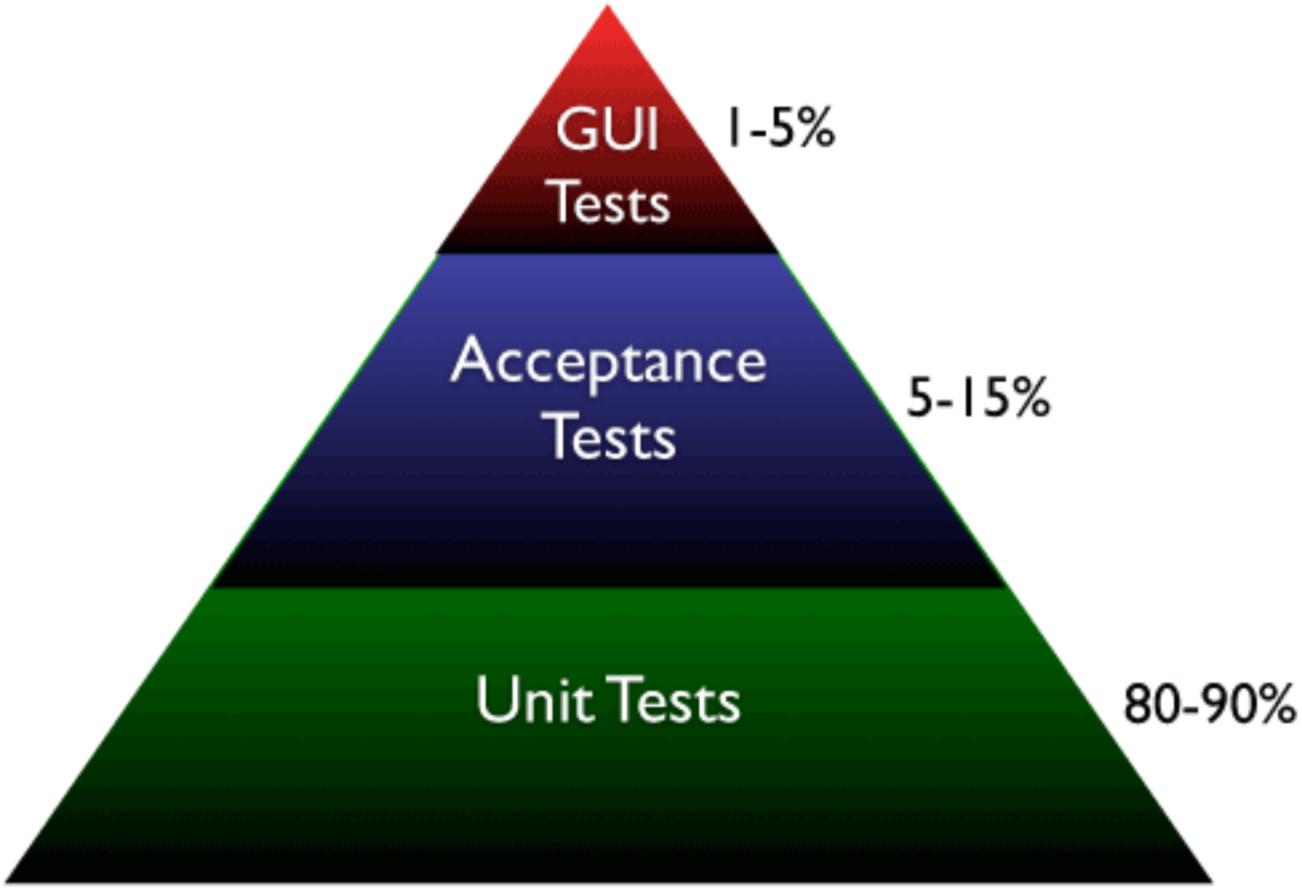
Tests must be fast!

Other kinds of tests?

Inverted Testing Pyramid



(Un-Inverted) Testing Pyramid

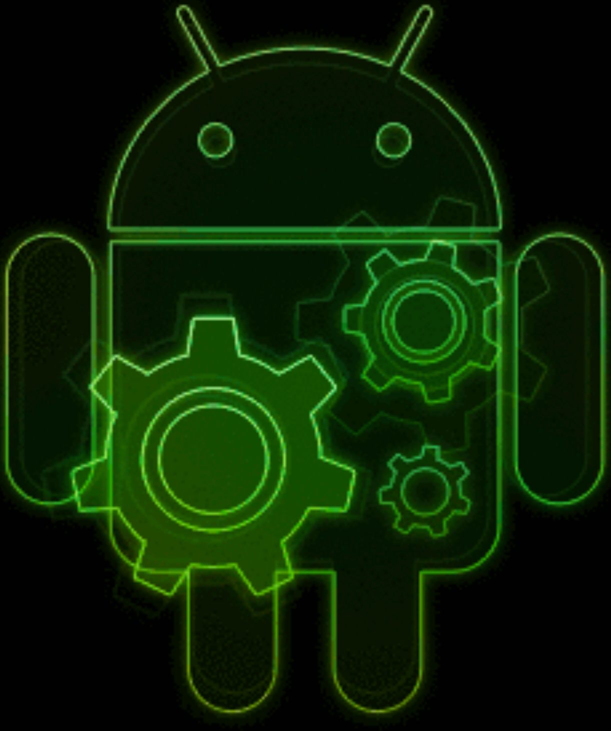


QA Job Security

Unit Testing is Awesome!!

Testing Approaches

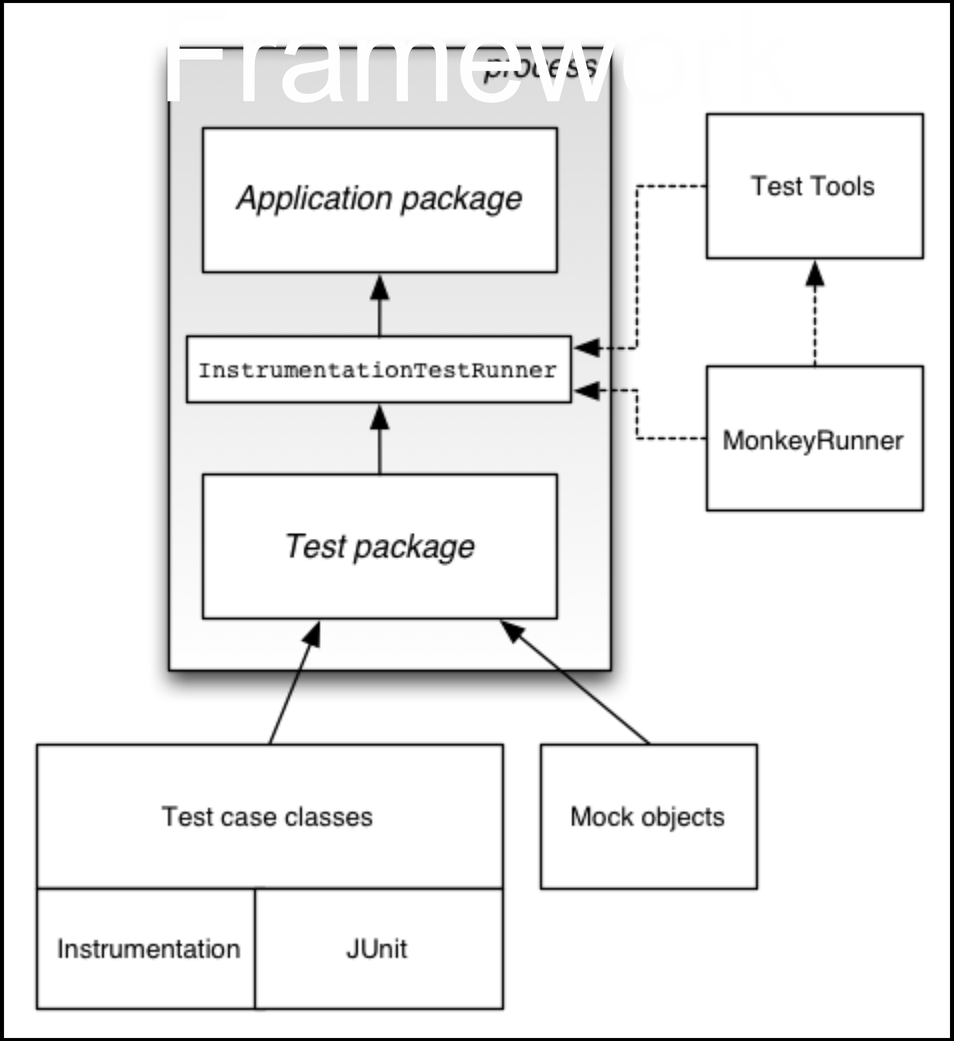
- F@#% It!
- Manual
- Android Testing Framework
- JUnit 4 + POJOs
- Robolectric
- Other



Android Testing Framework

JUnit 3 + Instrumentation

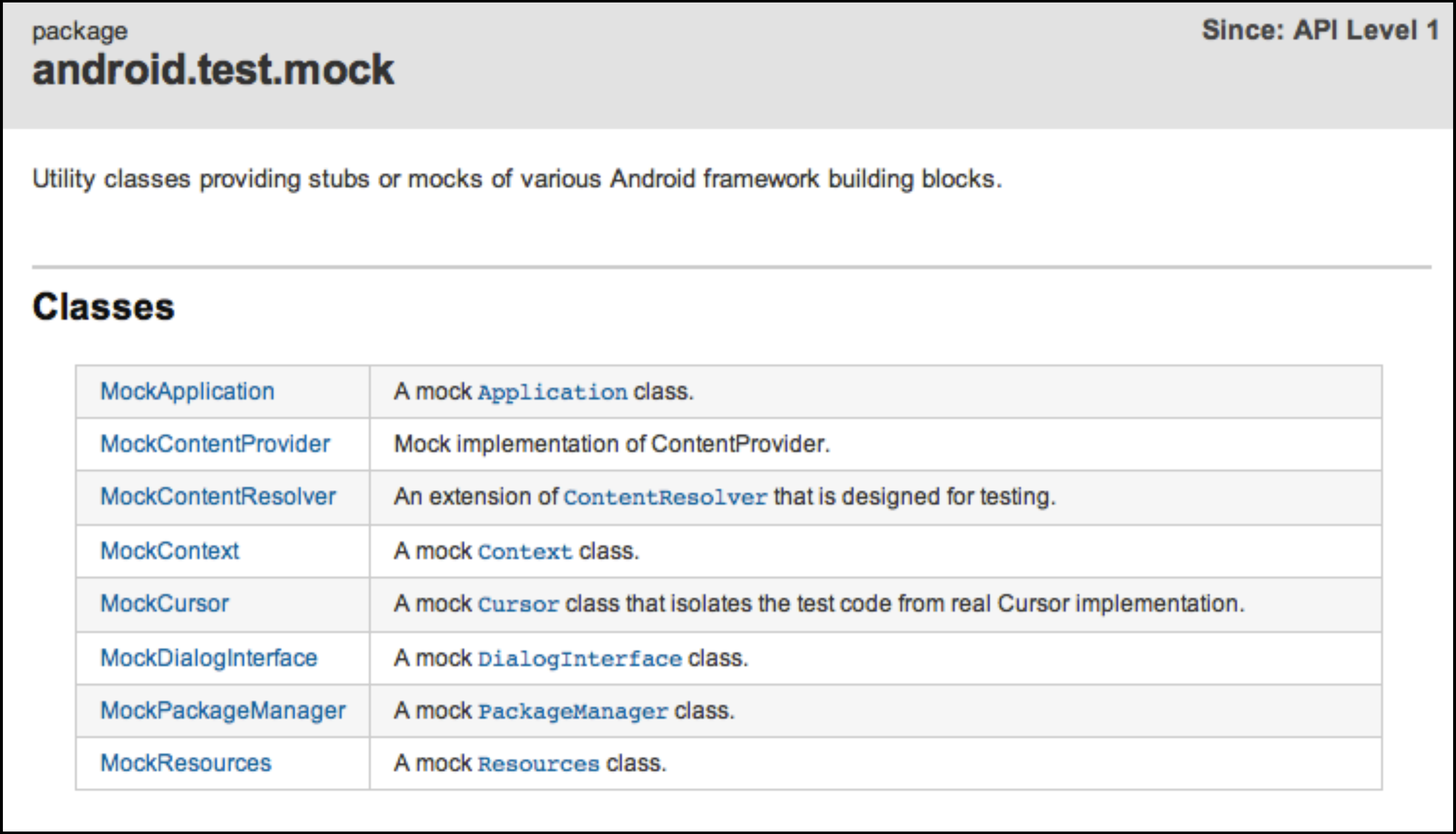
Android Testing Framework



Test Case Classes

- TestCase
- AndroidTestCase
- ActivityTestCase
- ActivityUnitTestCase
- ServiceTestCase
- ProviderTestCase2
- ActivityInstrumentationTestCase2

Android Mocks



package
android.test.mock

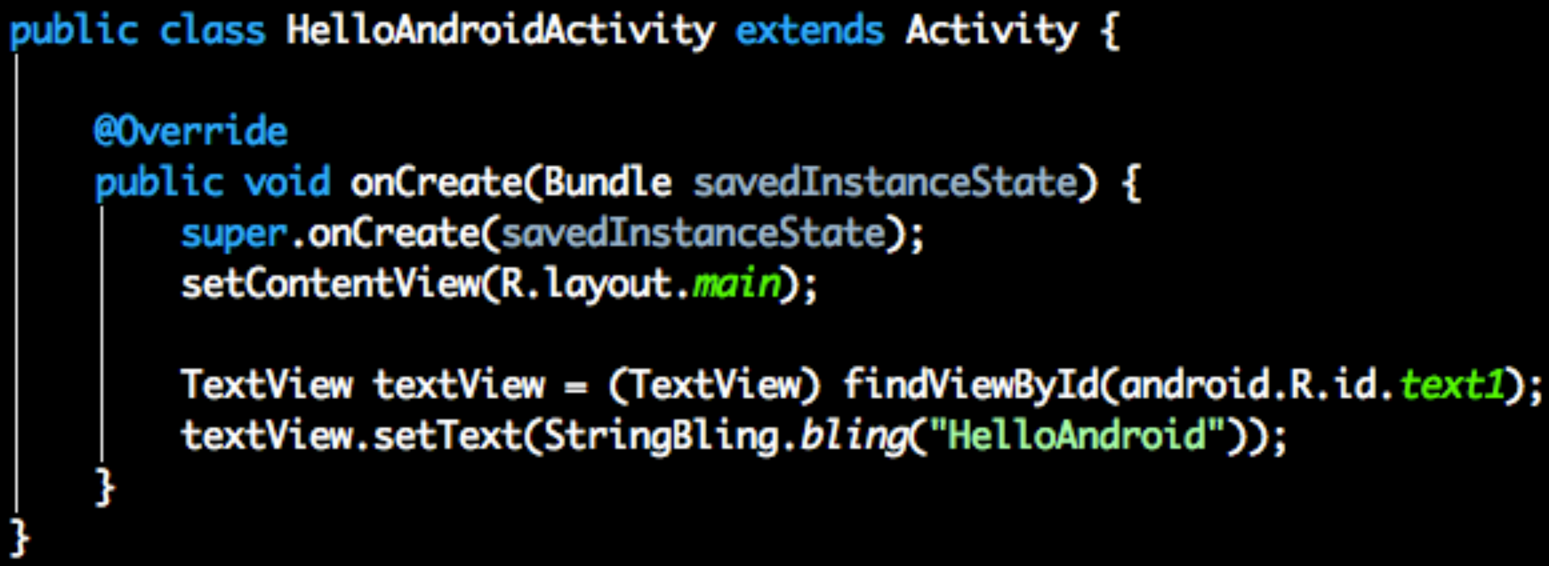
Since: API Level 1

Utility classes providing stubs or mocks of various Android framework building blocks.

Classes

MockApplication	A mock `Application` class.
MockContentProvider	Mock implementation of ContentProvider.
MockContentResolver	An extension of `ContentResolver` that is designed for testing.
MockContext	A mock `Context` class.
MockCursor	A mock `Cursor` class that isolates the test code from real Cursor implementation.
MockDialogInterface	A mock `DialogInterface` class.
MockPackageManager	A mock `PackageManager` class.
MockResources	A mock `Resources` class.

HelloAndroidActivity

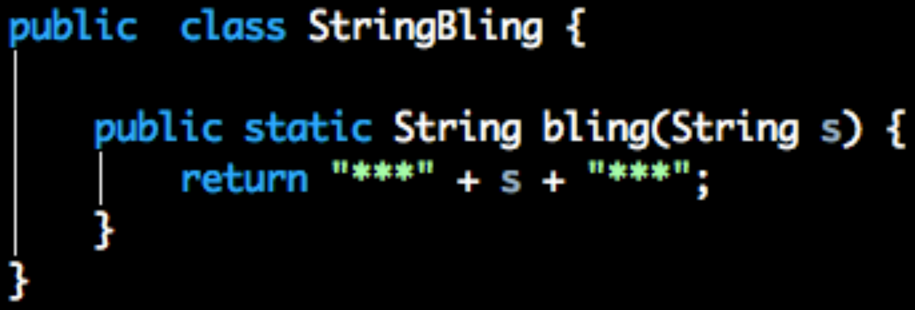


```java
public class HelloAndroidActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        TextView textView = (TextView) findViewById(android.R.id.text1);
        textView.setText(StringBling.bling("HelloAndroid"));
    }
}
```

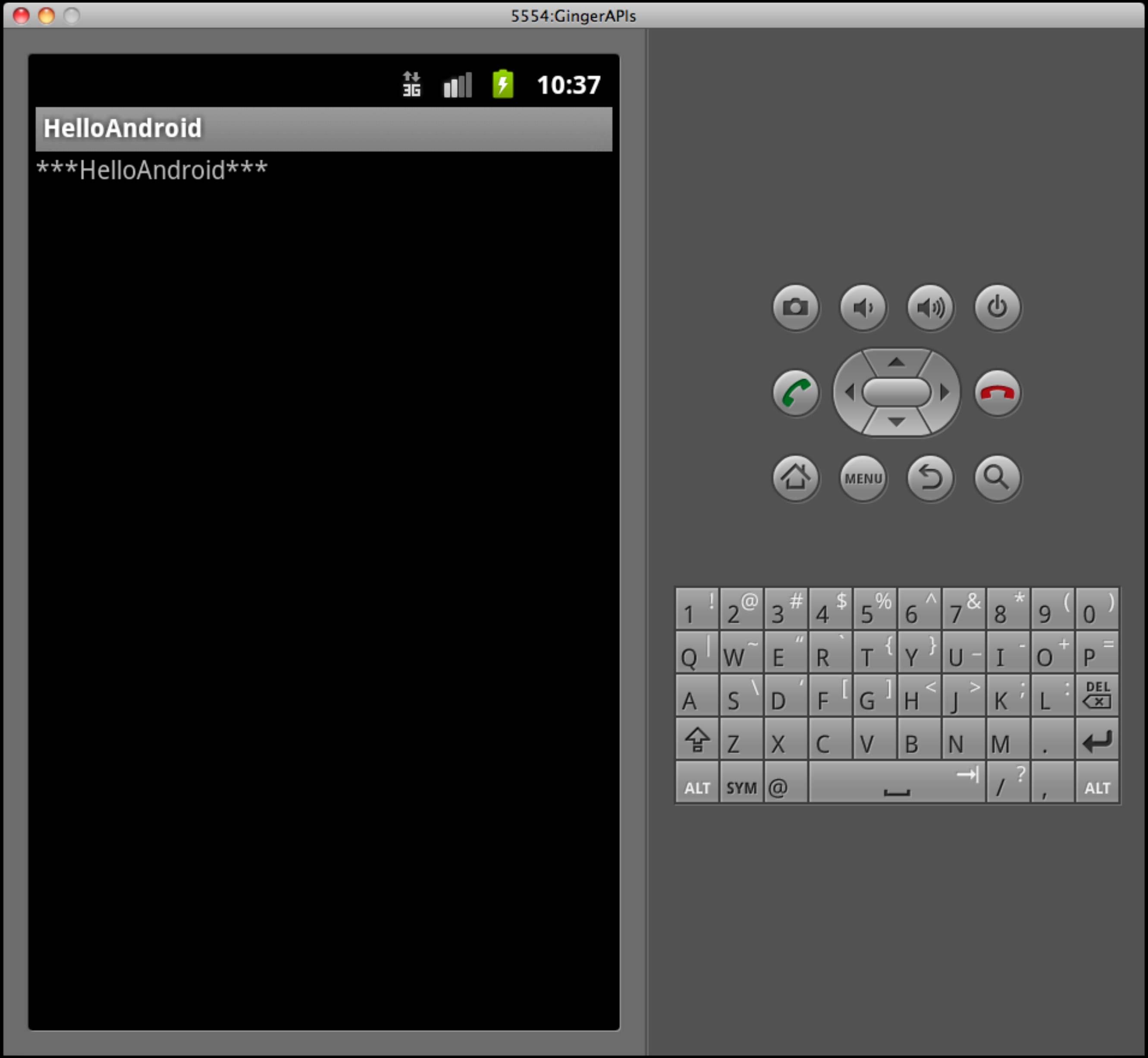
main.xml

```xml
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <TextView
        android:id="@android:id/text1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
```

StringBling



```
public  class StringBling {

    public static String bling(String s) {
        return "***" + s + "***";
    }
}
```

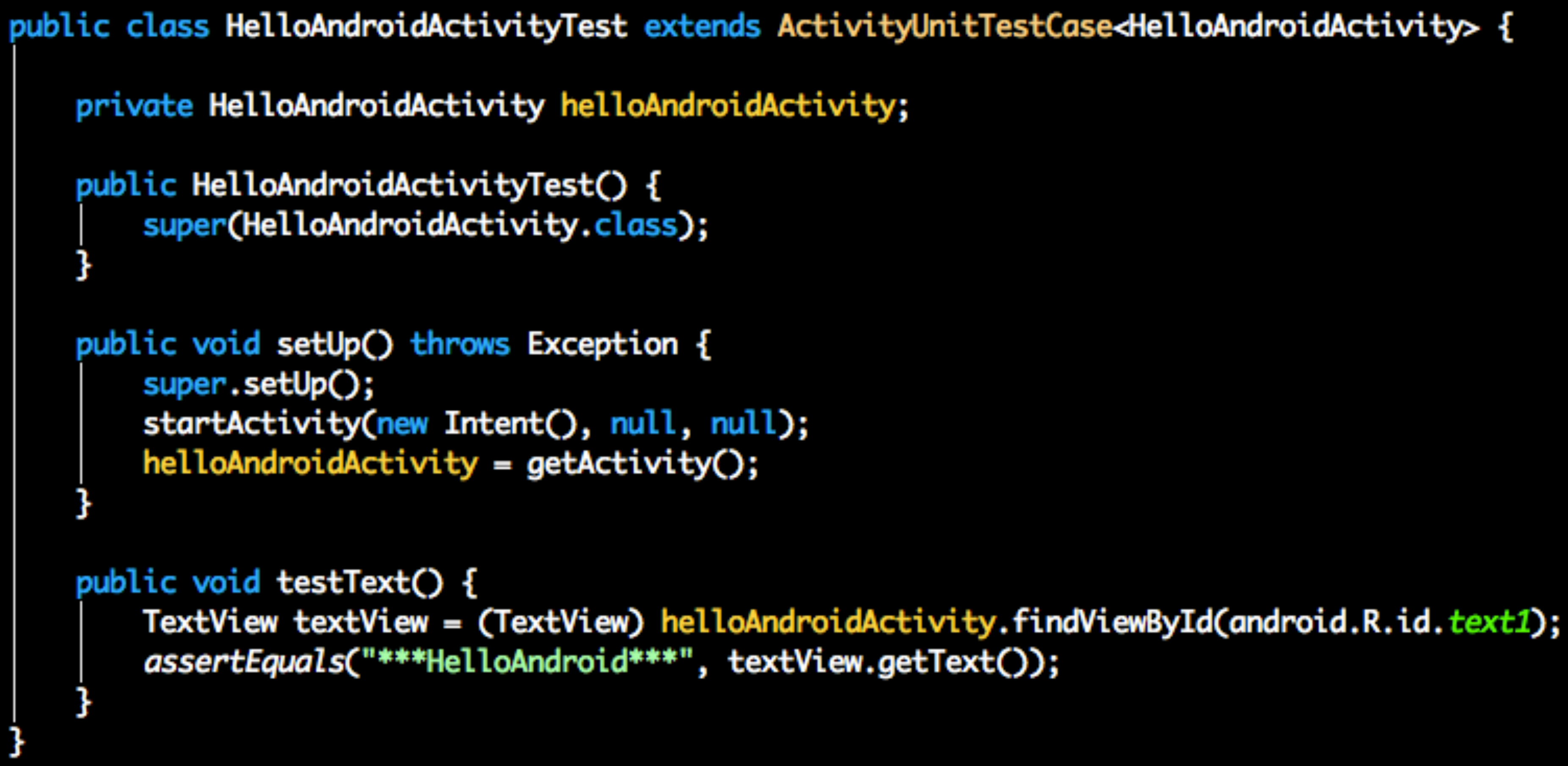

5554:GingerAPIs

10:37

HelloAndroid

HelloAndroid

HelloAndroidActivityTest



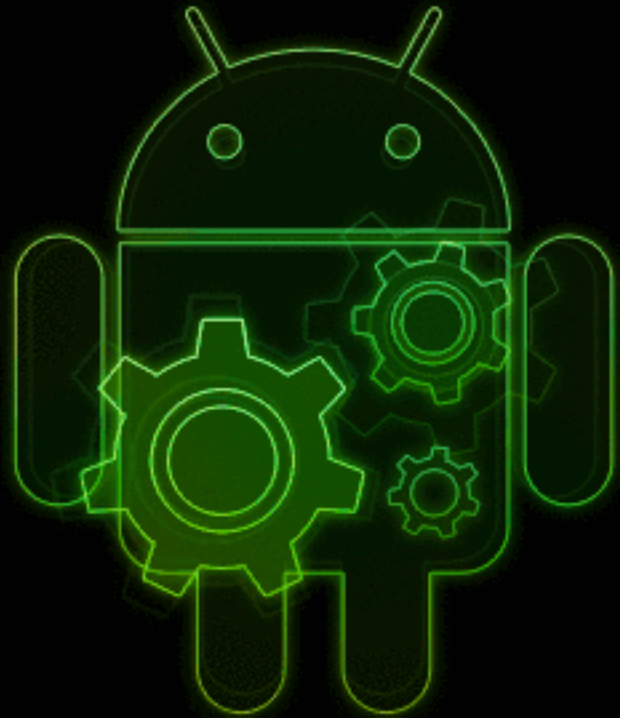
```java
public class HelloAndroidActivityTest extends ActivityUnitTestCase<HelloAndroidActivity> {

    private HelloAndroidActivity helloAndroidActivity;

    public HelloAndroidActivityTest() {
        super(HelloAndroidActivity.class);
    }

    public void setUp() throws Exception {
        super.setUp();
        startActivity(new Intent(), null, null);
        helloAndroidActivity = getActivity();
    }

    public void testText() {
        TextView textView = (TextView) helloAndroidActivity.findViewById(android.R.id.text1);
        assertEquals("***HelloAndroid***", textView.getText());
    }
}
```

Success!!!

Dexing, packaging, and installation on emulator or device

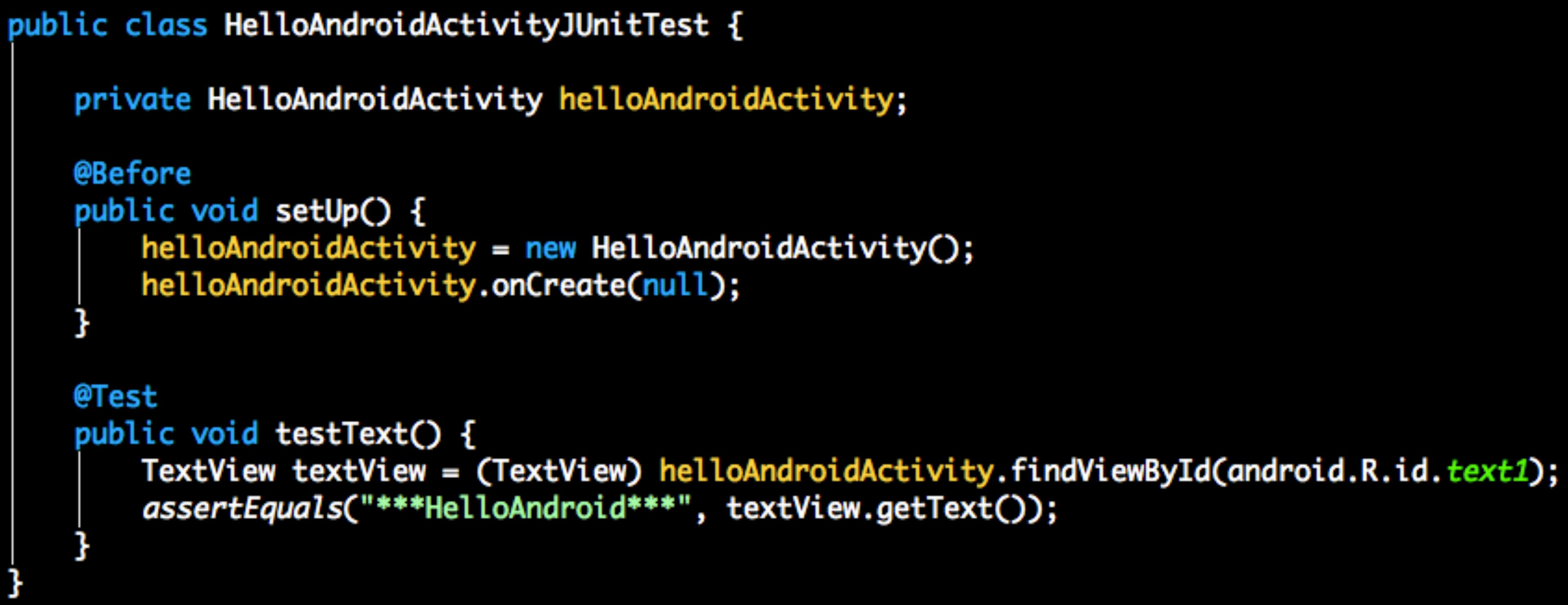
Additional Challenges

- Classes and methods declared *final*
- Lack of interfaces
- Non-public constructors
- Static methods



JUnit 4 + POJOs

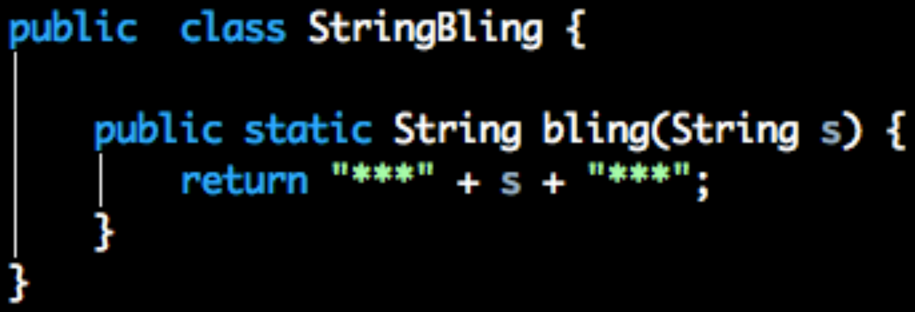
HelloAndroidActivityJUnitTest



```
public class HelloAndroidActivityJUnitTest {

    private HelloAndroidActivity helloAndroidActivity;

    @Before
    public void setUp() {
        helloAndroidActivity = new HelloAndroidActivity();
        helloAndroidActivity.onCreate(null);
    }

    @Test
    public void testText() {
        TextView textView = (TextView) helloAndroidActivity.findViewById(android.R.id.text1);
        assertEquals("***HelloAndroid***", textView.getText());
    }
}
```

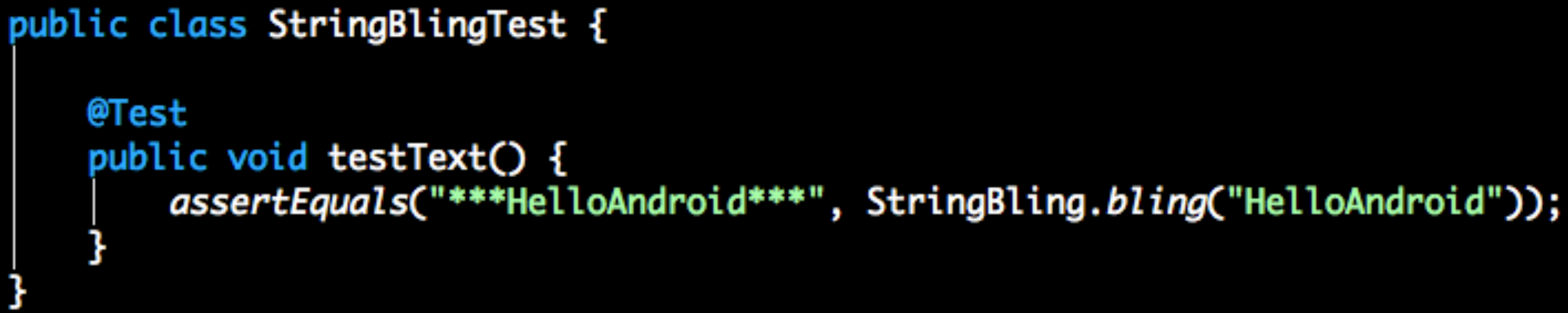
```
java.lang.RuntimeException: Stub!
    at android.content.Context.<init>(Context.java:4)
    at android.content.ContextWrapper.<init>(ContextWrapper.java:5)
    at android.view.ContextThemeWrapper.<init>(ContextThemeWrapper.java:5)
    at android.app.Activity.<init>(Activity.java:6)
    at com.example.HelloAndroidActivity.<init>(HelloAndroidActivity.java:8)
    at com.example.HelloAndroidActivityJUnitTest.setUp(HelloAndroidActivityJUnitTest.java:15)
```

StringBling



```java
public  class StringBling {

    public static String bling(String s) {
        return "***" + s + "***";
    }
}
```

StringBlingTest



```
public class StringBlingTest {

    @Test
    public void testText() {
        assertEquals("***HelloAndroid***", StringBling.bling("HelloAndroid"));
    }
}
```

Success!!!

But we want to test
ALL our code

ROBOLECTRIC

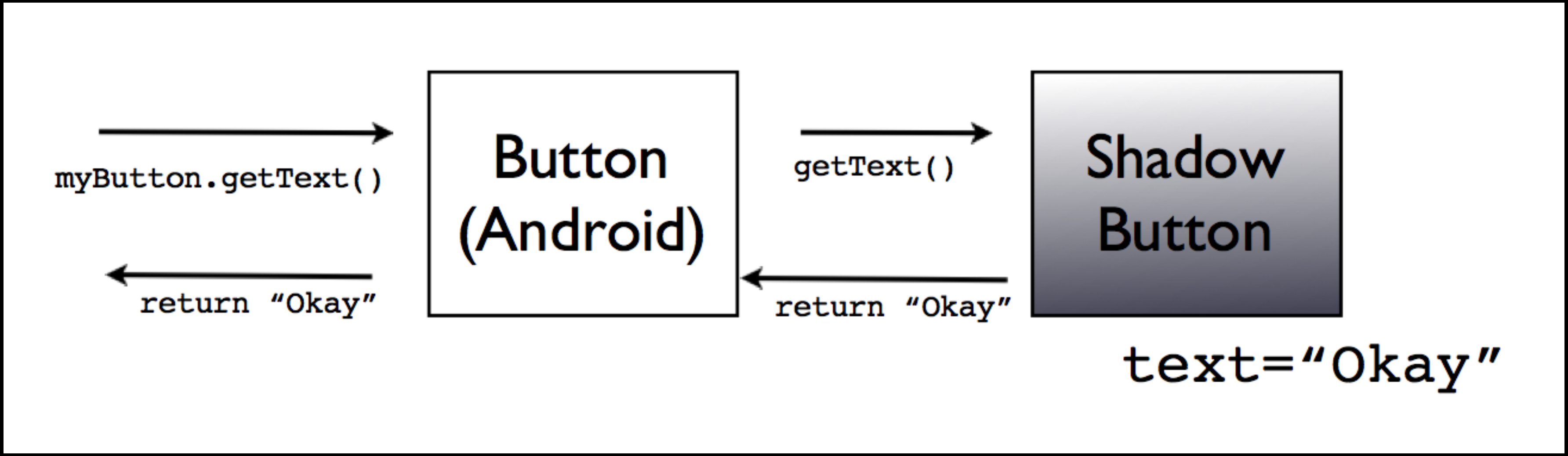
Robolectric

Robolectric

Test-Drive Your Android Code

Shadow Objects

Shadow Objects in Action



View and Resource Loading

HelloAndroidActivityRobolectricTest

```
@RunWith(RobolectricTestRunner.class)
public class HelloAndroidActivityRobolectricTest {

    private HelloAndroidActivity helloAndroidActivity;

    @Before
    public void setUp() {
        helloAndroidActivity = new HelloAndroidActivity();
        helloAndroidActivity.onCreate(null);
    }

    @Test
    public void testText() {
        TextView textView = (TextView) helloAndroidActivity.findViewById(android.R.id.text1);
        assertEquals("***HelloAndroid***", textView.getText());
    }
}
```

Runs: 4/4
Errors: 0
Failures: 0
Success!!!

More fun with Shadows

Using ShadowImageView

```
@Test
public void testImage() {
    ImageView imageView = (ImageView) helloAndroidActivity.findViewById(R.id.image);
    ShadowImageView shadowImageView = Robolectric.shadowOf(imageView);
    assertEquals(R.drawable.hello, shadowImageView.getResourceId());
}
```

Using ShadowActivity

```
@Test
public void testMarketLaunch() {
    ShadowActivity shadowActivity = Robolectric.shadowOf(helloAndroidActivity);
    Intent startedIntent = shadowActivity.getNextStartedActivity();

    Uri marketUri = Uri.parse("market://details?id=com.example");
    assertEquals(marketUri, startedIntent.getData());
}
```

Write your own custom shadows

Contribute to Robolectric

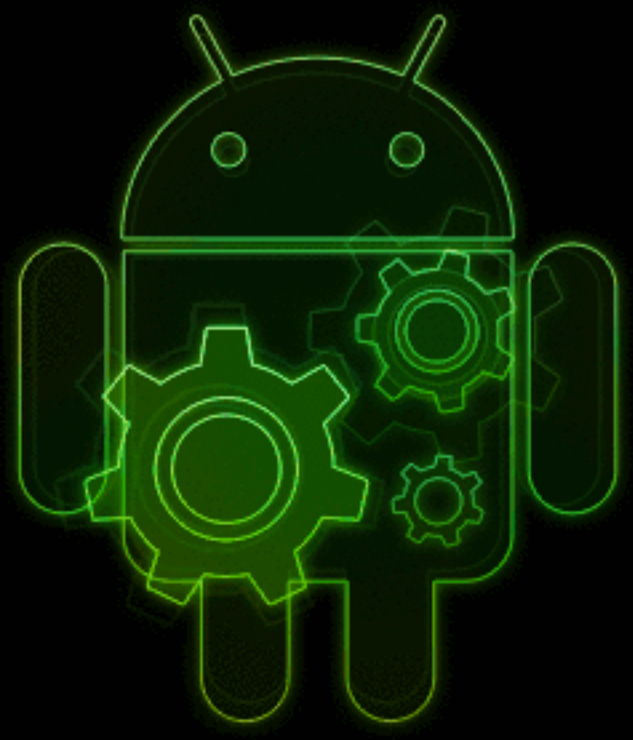
Resources

- developer.android.com/guide/topics/testing
- junit.org
- pivotal.github.com/robolectric

Reading

- Robert C. Martin (Uncle Bob)
- Michael Feathers
- Kent Beck

Find your testing Zen



Chuck Greb (@ecgreb)

http://ecgreb.com/android-testing

https://github.com/ecgreb/StringBling