

# Android rendszeren futó alkalmazások tesztelése

Kundra László

# Android tesztelés alapok

- Java alapú fejlesztés → JUnit
- @Annotációs segédosztályok a teszteléshez
- Android build tools támogatással
  - Teszt projekt generálása
  - Build fájlok generálása
- Android testing API: JUnit API kibővítve Android specifikus tesztekkel.

```
    */  
    @Test  
    void testFoo();  
  
    @Test  
    void testFoo2();  
}
```

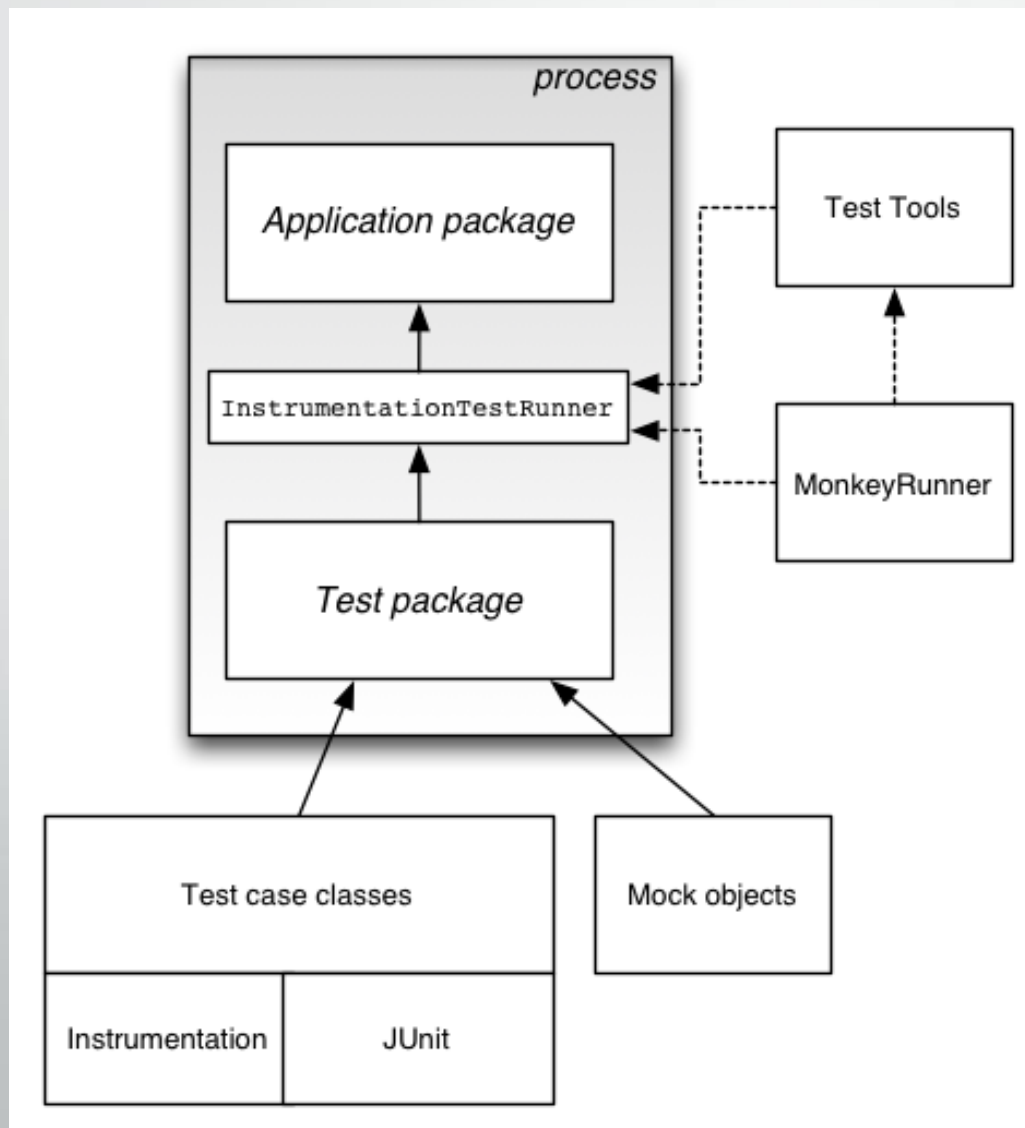
# Android Testing API

- JUnit TestCase osztály
  - A tesztelt osztály nem használja az Android API-t
- AndroidTestCase
  - A TestCase-ből származik
- JUnit Assert
  - Eredmények összevetése és eltérés esetén AssertionError
- Testing API: MoreAsserts
  - Bővített tudás, pl. reguláris kifejezés tesztelése
- Testing API: ViewAsserts
  - A felületi elemek tesztelésében segít, pl. adott View-nak van-e adott koordinátájú képpontja.

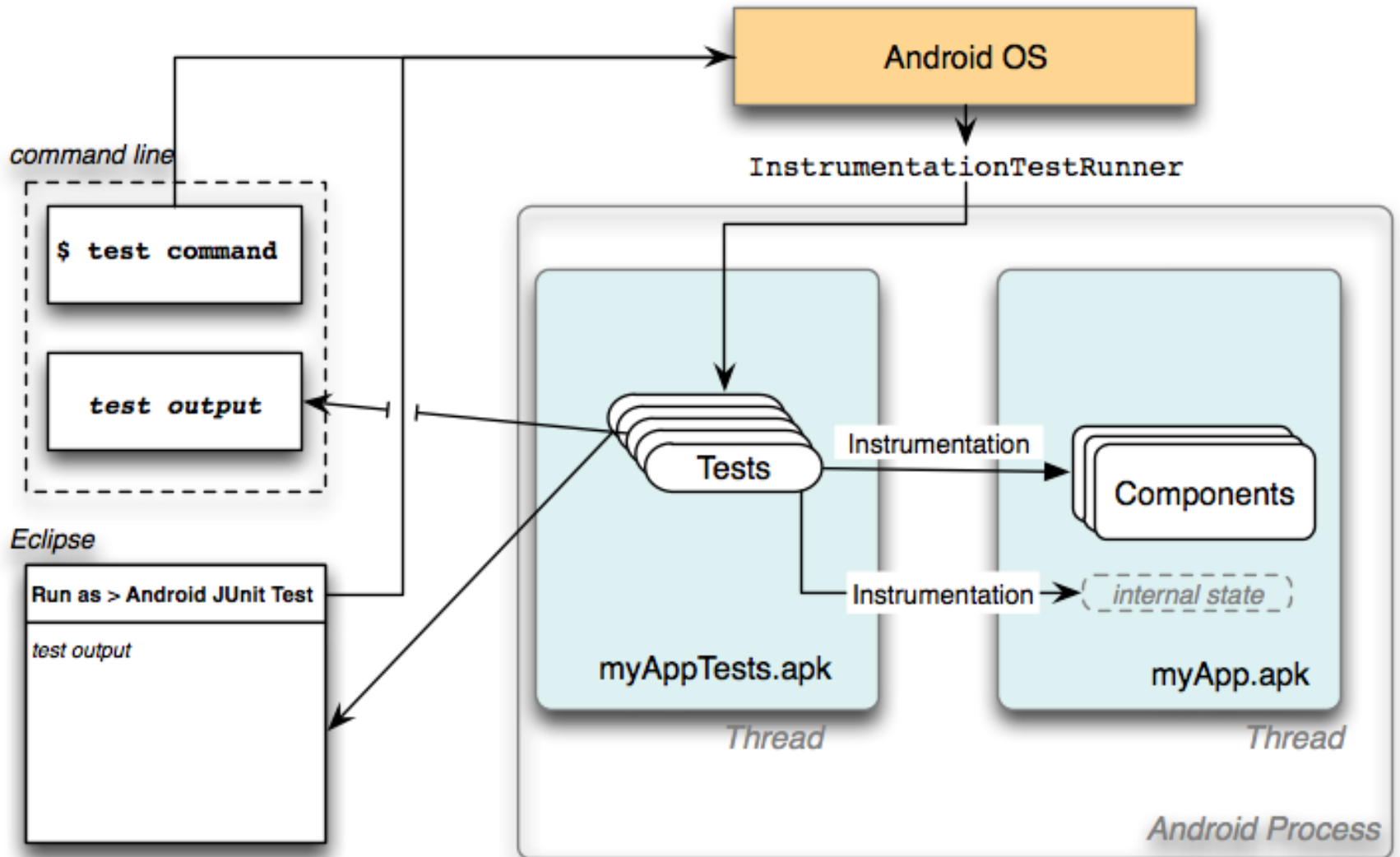
# Instrumentation és Mock objektumok

- Android instrumentation
  - Android framework osztályok „hook”-okkal
  - Életciklus befolyásolása
  - A teszt projekt és az alkalmazás egy folyamatként töltődik be
- Külső függőségek feloldása tesztelés alatt: Mock objektumokkal
  - Saját vagy OS szintű szolgáltatások (Service), kontextus (Context), stb. hiányának feloldása
  - A rendszer biztonságban marad a fejlesztés / tesztelés alatt

# A teszt felépítése I.



# A teszt felépítése II.



# Funkcionális tesztelés

- Monkey
  - Parancssoros eszköz (ADB: Android Debug Bridge)
  - Főleg UI és UX teszteléshez
  - Gombnyomás, érintés, és gesztusok injektálása futás közben
  - Stress-test
- Monkeyrunner
  - Parancssoros eszköz (ADB-n)
  - Python alapú futtatókörnyezet tesztelés
  - Csomagok telepítése, képernyőképek, kapcsolatok állítása, stb.
  - Teszt csomagok futtatása alkalmazásokon
  - Komplex tesztek felépíthetők segítségével

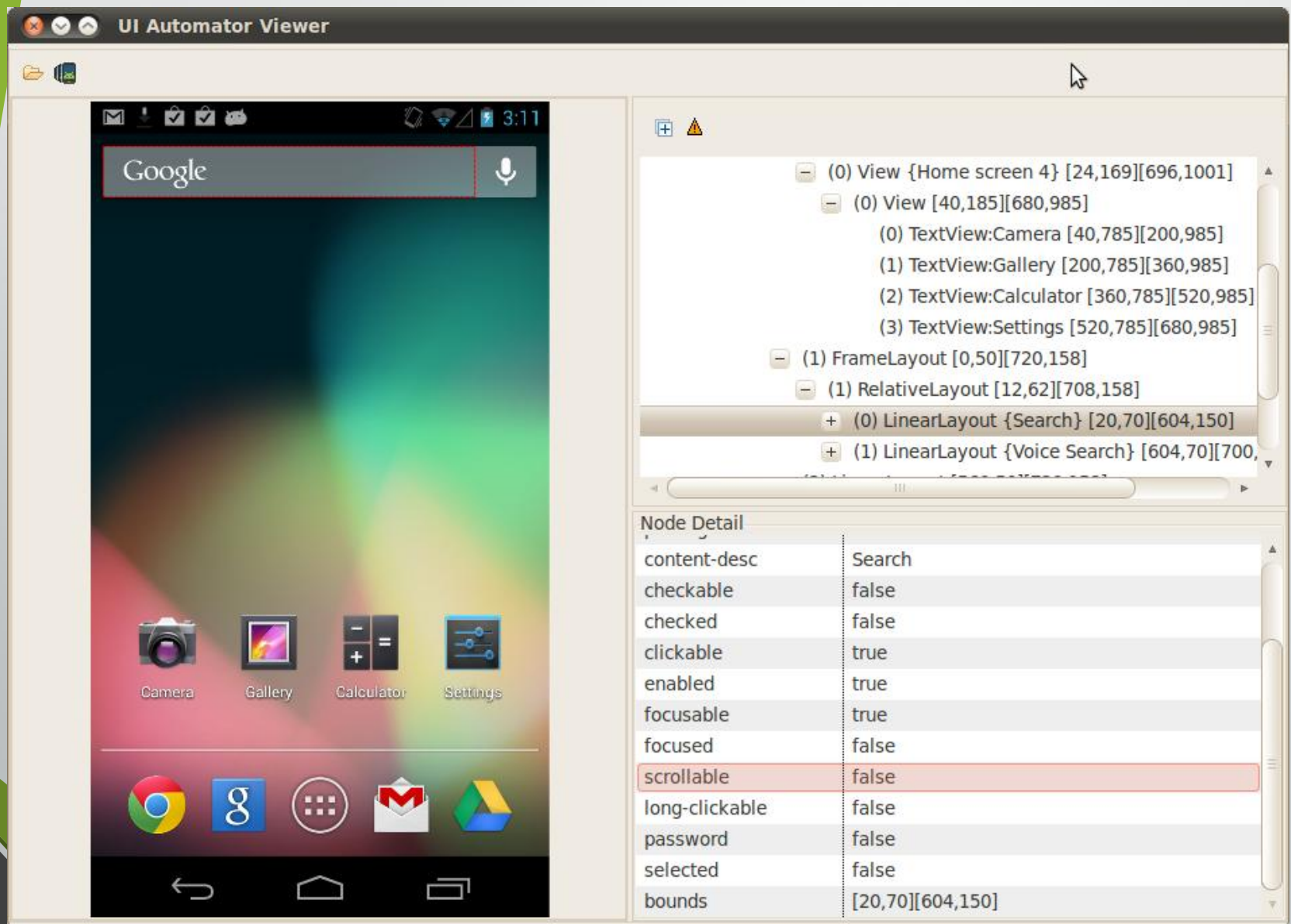
# UI tesztelése

- Automatizált blackbox UI tesztelés
  - Nem lényeg az alkalmazás belső működése
  - Az emberi munka drága, nem idő-hatékony és hibázhat
- Android tools: uiautomatorviewer
  - Az eszköz képernyőjének hierarchikus bontása
  - A nézetek (View-k) paramétereinek vizsgálata
- Android tools: uiautomator
  - ADB-n keresztül meghívható
  - Funkcionális tesztelés
  - Java könyvtár UI tesztek létrehozásához és futtatásához  
→ Teszt projekt Java kóddal

```
$ adb shell uiautomator runtest LaunchSettings.jar  
-c com.uia.example.my.LaunchSettings
```



# A uiautomatorviewer valós eszközzel



The image shows the UI Automator Viewer application interface. On the left, a simulated mobile home screen is displayed with a Google search bar at the top, a grid of app icons (Camera, Gallery, Calculator, Settings) in the middle, and a dock with icons for Chrome, Google, App Drawer, Email, and Drive at the bottom. On the right, the UI hierarchy is shown as a tree view. The selected node is a LinearLayout {Search} with the following properties:

Node Detail	
content-desc	Search
checkable	false
checked	false
clickable	true
enabled	true
focusable	true
focused	false
scrollable	false
long-clickable	false
password	false
selected	false
bounds	[20,70][604,150]

# A uiautomator API

- UI komponensek elérése és manipulálása
  - UiDevice
    - Az eszköz képernyőjének orientációja, felbontása; hardver gombok, stb.
  - UiSelector
    - A nézetek hierarchiájából lekéréssel visszaad egyet
  - UiObject
    - Grafikus elemet reprezentáló objektum, dinamikus: hozzáféréskor értékelődik ki!
  - UiCollection
    - Elemek gyűjteménye, viselkedésére hasonló a UiObject-hez. Pl.: email lista
  - UiScrollable
    - Görgethető komponenst reprezentál, melyet így tesztyszerűen görgethető

# Auiautomator API

- UI komponensek elérése és manipulálása

- UiDevice

- Az eszköz képernyőjének orientációja, f
- ```
getUiDevice().pressHome();
```

- UiSelector

- A nézetek hier

```
UiObject appItem = new UiObject(new UiSelector()  
.className("android.widget.ListView").instance(1)  
.childSelector(new UiSelector().text("Apps")));
```

- UiObject

- Grafil
- ```
UiObject okButton = new UiObject(new UiSelector().text("OK"));
```
- értékelődik ki!

- UiCollection

- Elemek gyűjtése  
email lista

```
UiCollection videos = new UiCollection(new UiSelector()  
.className("android.widget.FrameLayout"));  
int count = videos.getChildCount(new UiSelector()  
.className("android.widget.LinearLayout"));
```

- UiScrollable

- Görgethető komponensre reprezentatív, melyet így tesztelhetünk  
görgethető

```
UiScrollable settingsItem = new UiScrollable(new UiSelector()  
.className("android.widget.ListView"));
```

# Egyszerű példa

```
public class LaunchSettings extends UiAutomatorTestCase {

    public void testDemo() throws UiObjectNotFoundException {

        getUiDevice().pressHome();
        UiObject allAppsButton = new UiObject(new UiSelector()
            .description("Apps"));

        allAppsButton.clickAndWaitForNewWindow();

        UiObject appsTab = new UiObject(new UiSelector()
            .text("Apps"));

        appsTab.click();

        UiScrollable appViews = new UiScrollable(new UiSelector()
            .scrollable(true));

        appViews.setAsHorizontalList();

        UiObject settingsApp = appViews.getChildByText(new UiSelector()
            .className(android.widget.TextView.class.getName()),
            "Settings");
        settingsApp.clickAndWaitForNewWindow();

        UiObject settingsValidation = new UiObject(new UiSelector()
            .packageName("com.android.settings"));
        assertTrue("Unable to detect Settings",
            settingsValidation.exists());
    }
}
```

# Mit mást érdemes tesztelni?

- Képernyő felbontás / orientáció változás
  - Táblagépek támogatása, elforgatás kezelése minden felületen
- Konfiguráció változása
  - Hálózati kapcsolat, billentyűzet megléte, rendszer nyelve, stb.
- Akkumulátor élettartam
  - Bővebben: Google IO konferencia: „[Coding for Life -- Battery Life, That Is](#)”
- Egyéb külső függőségek
  - GPS, Bluetooth megléte; SMS fogadás, mobilinternet, stb.

# Tesztelést támogató eszközök (ingyenes, többnyire nyílt forrás)

- Hamcrest (<https://code.google.com/p/hamcrest/>)
- Android FEST (<https://github.com/square/fest-android>)
  - Android UI tesztelés és hibafelderítés
- Awaitility (<https://code.google.com/p/awaitility/>)
  - Aszinkron tesztelés
- Spoon (<http://square.github.io/spoon/>)
  - Elosztott tesztelés sok eszközön, vizuális jelentések, képernyőképek, stb.



Köszönöm a figyelmet!