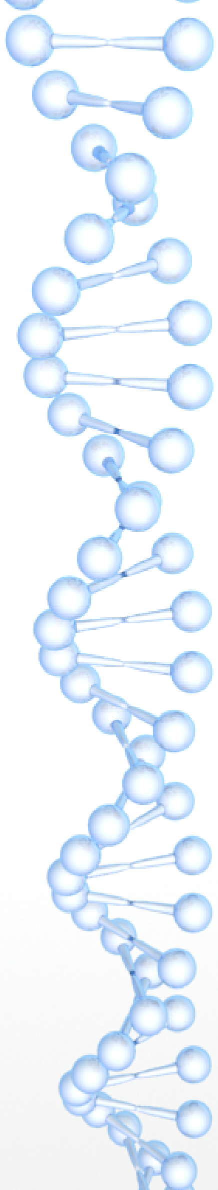
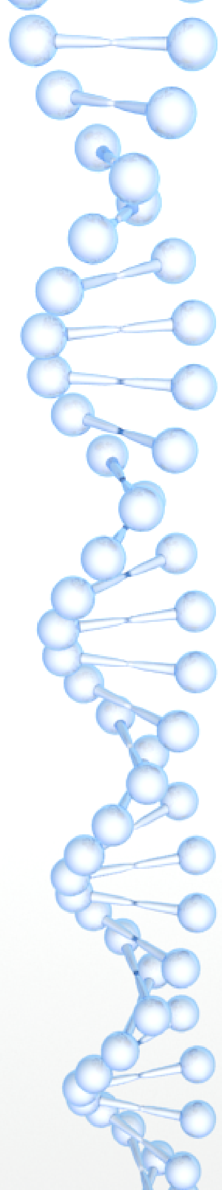
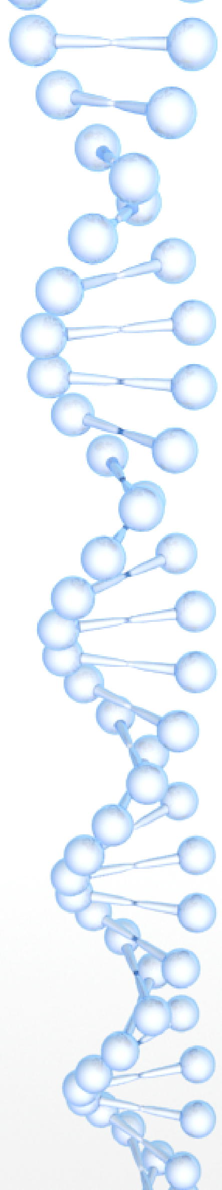


Android Testing

What to test?

- Strictly speaking, you should test every statement in your code,
- But this also depends on different criteria and can be reduced to testing the main path of execution or just some key methods.
- **Activity lifecycle events:** test whether your activities handle lifecycle events correctly



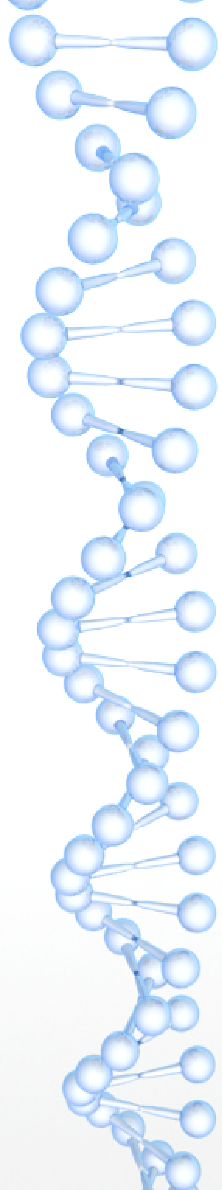


Cont ...

- **Database and filesystem operations:**should be tested to ensure that the operations and any errors are handled correctly.
- These operations should be tested in isolation at the lower system level, at a higher level through ContentProviders, or from the application itself.
- To test these components in isolation, Android provides some mock objects in theandroid.test.mock package

Cont ...

- **Physical characteristics of the device:** Before shipping your application, you should be sure that all of the different devices it can be run on are supported, or at least you should detect the unsupported situation and take pertinent measures.
- The characteristics of the devices that you should test are:
  - Network capabilities
  - Screen densities

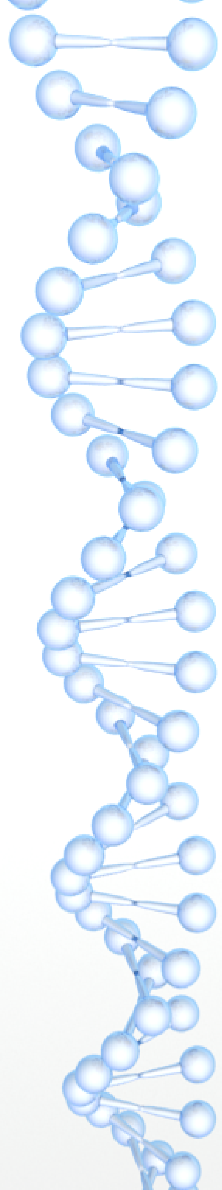


Cont ...

- Screen resolutions
- Screen sizes
- Availability of sensors
- Keyboard and other input devices GPS
- External storage

Mock objects

- Mock objects are mimic objects used instead of calling the real domain objects to enable testing units in isolation.
- Several classes are provided by the Android testing framework in the android.test.mock package:
  - MockApplication
  - MockContentProvider

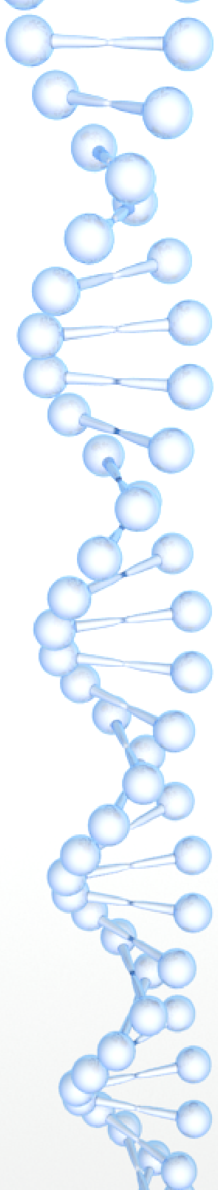


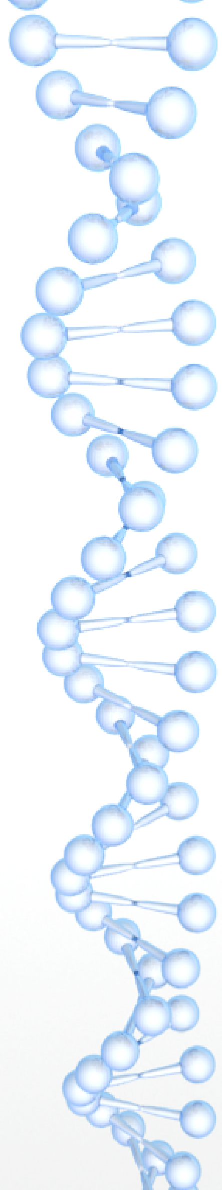
Cont ...

- MockContentResolver
- MockContext
- MockCursor
- MockDialogInterface
- MockPackageManager
- MockResources

Android testing framework

- Android provides a very advanced testing framework that extends the industry standard JUnit library with specific features that are suitable to implement all of the testing strategies and types.
- Most relevant key features of the Android testing environment include:
  - Android extensions to the Junit framework that provide access to Android system objects
  - An instrumentation framework that lets the tests control and examine the application
  -





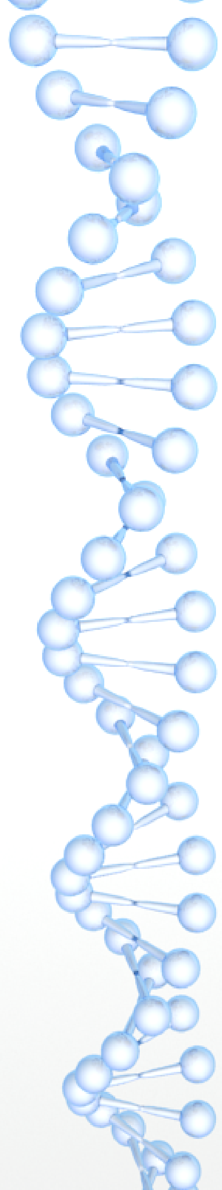
Cont ...

- Mock versions of commonly used Android system objects
- Tools to run single tests or test suites, with or without instrumentation
- Support to manage tests and test projects in Android Studio and at the command line

Instrumentation

- The instrumentation framework is the foundation of the testingframework.
- Instrumentation controls the application under tests and permits the injection of mock components required by the application torun.

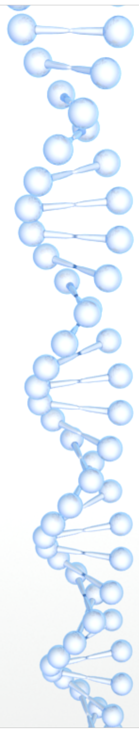
Test annotations



Android Testing

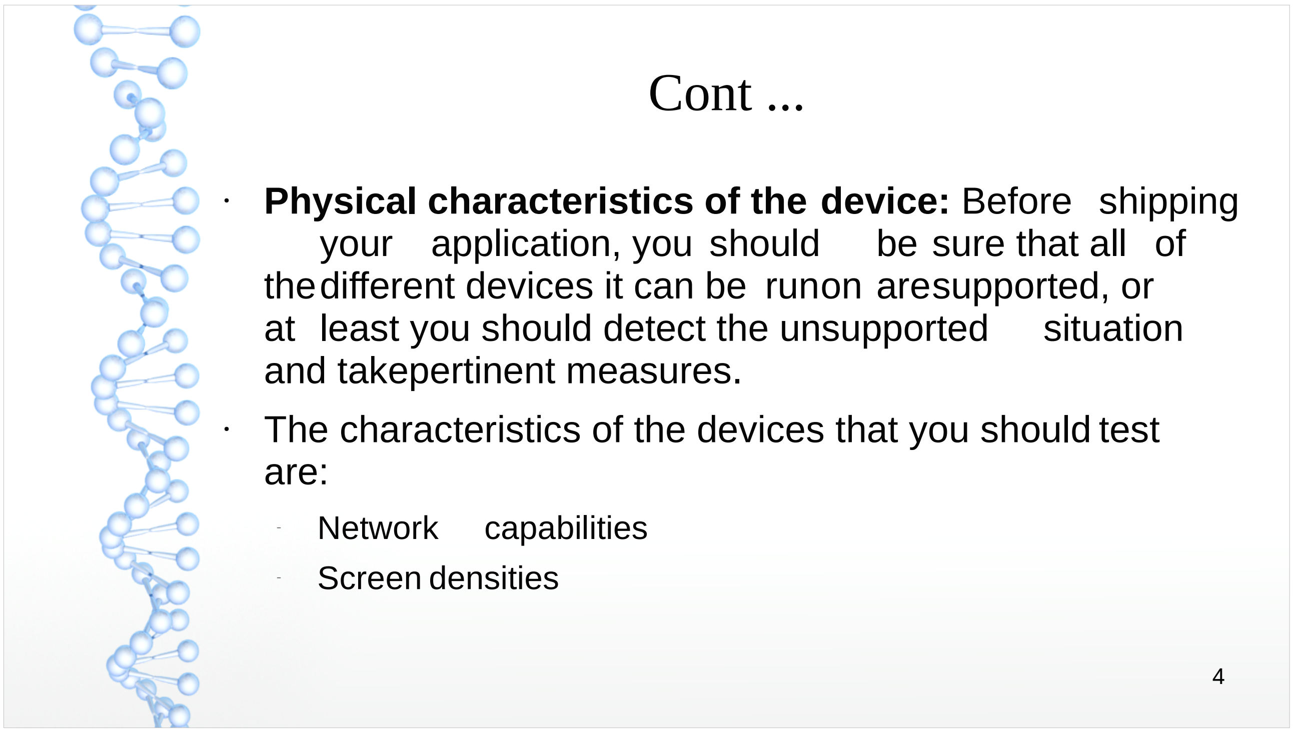
What to test?

- Strictly speaking, you should test every statement in your code,
- But this also depends on different criteria and can be reduced to testing the main path of execution or just some key methods.
- **Activity lifecycle events:** test whether your activities handle lifecycle events correctly



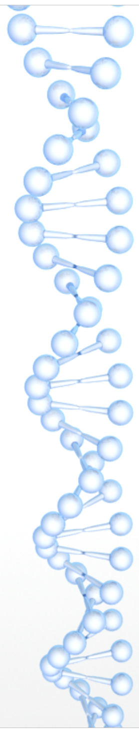
Cont ...

- **Database and filesystem operations:**should be tested to ensure that the operations and any errors are handled correctly.
- These operations should be tested in isolation at the lower system level, at a higher level through ContentProviders, or from the application itself.
- To test these components in isolation, Android provides some mock objects in theandroid.test.mock package



Cont ...

- **Physical characteristics of the device:** Before shipping your application, you should be sure that all of the different devices it can be runon aresupported, or at least you should detect the unsupported situation and takepertinent measures.
- The characteristics of the devices that you should test are:
  - Network capabilities
  - Screen densities



Cont ...

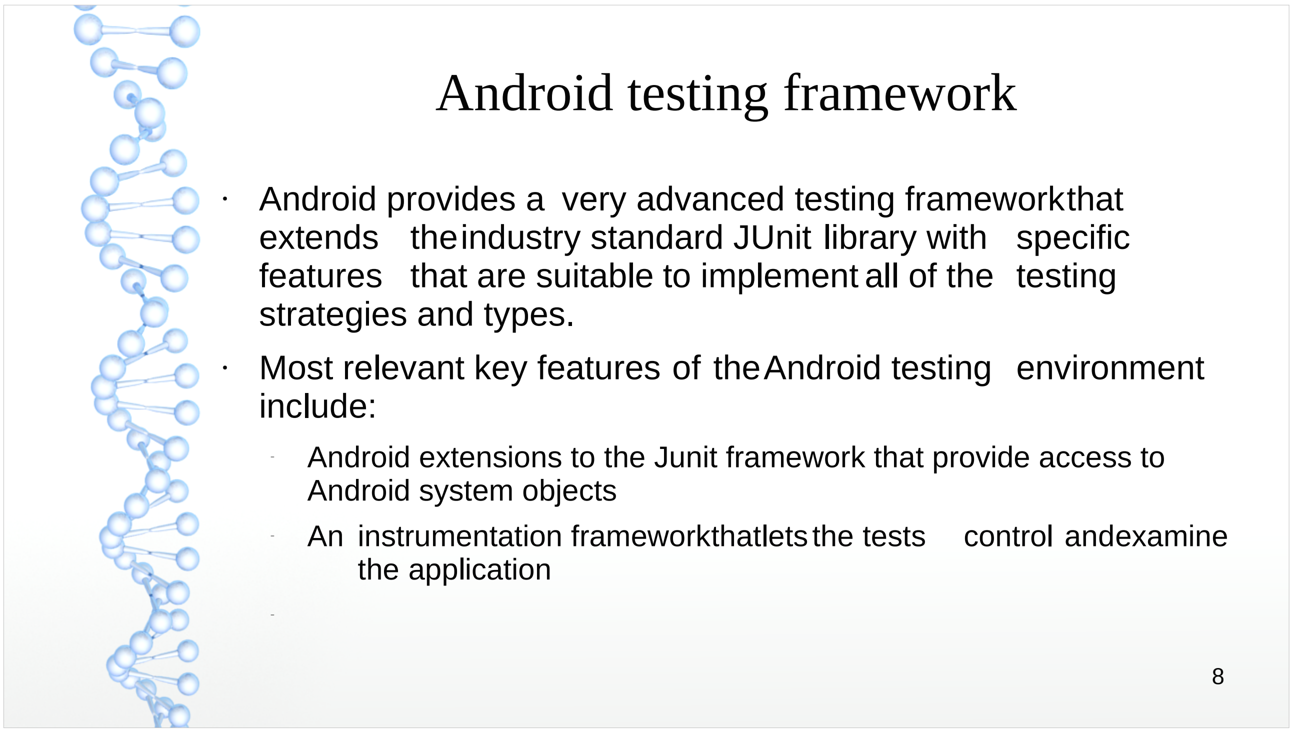
- Screen resolutions
- Screen sizes
- Availability of sensors
- Keyboard and other input devices GPS
- External storage

Mock objects

- Mock objects are mimic objects used instead of calling the real domain objects to enable testing units in isolation.
- Several classes are provided by the Android testing framework in the android.test.mock package:
  - MockApplication
  - MockContentProvider

Cont ...

- MockContentResolver
- MockContext
- MockCursor
- MockDialogInterface
- MockPackageManager
- MockResources



Android testing framework

- Android provides a very advanced testing framework that extends the industry standard JUnit library with specific features that are suitable to implement all of the testing strategies and types.
- Most relevant key features of the Android testing environment include:
  - Android extensions to the Junit framework that provide access to Android system objects
  - An instrumentation framework that lets the tests control and examine the application
  -

Cont ...

- Mock versions of commonly used Android system objects
- Tools to run single tests or test suites, with or without instrumentation
- Support to manage tests and test projects in Android Studio and at the command line

Instrumentation

- The instrumentation framework is the foundation of the testingframework.
- Instrumentation controls the application under tests and permits the injection of mock components required by theapplication torun.

Test annotations