Software Engineering Large Practical

Debugging and Testing Android apps

Stephen Gilmore

School of Informatics, University of Edinburgh

November 13th, 2013

News

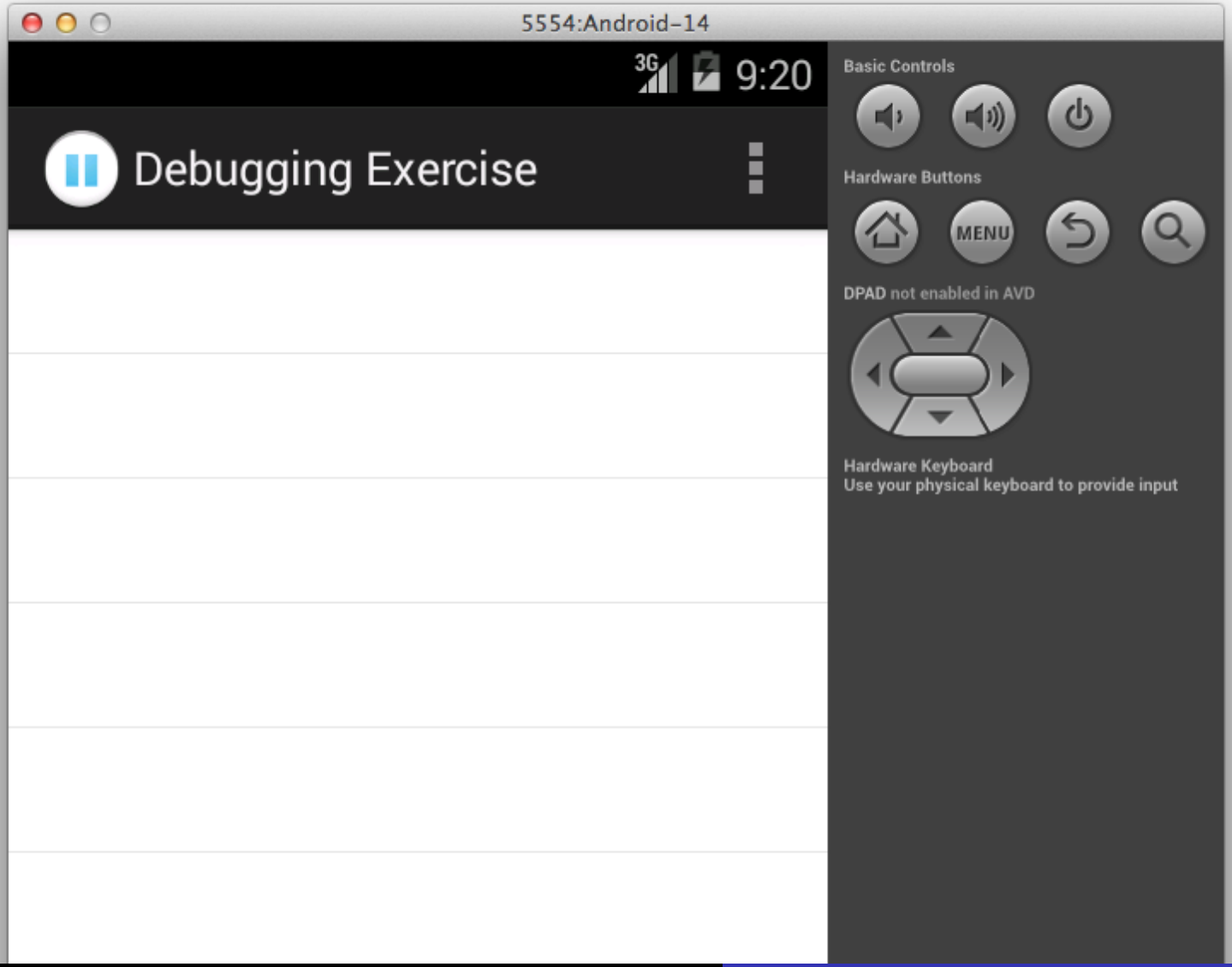
- This is the last SELP lecture.

Debugging Android Apps

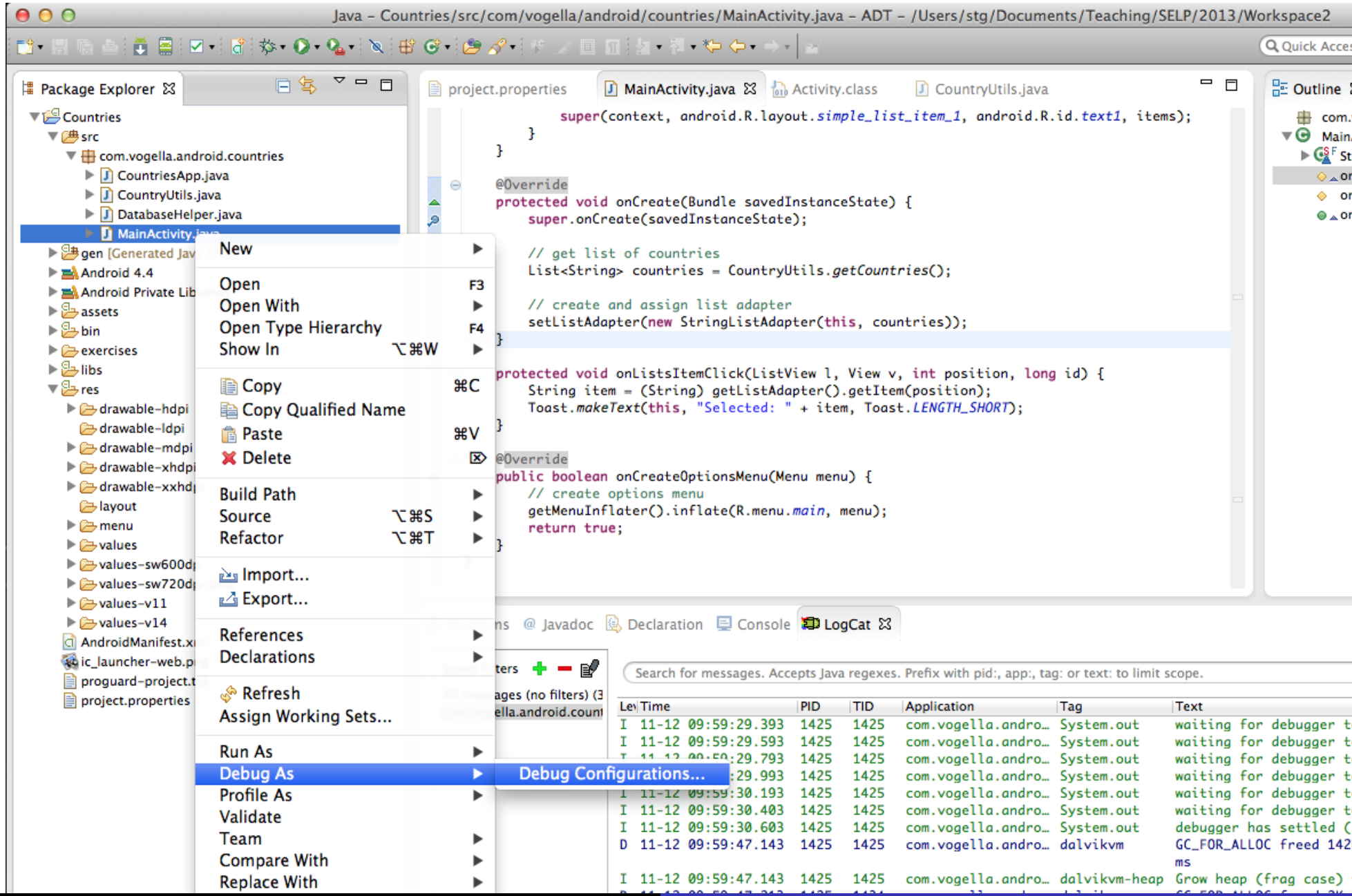
In this lecture we consider debugging and testing our Android apps, using the Android debugger and the Android Robotium unit testing suite.

We follow an example due to Lars Vogel, available at http://www.vogella.com/articles/AndroidDebugging/article.html

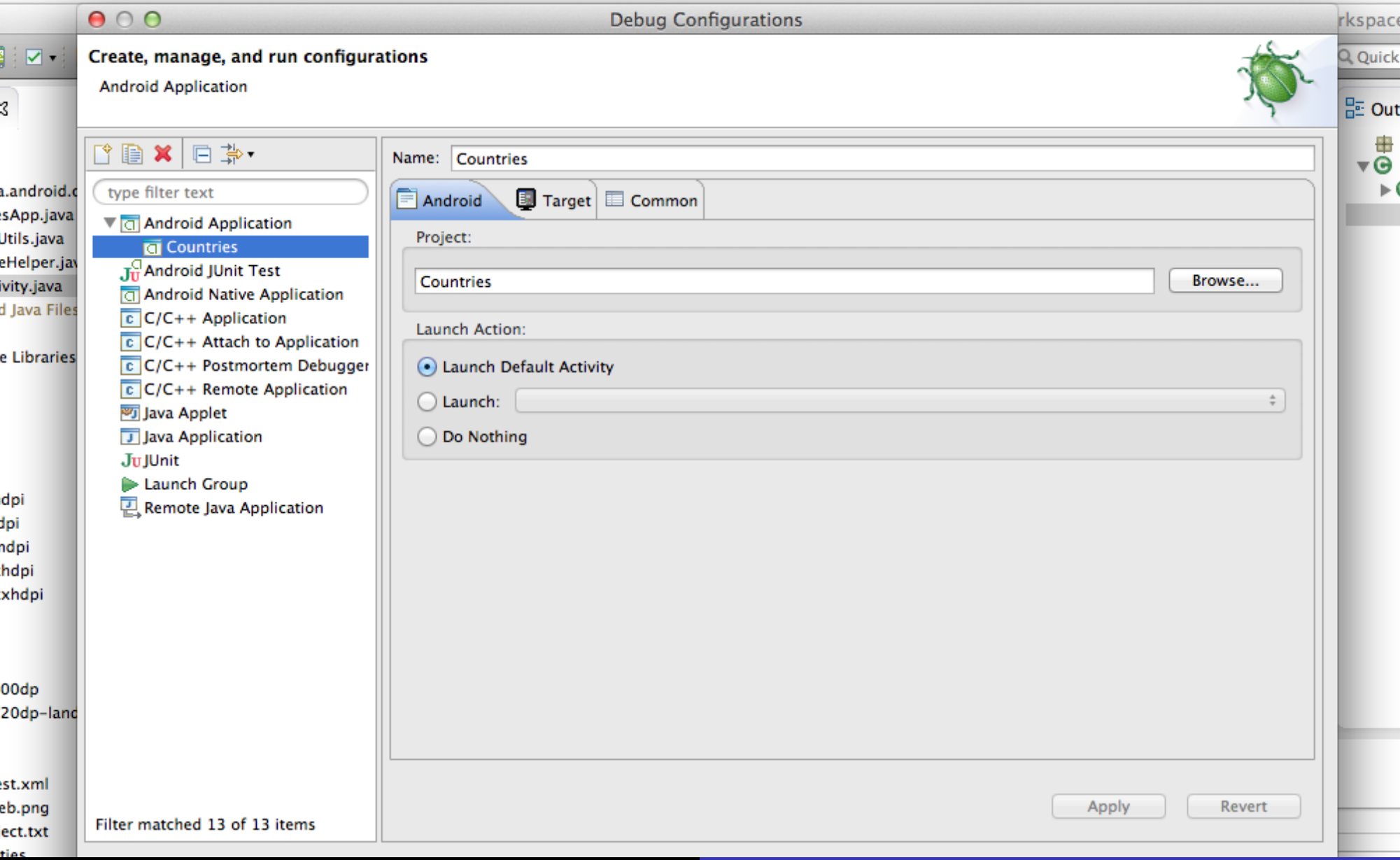
The problem: our app should display a list of countries, but it doesn't



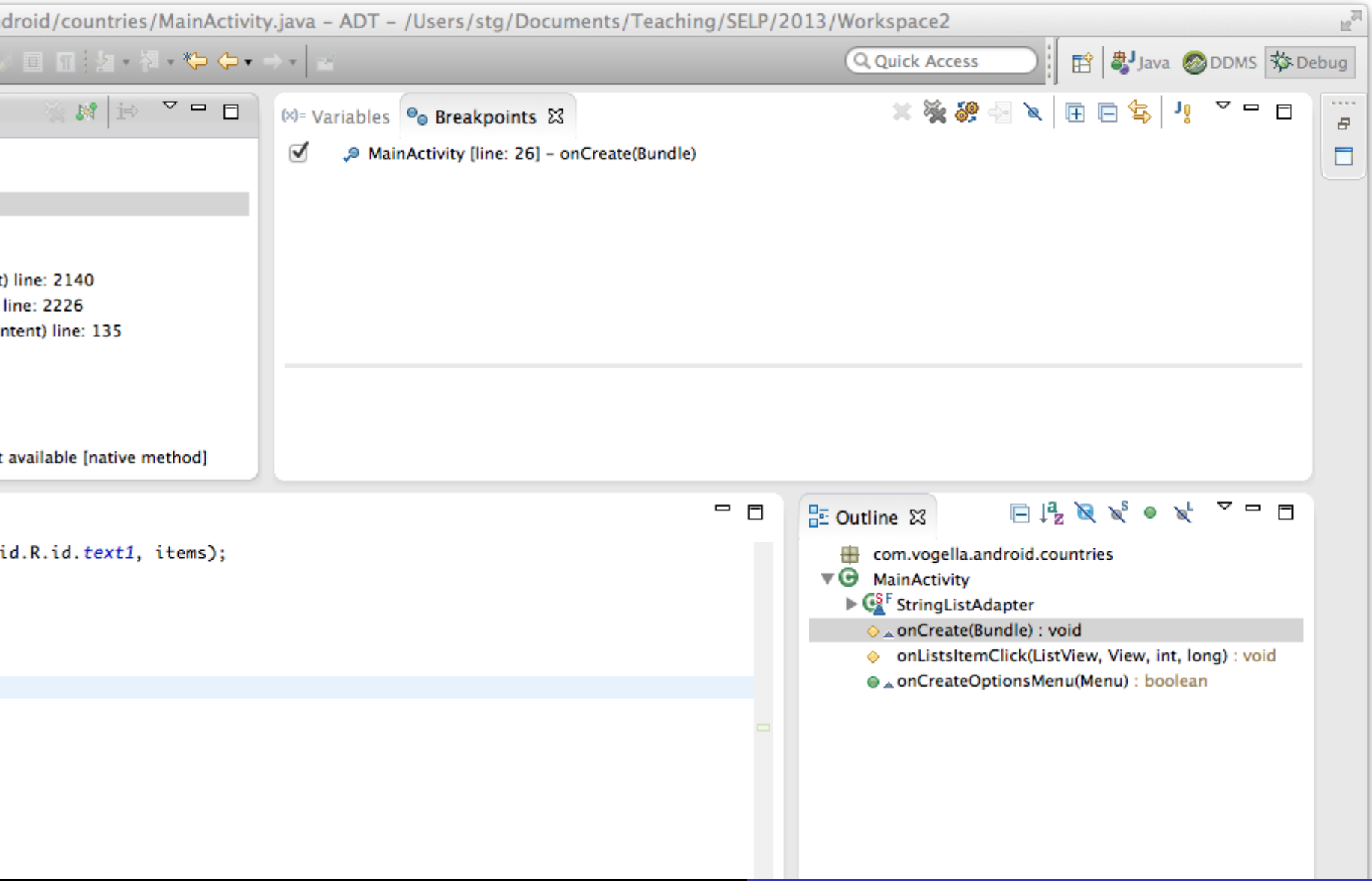
Getting started: Choose "Debug As"



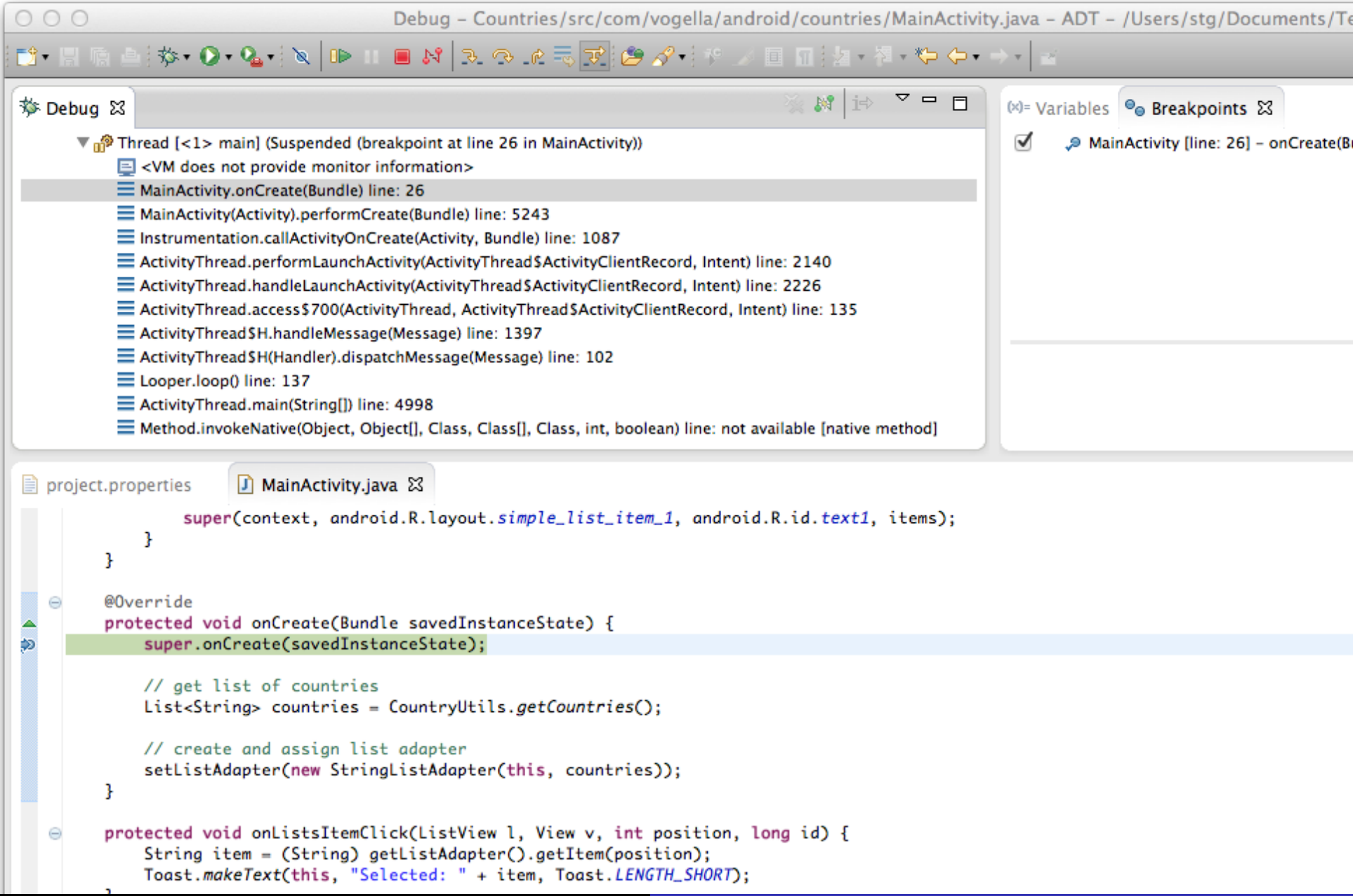
Debug configurations



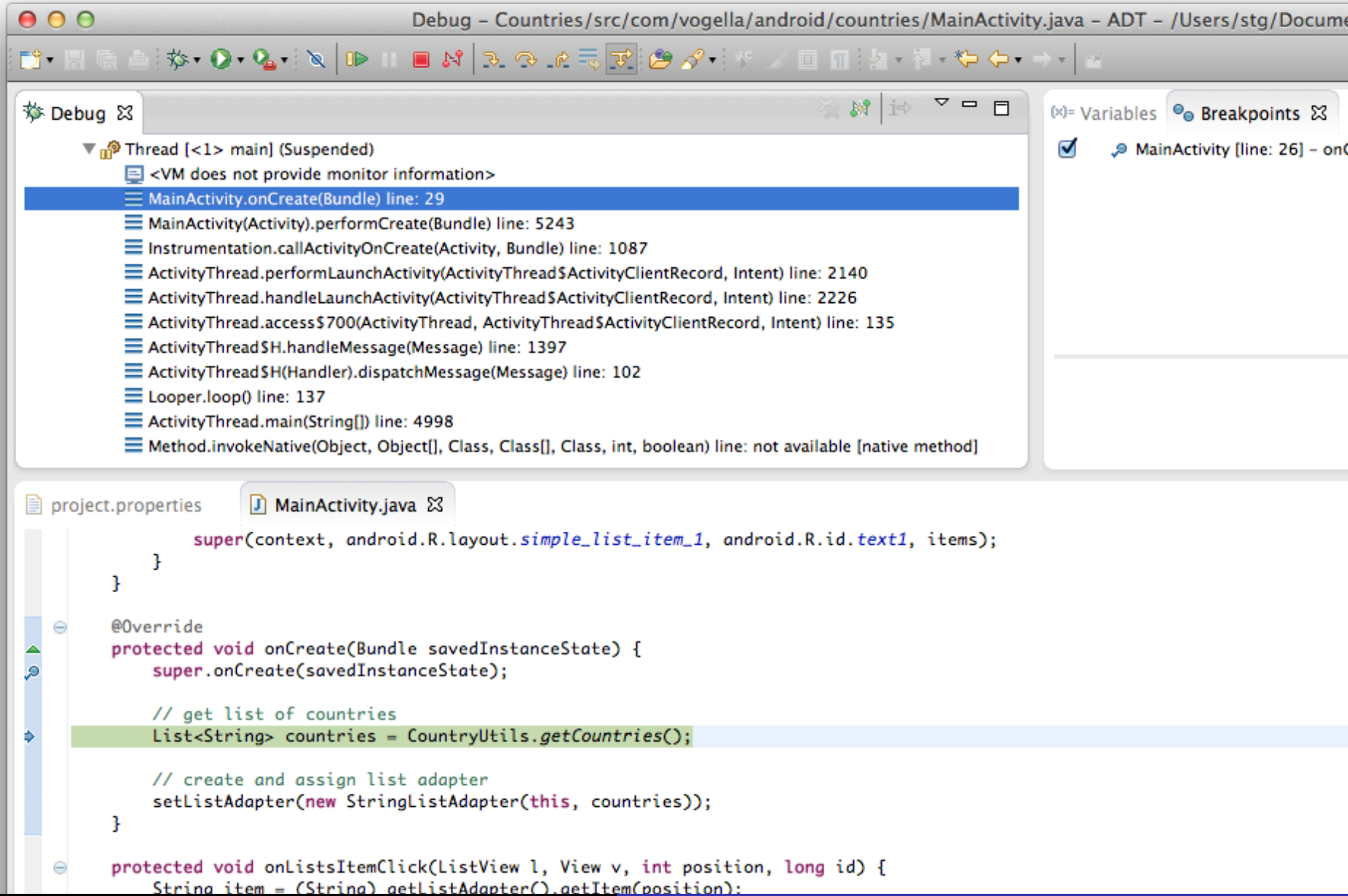
Switch to the "Debug" perspective to see new views such as "Variables" and "Breakpoints"



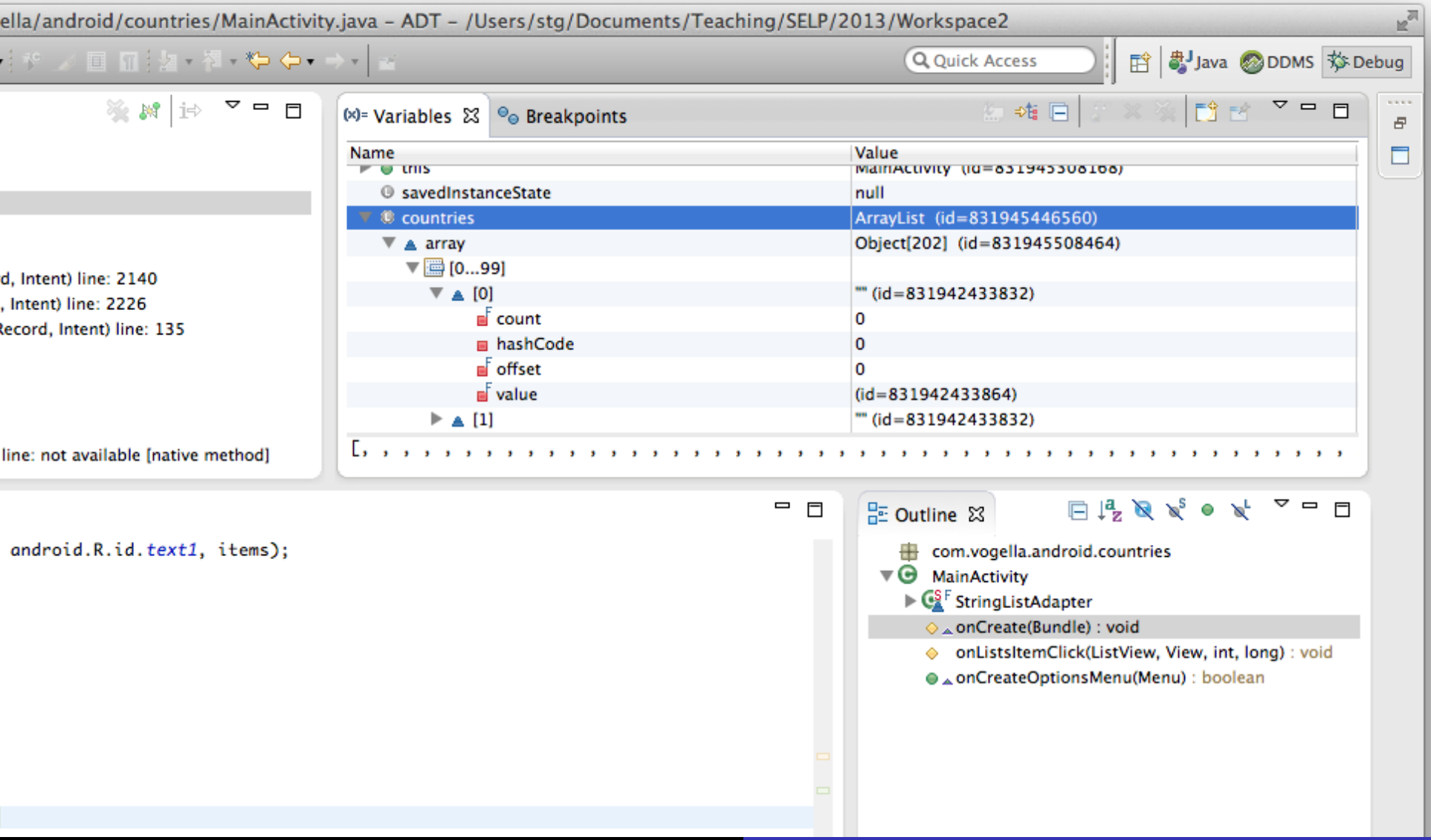
Stopping at a breakpoint in the *onCreate* method



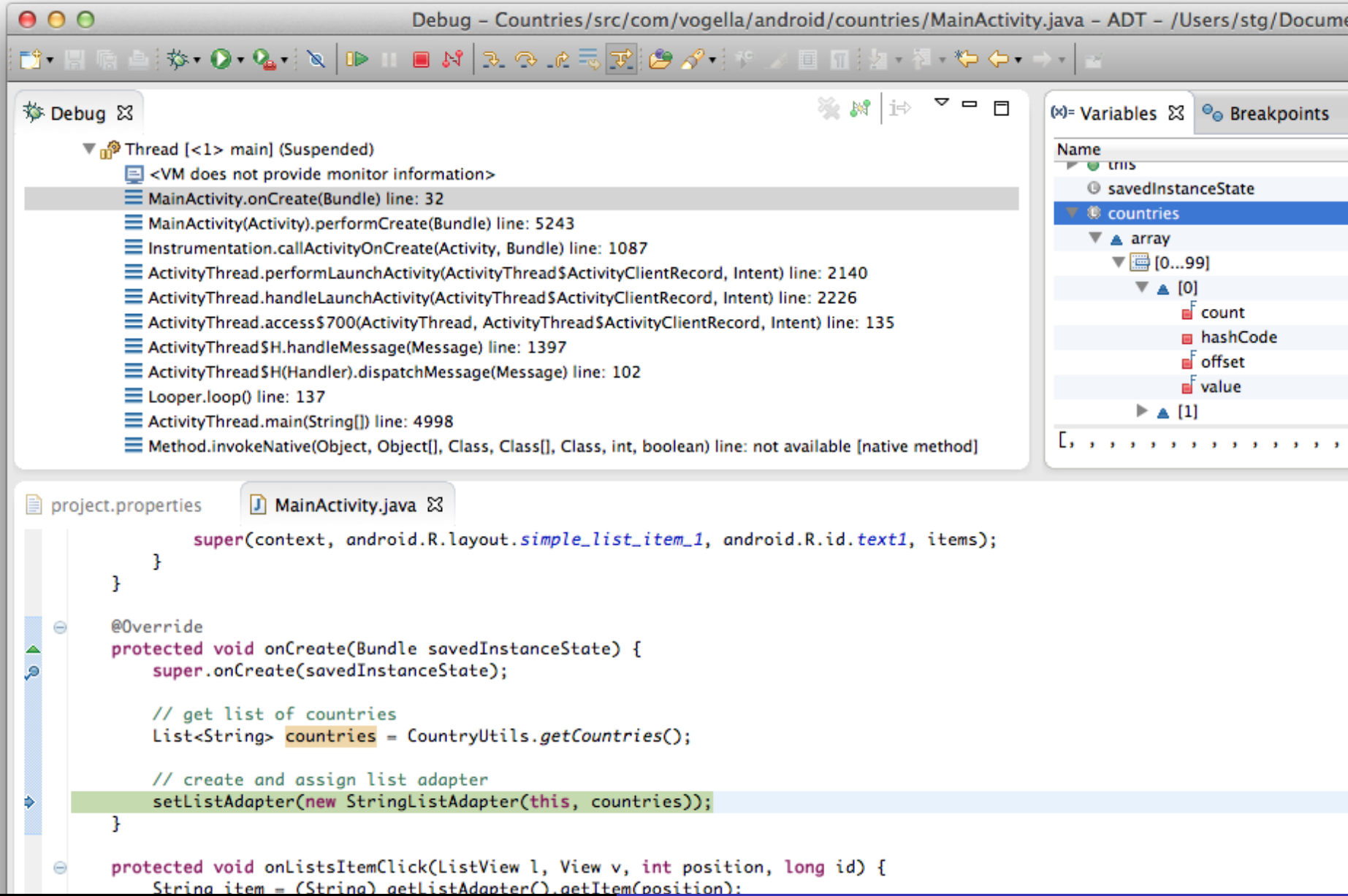
Assigning a value to the list of countries



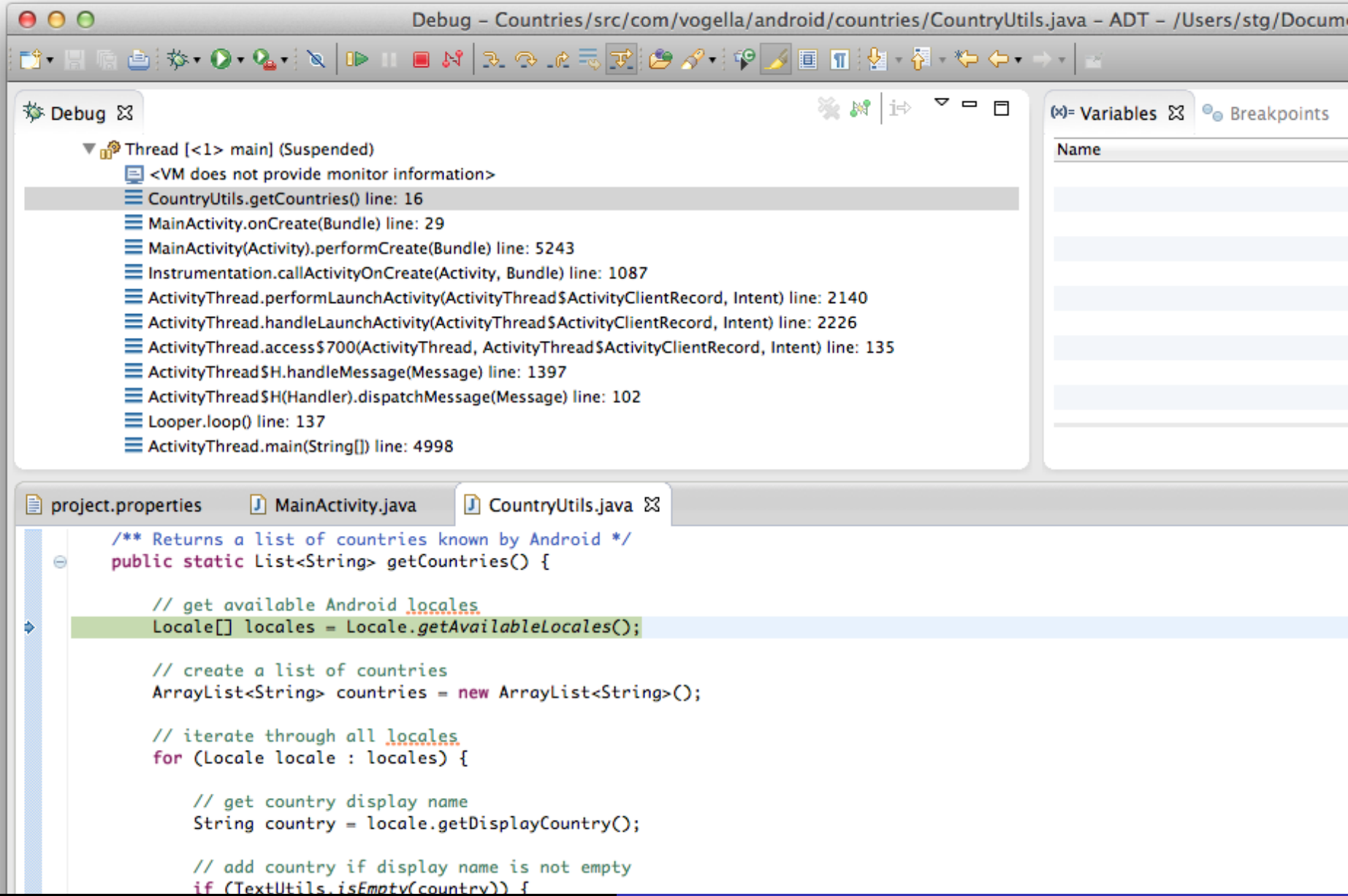
Inspecting the value of the *countries* variable: an array of empty strings [””, ””, ””, ””, ””, …, ””]



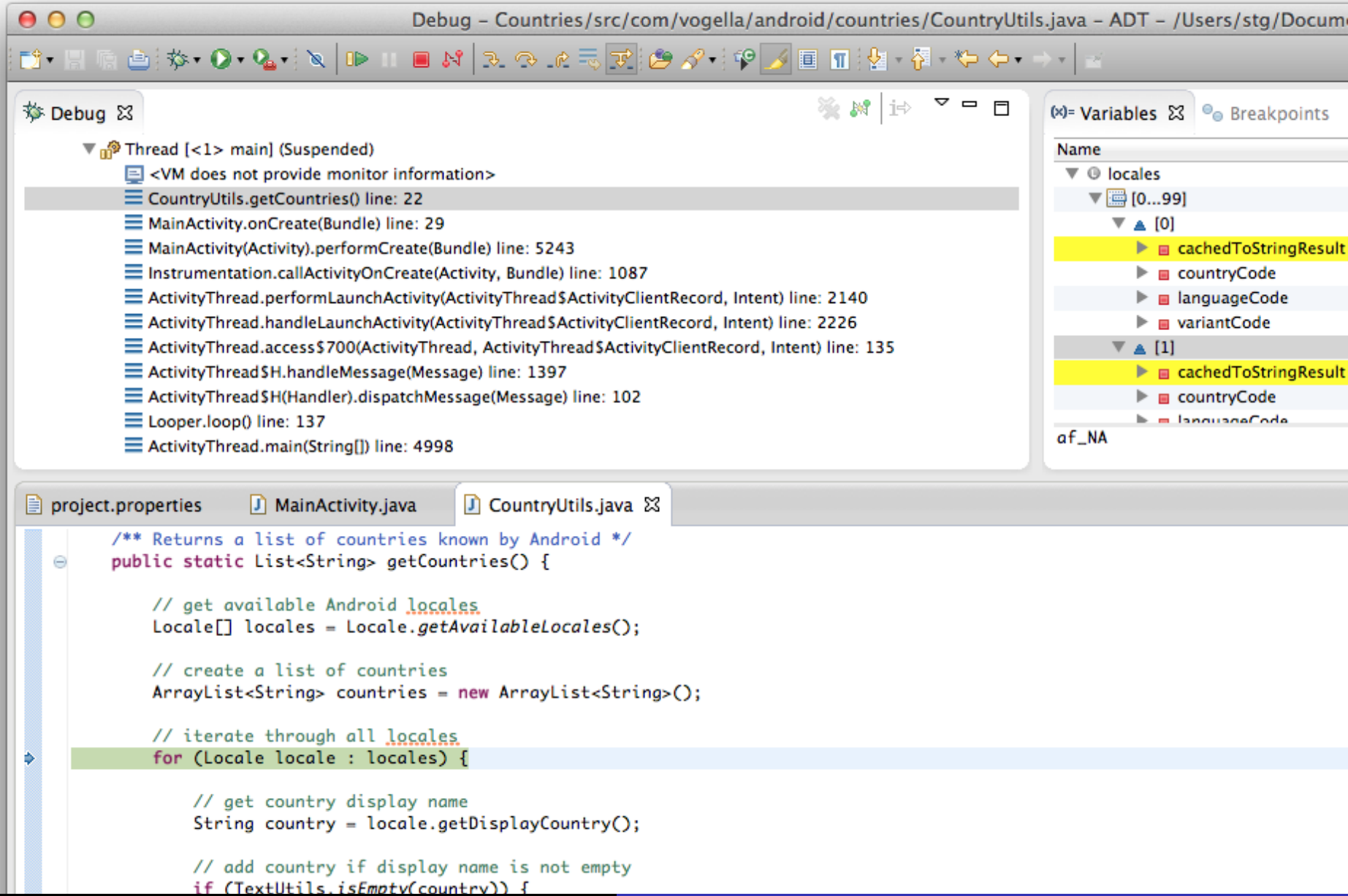
Problem identified: Restart the debugging session



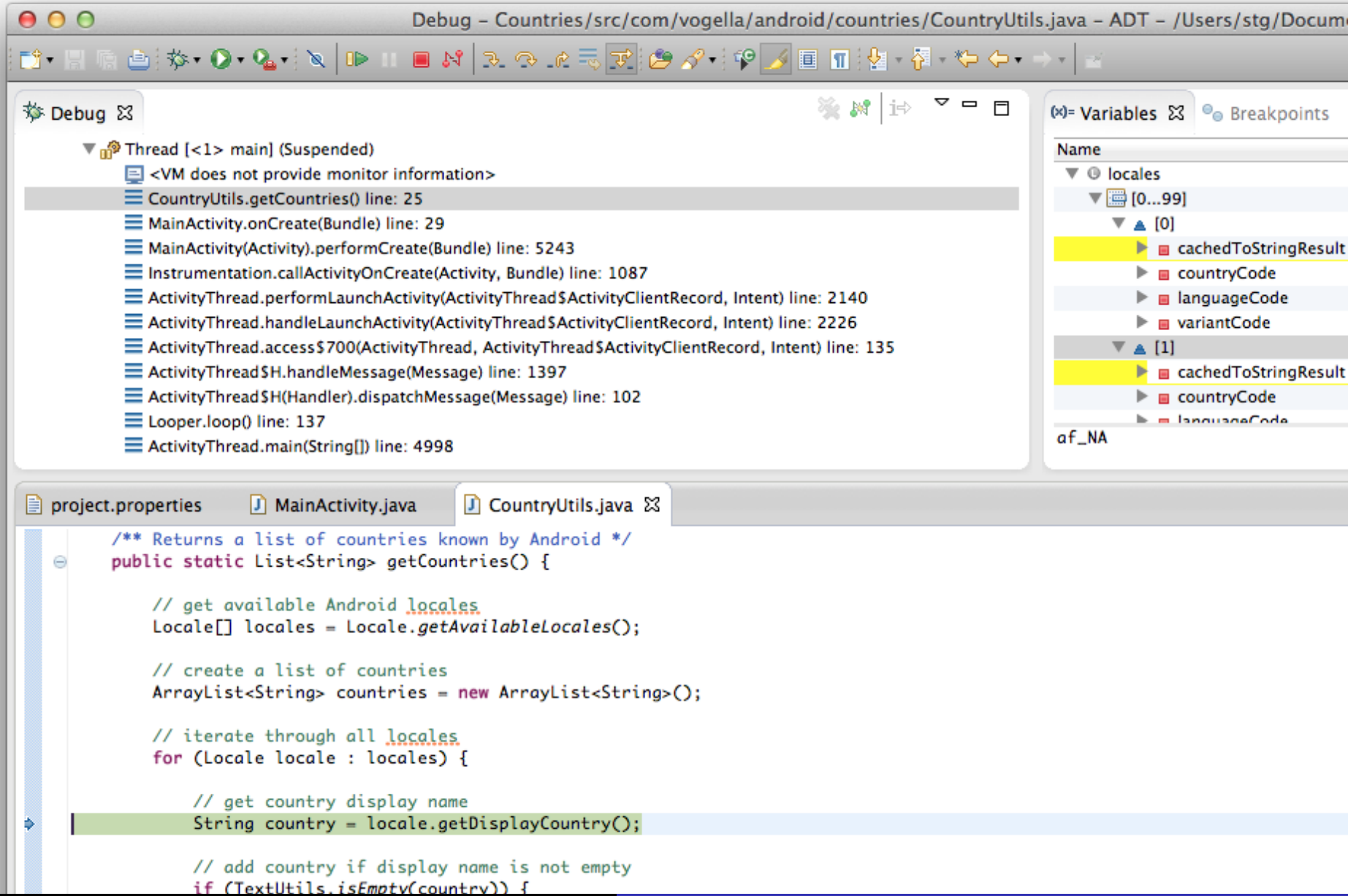
Entering the getCountries method



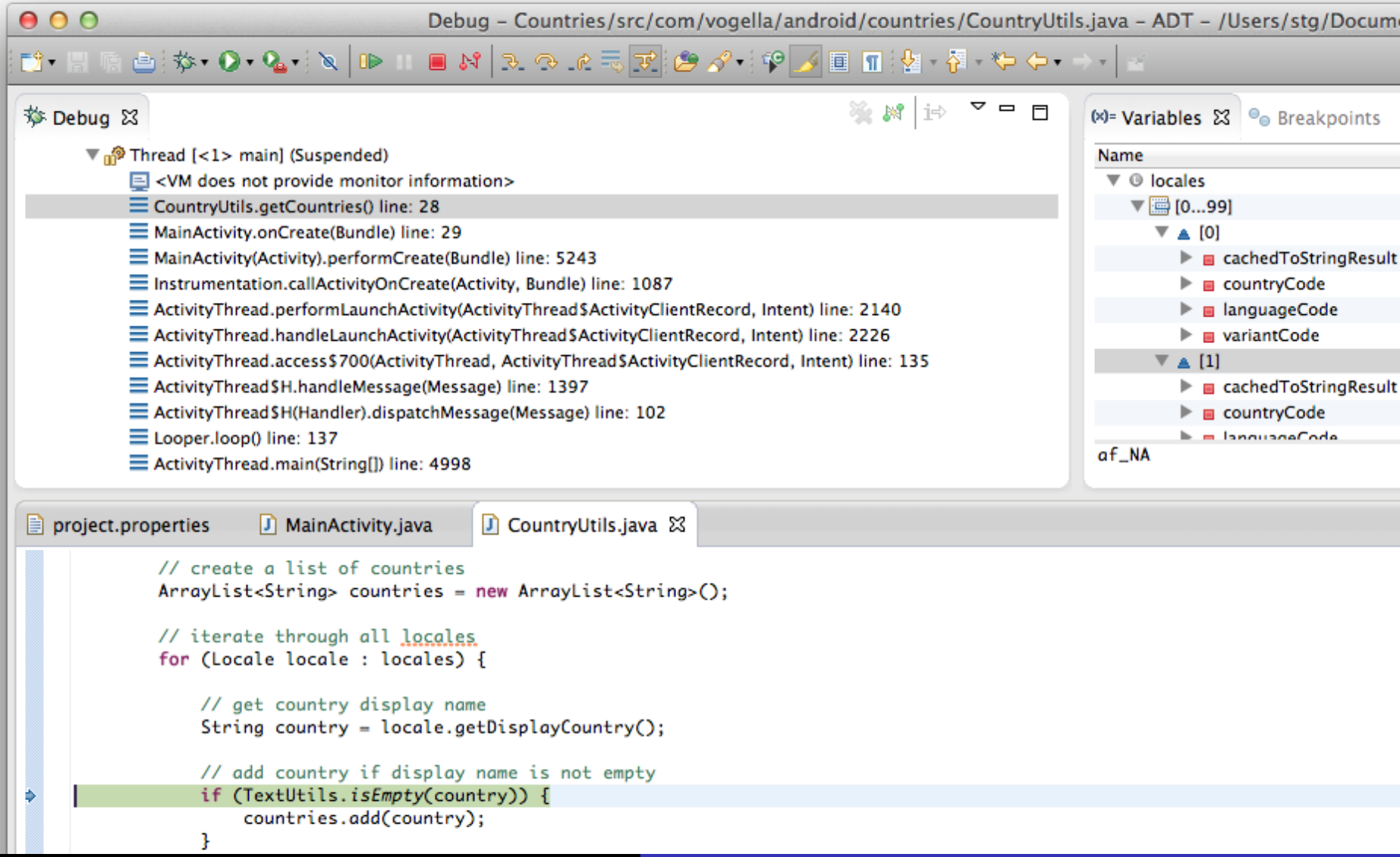
Continue executing, looking for updates to countries



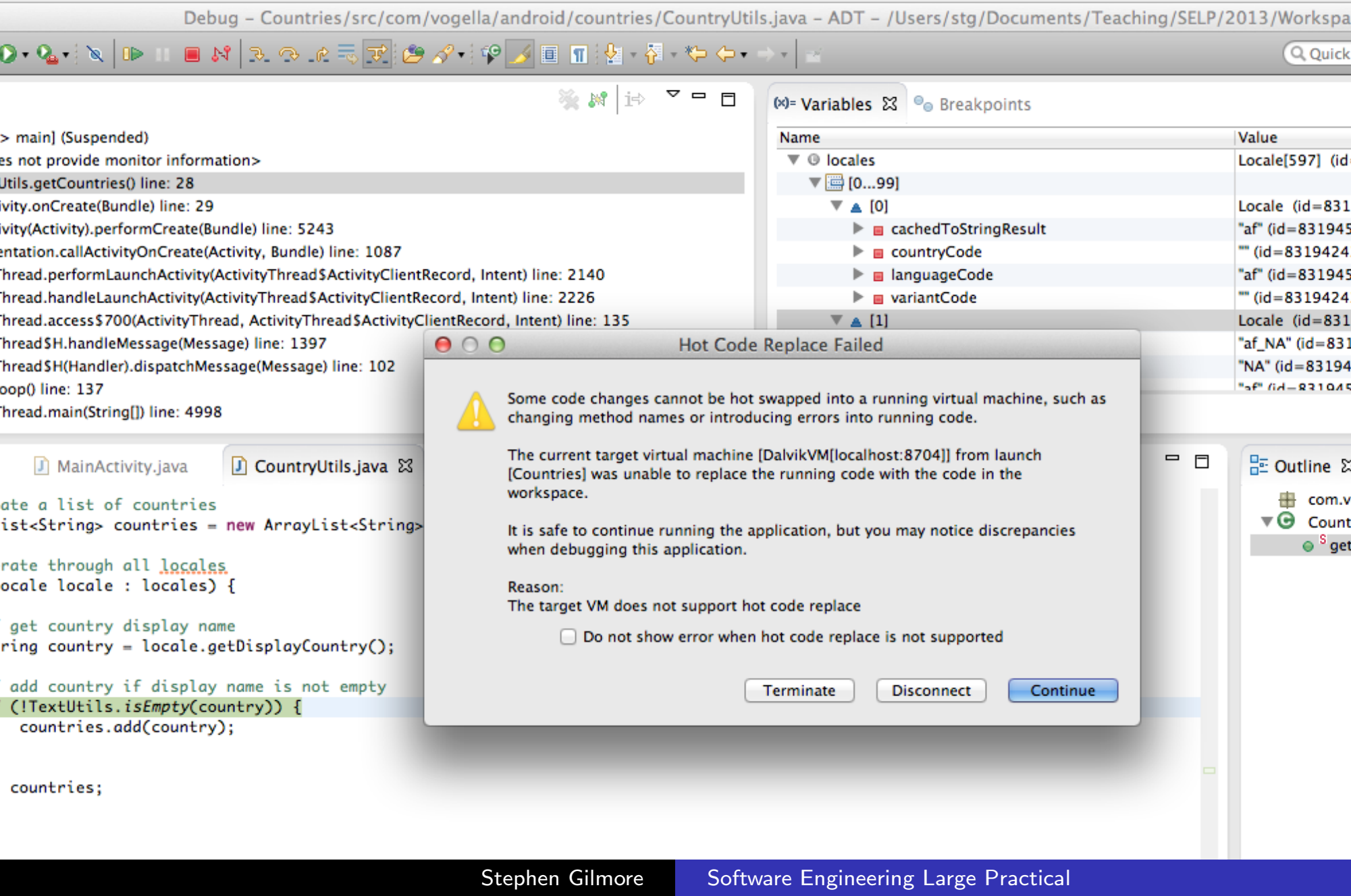
Continue execution



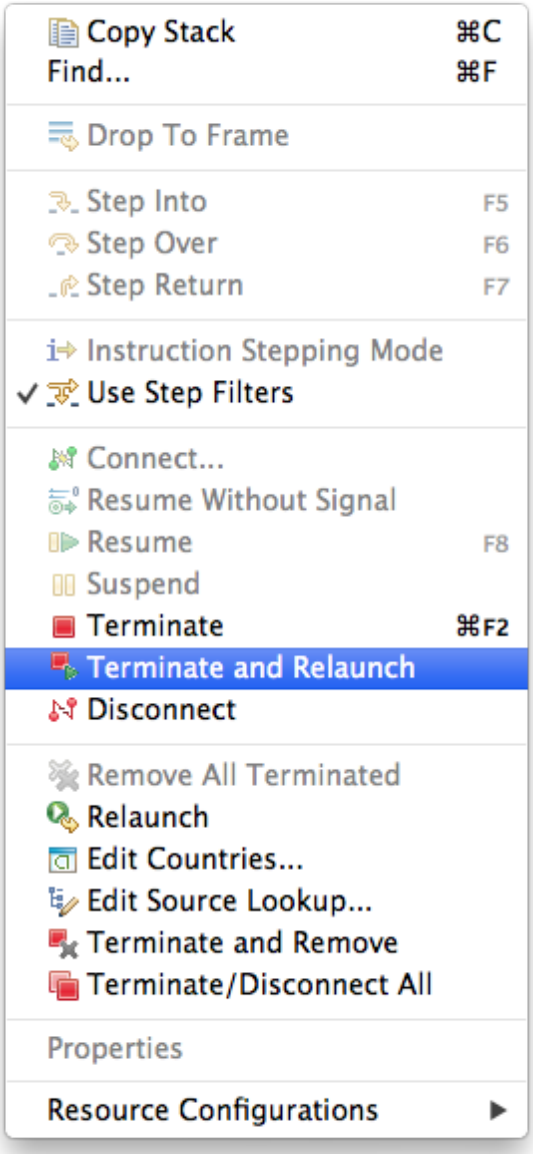
A simple logic error is discovered: we want the country name if it is *not* empty



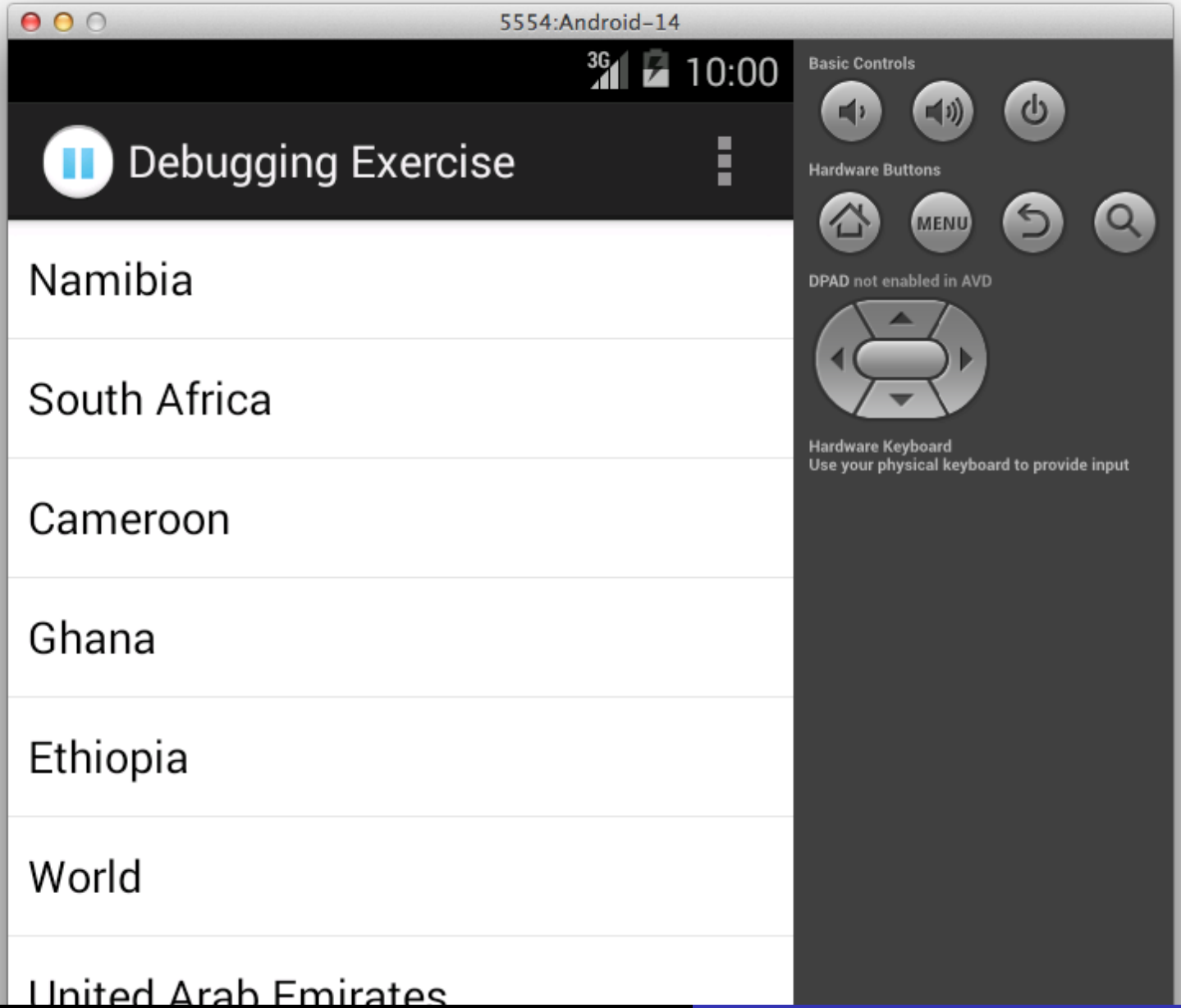
We are not able to swap in the new version of the code



We can terminate and relaunch this debugging session



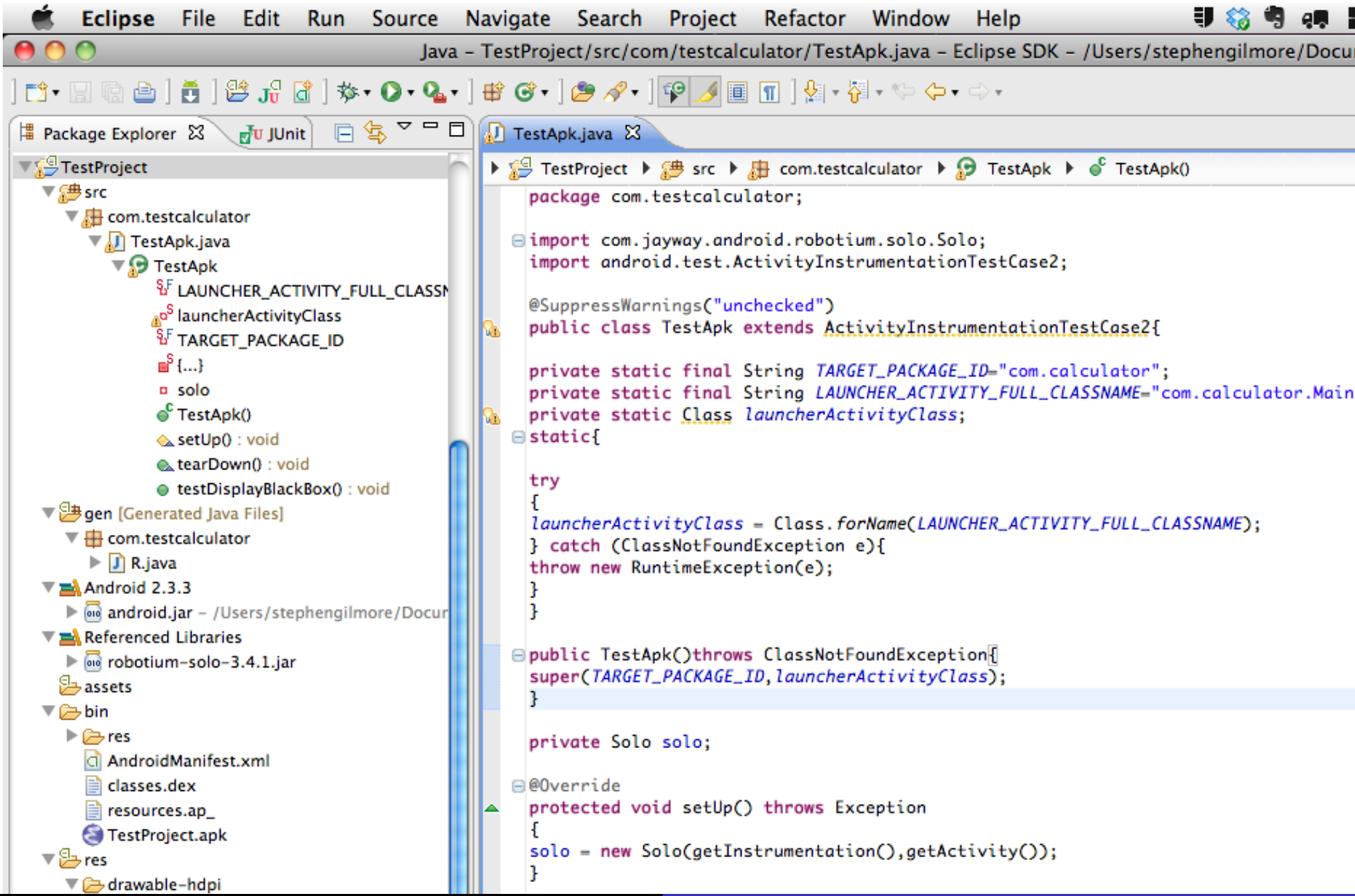
Success: a list of countries is displayed



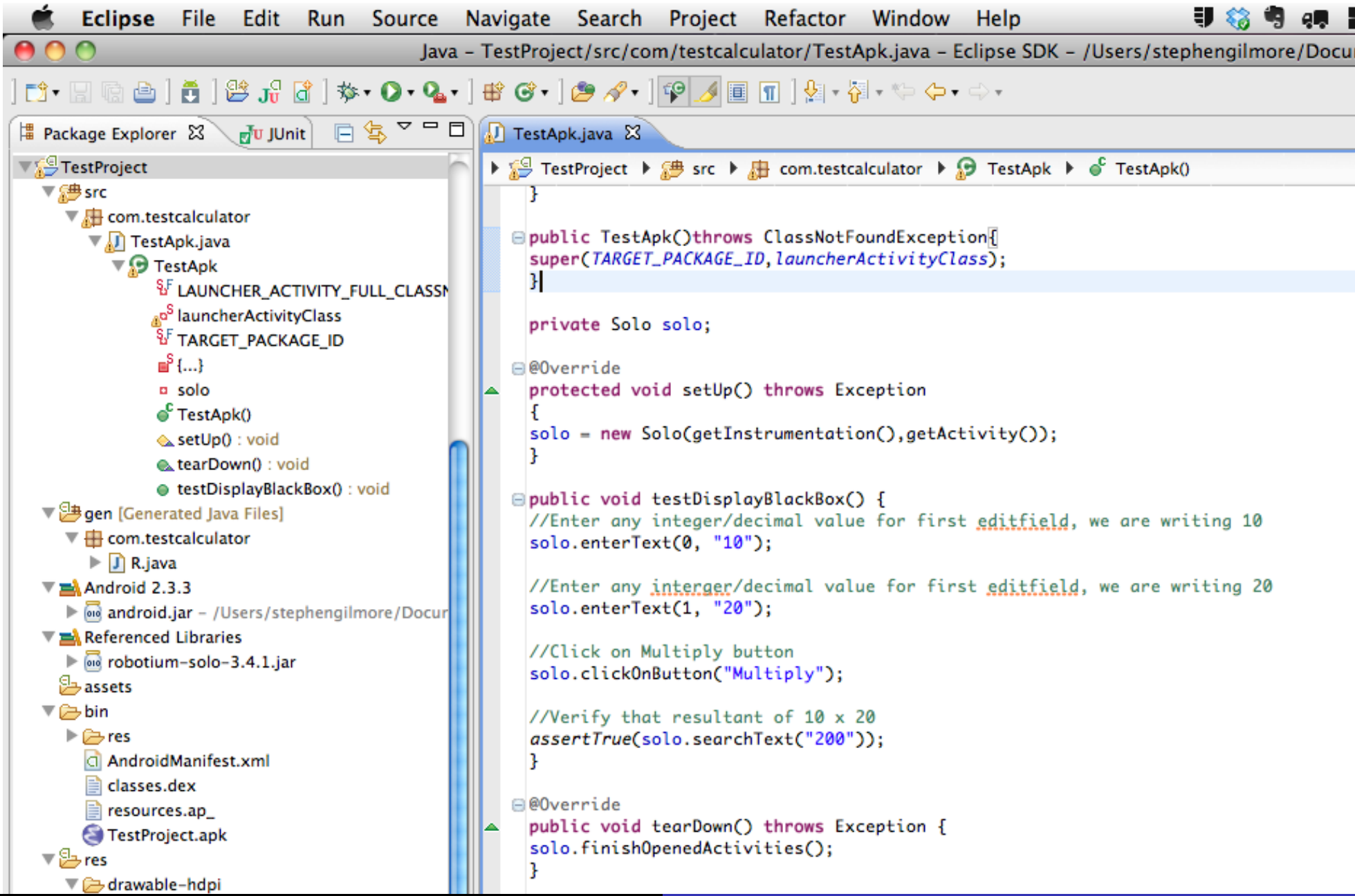
Software testing

A test framework called Robotium is available for testing Android applications. Using Robotium we can automate tests of Android apps, and collect statistics on which tests passed and which tests failed, using the JUnit test framework.

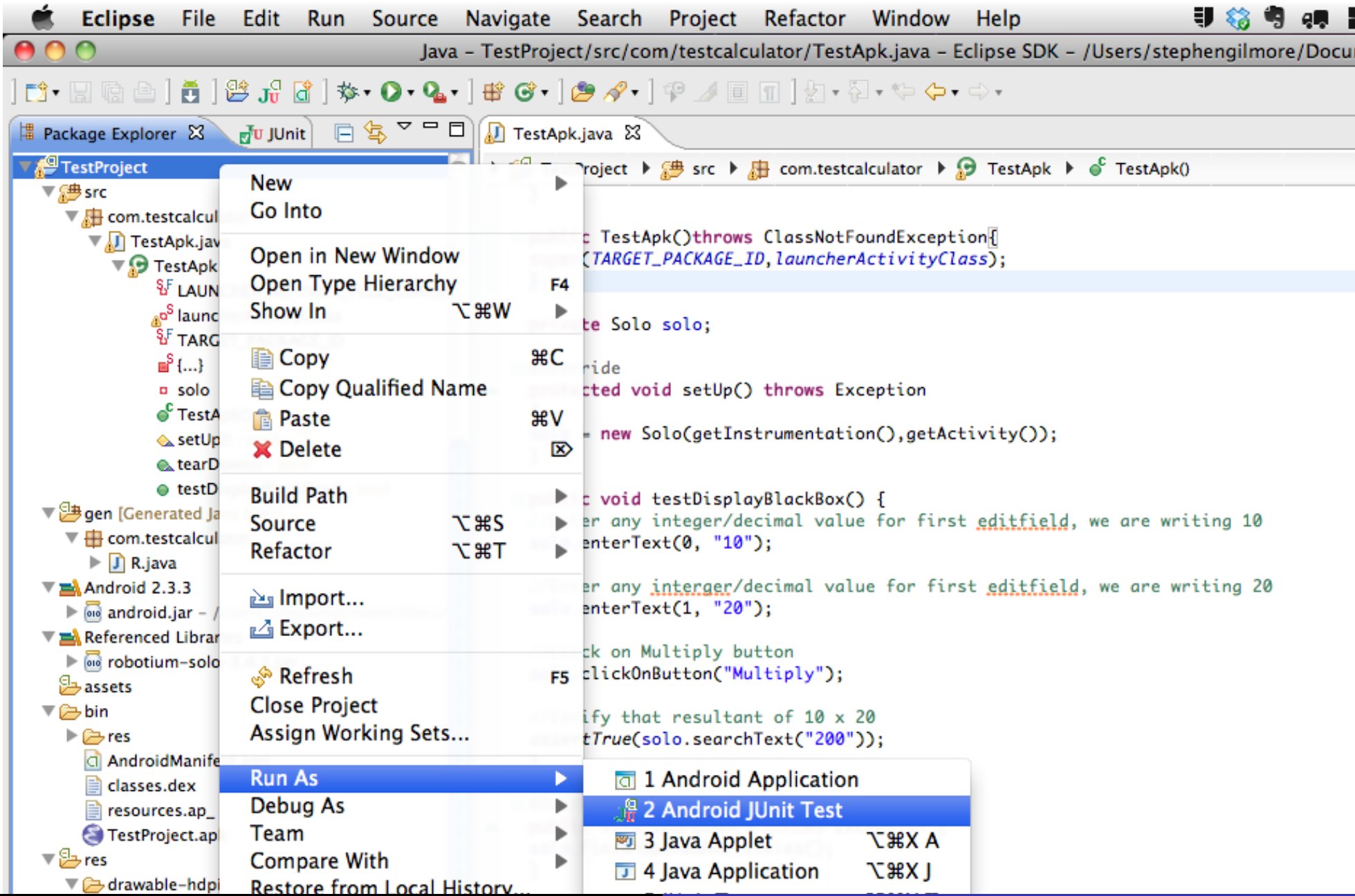
Import Robotium and make an instance of Solo



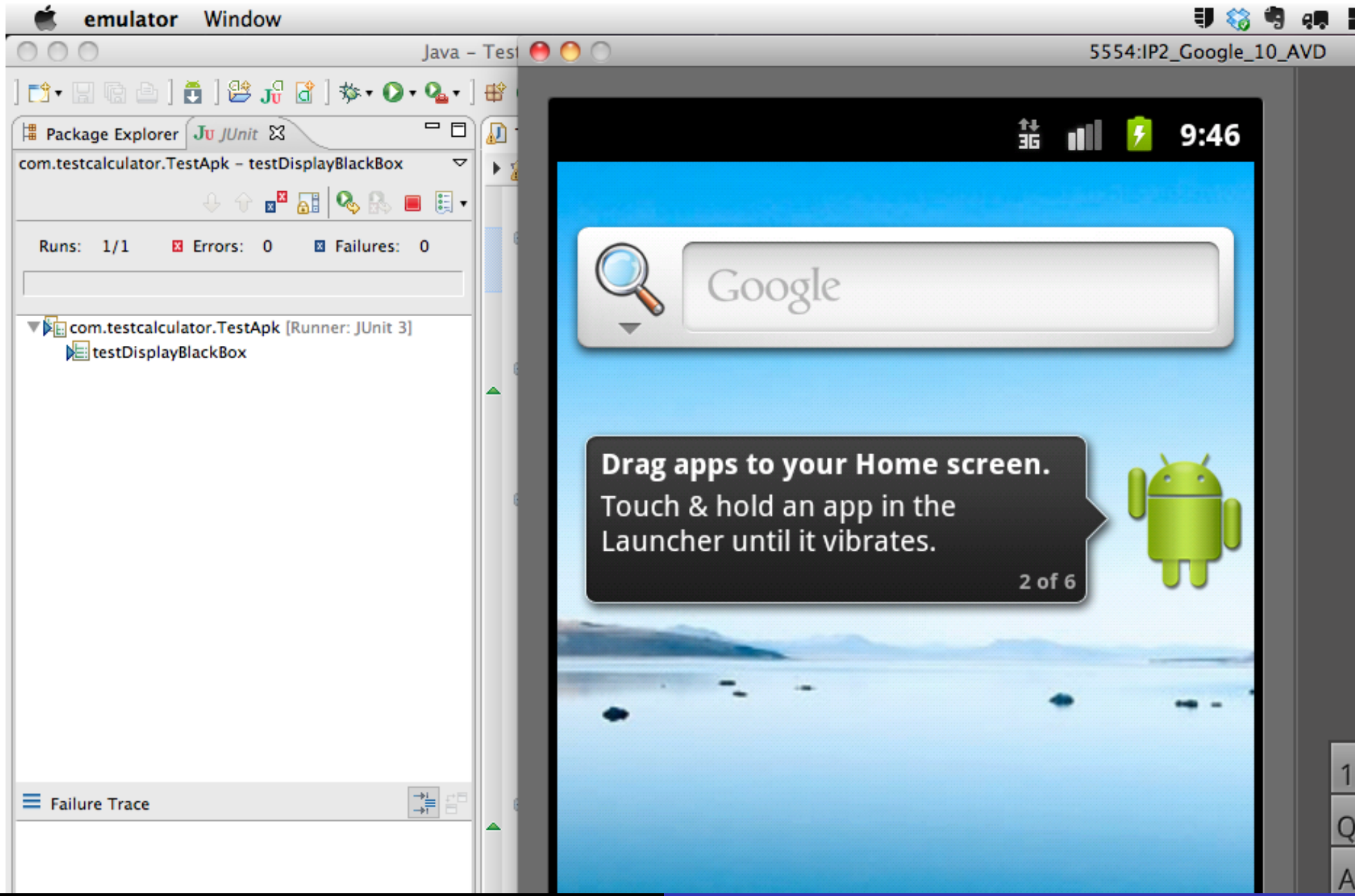
Write a void method whose name begins with "test"



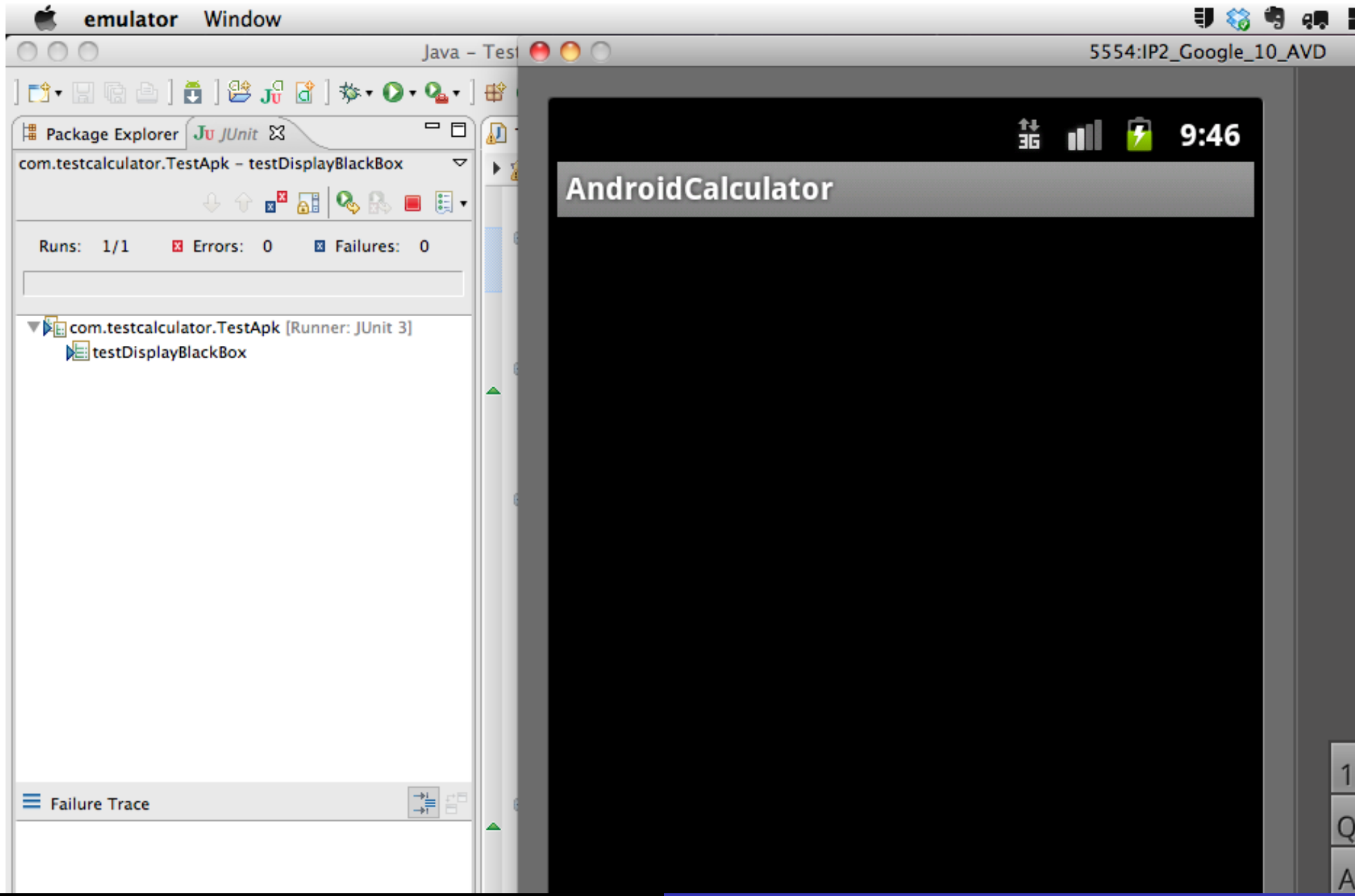
Run as an Android JUnit test



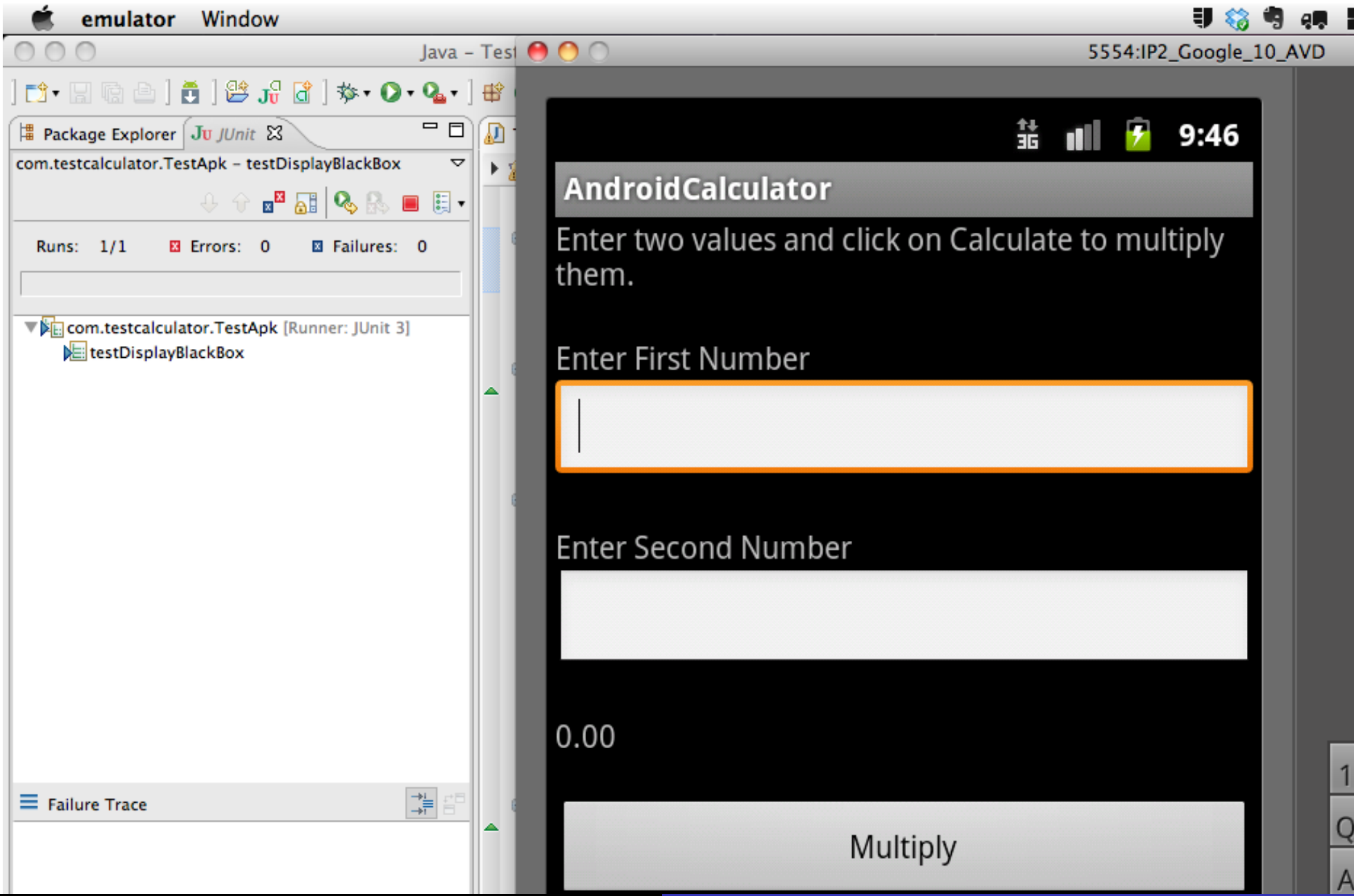
The emulator appears



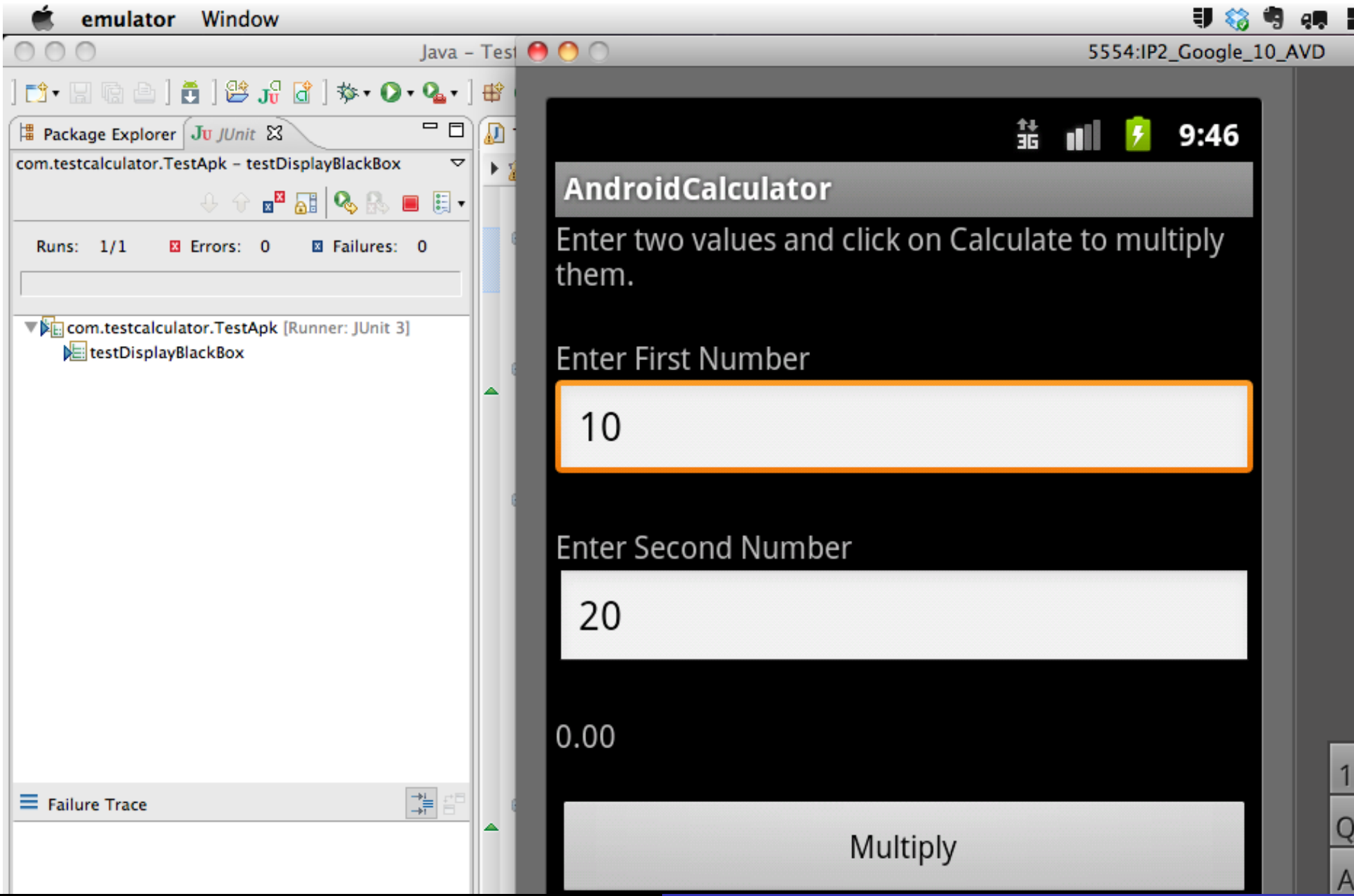
The app under test launches



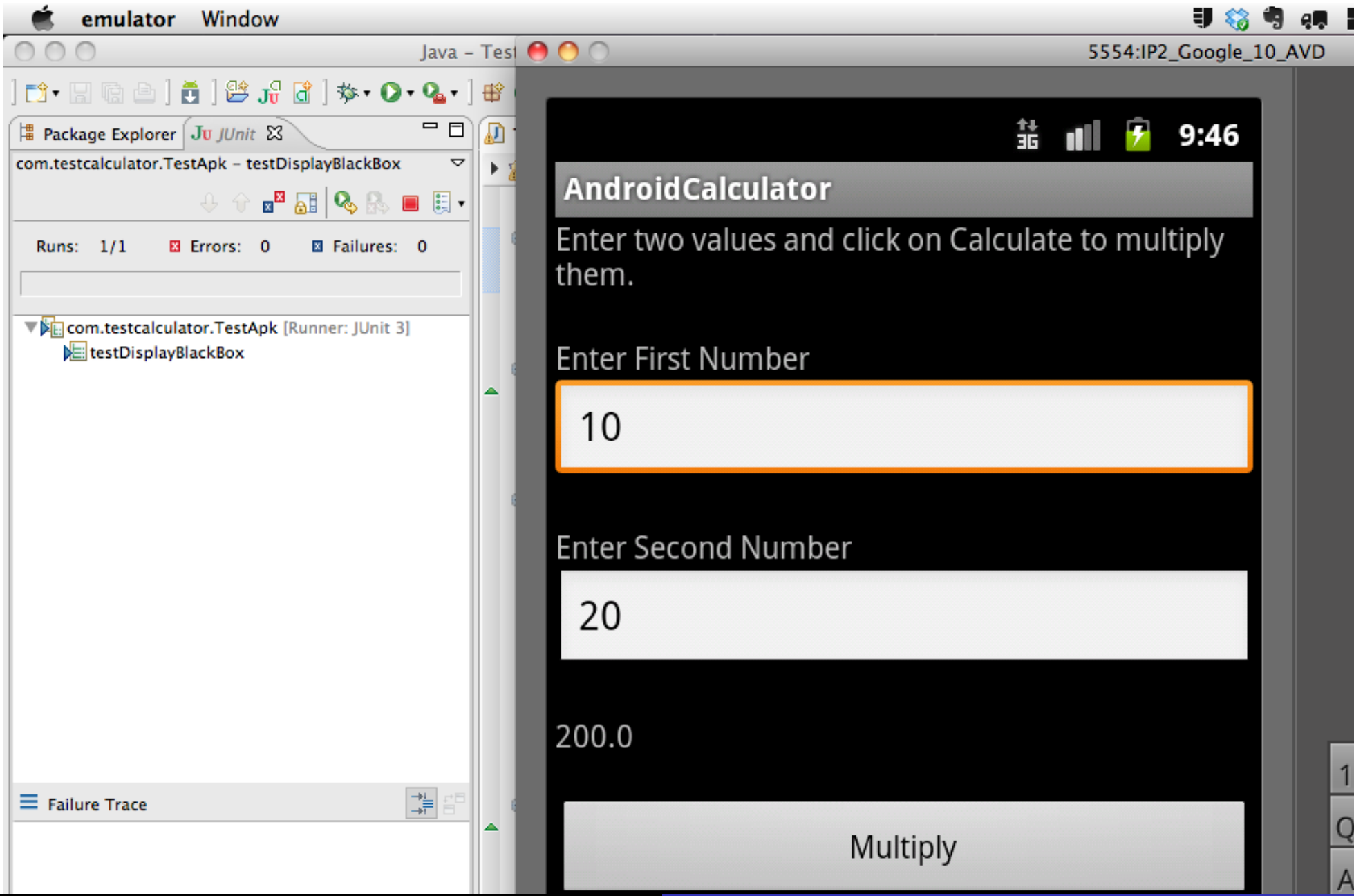
Values are entered in the fields



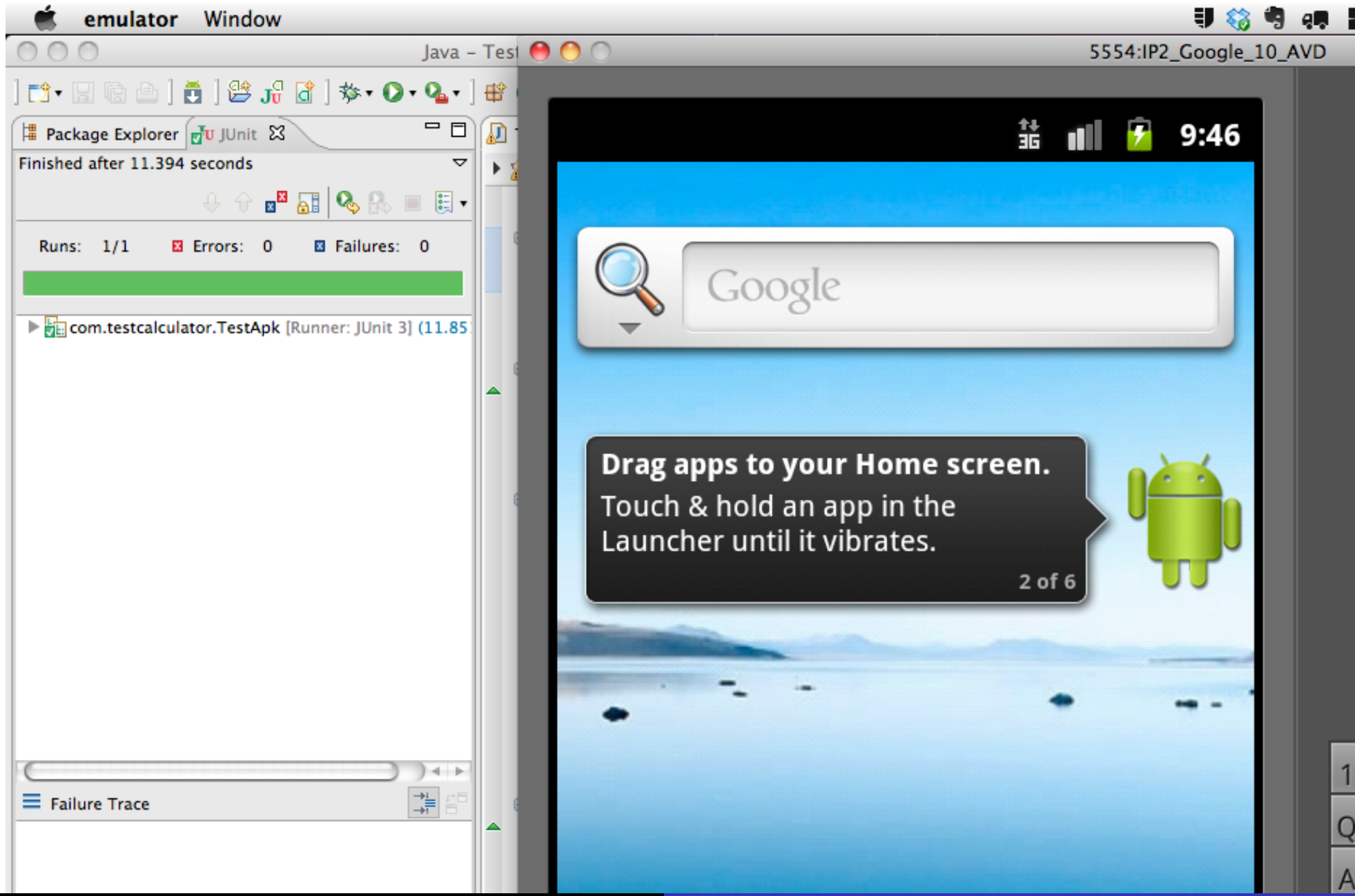
These values have been entered programmatically



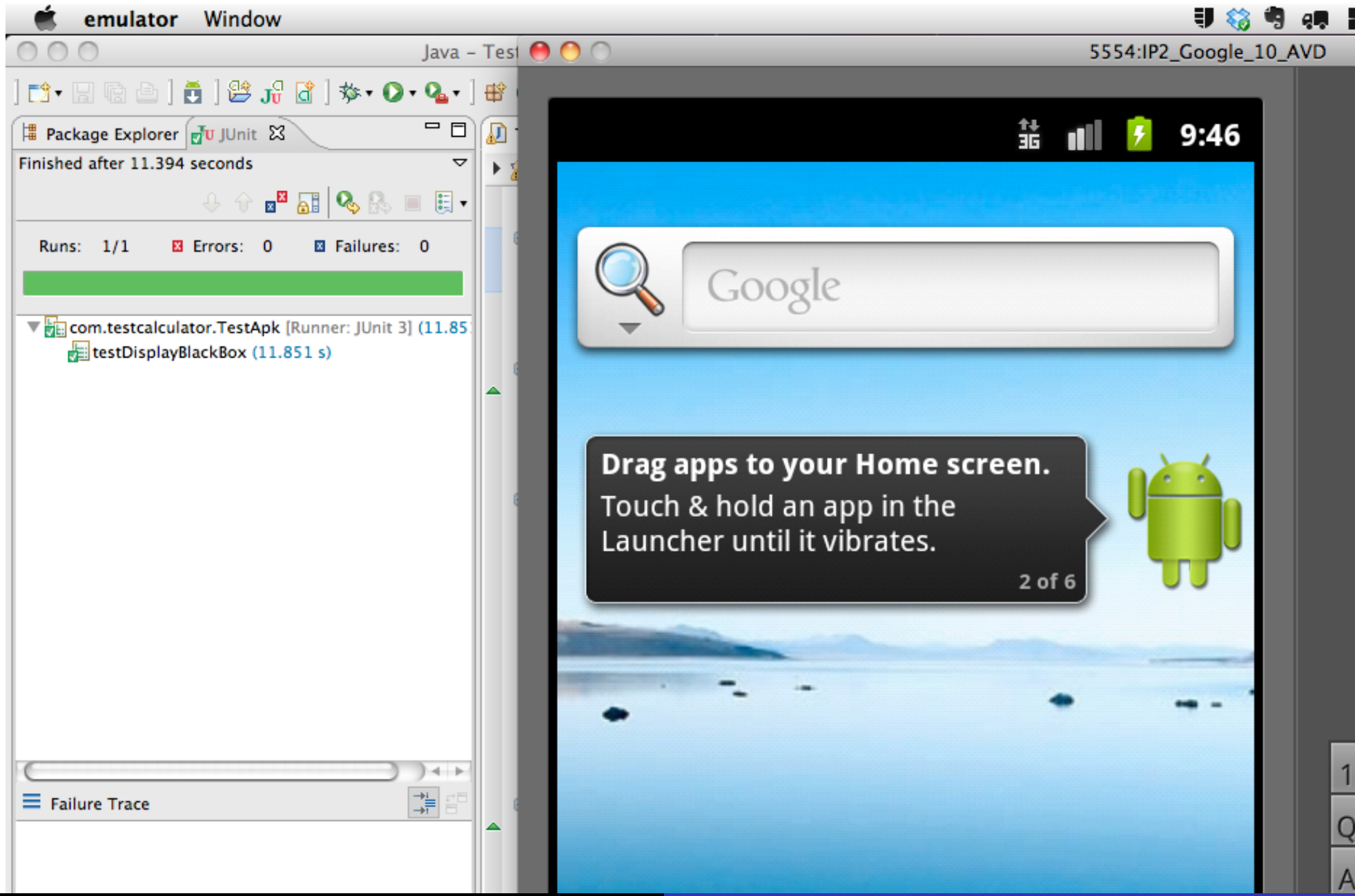
The button click also is performed for us



All tests passed (there was only one)



If some failed, we can see which



Software testing

A testing framework such as Robotium allows us to formally document a series of tests which we expect our app to pass. These are embedded in the code of the test harness. After every change to our code we can re-run the test harness and check that all tests still pass.

Automating testing in this way can turn a boring manual task which is a chore to do into a simple automatic task which is easy to re-run and operates entirely without human intervention.