Введение в разработку под Android

Семинар 14. The end.

Макаров И.О. 25.12.2020

Сегодня

- Автотестирование UI компонентов
- Подведение итогов
- Вопросы/пожелания

Automate user interface tests

https://developer.android.com/training/testing/ui-testing

User interface (UI) testing lets you ensure that your app meets its functional requirements and achieves a high standard of quality such that it is more likely to be successfully adopted by users.

One approach to UI testing is to simply have a human tester perform a set of user operations on the target app and verify that it is behaving correctly. However, this manual approach can be time-consuming, tedious, and error-prone. A more efficient approach is to write your UI tests such that user actions are performed in an automated way. The automated approach allows you to run your tests quickly and reliably in a repeatable manner.

Note: It is strongly encouraged that you use Android Studio for building your test apps, because it provides project setup, library inclusion, and packaging conveniences. This class assumes you are using Android Studio.

To automate UI tests with Android Studio, you implement your test code in a separate Android test folder (`src/androidTest/java`). The Android Plug-in for Gradle builds a test app based on your test code, then loads the test app on the same device as the target app. In your test code, you can use UI testing frameworks to simulate user interactions on the target app, in order to perform testing tasks that cover specific usage scenarios.

Automate user interface tests

https://developer.android.com/training/testing/ui-testing

For testing Android apps, you typically create these types of automated UI tests:

- *UI tests that span a single app:* This type of test verifies that the target app behaves as expected when a user performs a specific action or enters a specific input in its activities. It allows you to check that the target app returns the correct UI output in response to user interactions in the app's activities. UI testing frameworks like Espresso allow you to programmatically simulate user actions and test complex intra-app user interactions.

  Test UI for a single app - Espresso testing framework

- *UI tests that span multiple apps:* This type of test verifies the correct behavior of interactions between different user apps or between user apps and system apps. For example, you might want to test that your camera app shares images correctly with a 3rd-party social media app, or with the default Android Photos app. UI testing frameworks that support cross-app interactions, such as UI Automator, allow you to create tests for such scenarios.

  Test UI for multiple apps - UI Automator testing framework

Дома

https://developer.android.com/training/testing/ui-testing#additional-resources-codelabs

- BasicSample: Basic UI Automator sample.
- Espresso Code Samples includes a full selection of Espresso samples.
- Testing Basics
- Dependency Injection and Test Doubles
- Survey of Testing Topics