**Ручное и автоматизированное тестирование   
Android-приложений**

*Ковалевская А.А.  
Дыкоменко К.С.*

*Научный руководитель: Макаренко Н.В.*

*Филиал учреждения образования "Белорусский государственный технологический университет" "Витебский государственный технологический колледж"*

**Цель исследования**: изучить особенности ручного и автоматизированного тестирования Android-приложений.

**Тезисы**:

1. Ручное тестирование Android-приложений

В данном разделе раскрыто понятие ручного тестирования, рассмотрены основные моменты, которые необходимо протестировать в Android-приложении, а также приведен пример тест-кейса и описания бага.

2. Автоматизированное тестирование Android-приложений

В данном разделе раскрыто понятие автоматизированного тестирования программного обеспечения и описана ситуация, когда его целесообразно применять.

2.1 Android SDK

В данном подразделе описывается, для чего служит Android SDK.

2.2 Запись и воспроизведение теста

В данном подразделе описываются возможности инструмента MonkeyRunner, а также перечень этапов работы с MonkeyRecorder.

**Введение**

В связи с ростом количества смартфонов и устройств на Android все большую популярность набирают мобильные приложения.

Мобильный пользователь ожидает, что устанавливаемые им приложения просты, интуитивно понятны и работают без сбоев. Поэтому качество приложения играет очень большую роль в его популярности.

Качество приложения – это совокупность характеристик данного приложения, относящихся к его способности удовлетворять установленные и предполагаемые потребности пользователей. Оценить уровень качества приложения и выявить различные ошибки в его работе можно с помощью тестирования.

Тестирование – это поиск дефектов или багов (от англ. bug (жук)) в приложении. Термин «баг» используют для обозначения ошибки, из-за которой приложение выдает неожиданное поведение. По одной из версий данный термин стал активно использоваться, после того как в 1946 году учёные Гарвардского университета, тестировавшие вычислительную машину Mark II Aiken Relay Calculator, нашли мотылька, застрявшего между контактами электромеханического реле, и Грейс Хоппер (американский ученый, одна из первых, кто писала программы для гарвардского компьютера Марк I) произнесла этот термин. Извлечённое насекомое было вклеено скотчем в технический дневник с сопроводительной надписью: «First actual case of bug being found» («первый реальный случай, когда был найден жук»).

Планируя процесс тестирования тестировщик часто составляет тест-кейсы. В самом простом случае тест-кейс включает в себя:

* краткое описание (то, что тестируем);
* последовательность действий (шаги, которые выполняет тестировщик, чтобы проверить какую-либо функциональность);
* ожидаемый результат (то, что должно получиться после выполнения действий).

В дальнейшем по данным тест-кейсам тестируется приложение. Выполняя последовательность действий из тест-кейса, тестировщик получает фактический результат, который сравнивает с ожидаемым. Если данные результаты не совпадают, то это означает, что тестировщик нашел баг.

В данном исследовании мы рассмотрим особенности ручного тестирования Android-приложений, а также изучим автоматизированное тестирование с помощью инструмента MonkeyRunner.

# Ручное тестирование Android-приложений

Под ручным понимают тестирование, которое проводится тестировщиком без специальных утилит или это такое тестирование, при котором тестировщик располагается за компьютером (планшетом, телефоном) и производит самостоятельно те же действия, что и пользователь.

Тестирование мобильных приложений существенно отличается от тестирования приложений, предназначенных для использования на персональных компьютерах.

Приведем ряд основных моментов, которые нужно протестировать:

* Установка и запуск приложения, выход из приложения, повторный вход, удаление приложения с мобильного устройства.
* Мультитач и размер экрана. Корректность удаления 2-х элементов или просмотр двух элементов, нажатием на них одновременно. Проверка многократного быстрого нажатия на кнопку – часто при этом может случиться падение приложения. В приложении должны отсутствовать пустые экраны, чтобы пользователь не оказался в ситуации, в которой не очевидно, что делать. Также все элементы должны быть такого размера, чтобы пользователь мог однозначно нажать на них.
* Стабильность. Работа приложения при множестве запущенных приложений и долгое время, а также в случае недостатка места для установки или работы приложения. Поведение приложения при отсутствии в некоторых устройствах поддерживаемых приложением функций.
* Адаптация приложения к портретной и альбомной ориентациям устройства.
* Стресс. Реакция приложения на внешние прерывания:
* входящие и исходящие SMS, MMS, звонки, оповещения других приложений;
* переход устройства в режим ожидания;
* выключение устройства, разрядка устройства;
* зарядка устройства;
* отключение интернета;
* переход в другое приложение.
* Интернационализация. Проверка корректности работы приложения на разных языках (если данное приложение мультиязычное).
* Обратная связь с пользователем. Наличие информативных сообщений при попытке выполнить какое-либо действие (например, при удалении важной информации), а также присутствие визуальной индикации хода выполнения функций. У всех нажимаемых элементов должно быть «нажатое состояние» (отклик на действие), благодаря этому пользователь всегда будет видеть, действительно ли произошло нажатие.
* Обновление. Корректность обновления приложения до новой версии.
* Орфографические ошибки.

Как показывает практика тестирования мобильных приложений, наиболее корректной работы приложения можно добиться при ручном тестировании на реальных мобильных устройствах.

Нами было произведено тестирование мобильного приложения «FM» (файловый менеджер для Android). Планируя процесс тестирования мы составили ряд тест-кейсов. Например, тест-кейс для проверки неактивности кнопки перехода по закладке, если данная закладка была удалена:

|  |  |  |
| --- | --- | --- |
| **TC ID/Priority** | **51o001** | **средний** |
| **IDEA**: Неактивность кнопки перехода по закладке, если данная закладка была удалена | | |
| **Revision History** | | |
| **Created on**: 09/02/2015 by А. Ковалевская, К. Дыкоменко | | Новый тест-кейс |
| **Execution part** | | |
| **PROCEDURE** | | **EXPECTED RESULT** |
| 1. Запустить приложение FM  2. Нажать на иконку с изображением дискеты для добавления закладки  3. Ввести в поле «Имя» название закладки, например, «Главная»  4. Нажать на пункт списка «Закладки»  5. Нажать на пункт списка «Главная»  6. Удалить закладку (нажать на иконку с изображением корзины в правом верхнем углу)  7. В окне подтверждения удаления нажать «Yes»  8. Попытаться нажать на кнопку-стрелку рядом с надписью «Избранное» | | Кнопка-стрелка неактивна. |

TC ID – уникальный идентификатор тест-кейса (или номер тест-кейса)

Priority – приоритет: низкий, средний, высокий

IDEA – краткое описание того, что проверяется с помощью данного тест-кейса; здесь же находятся предпосылки для тест-кейса (необходимые данные для проверки по данному тест-кейсу)

Revision History – история изменений. Содержит информацию о дате создания/изменения тест-кейса, имени тестировщика, а также действиях: новый тест-кейс; тест-кейс изменен (можно указать, что конкретно менялось).

PROCEDURE – последовательность шагов (сценарий)

EXPECTED RESULT – ожидаемый результат

Протестировав приложение по данному тест-кейсу, мы обнаружили следующий баг:

**Краткое описание**: Активность кнопки перехода по закладке, после удаления закладки.

**Шаги по воспроизведению:**

1. Запустить приложение FM

2. Нажать на иконку с изображением дискеты для добавления закладки

3. Ввести в поле «Имя» название закладки, например, «Главная»

4. Нажать на пункт «Закладки»

5. Нажать на «Главная»

6. Удалить закладку (нажать на иконку с изображением корзины в правом верхнем углу)

7. В окне подтверждения удаления нажать «Yes»

8. Попытаться нажать на кнопку-стрелку рядом с надписью «Избранное»

**Фактический результат**: Кнопка-стрелка активна. После нажатия на нее появляется неинформативная ошибка, которая показана на рисунке 1:

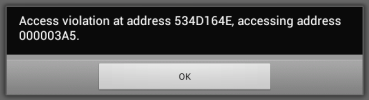


Рисунок 1 - Окно с ошибкой

**Ожидаемый результат**: Кнопка-стрелка неактивна.

# Автоматизированное тестирование Android-приложений

**Автоматизированное тестирование программного обеспечения (ПО)** — процесс тестирования программного обеспечения, при котором основные функции и шаги теста, такие как запуск, инициализация, выполнение, анализ и выдача результата, производятся автоматически с помощью инструментов для автоматизированного тестирования.

Планируя процесс тестирования, тестировщик пишет тест-кейсы. Затем он проверяет приложение по данным тест-кейсам. Когда тестировщик находит баги, то он сообщает о них разработчику (документирует баги; часто для хранения багов используются специальные баг-трэкинговые системы). После того как разработчик исправляет какой-либо баг, тестировщику необходимо заново выполнить тест-кейс, который обнаружил данный баг, т.е. повторно пройтись по всем шагам из этого тест-кейса (ручное тестирование). Также для надежности необходимо выполнить и все тест-кейсы, на которые мог повлиять найденный баг. В этом случае для ускорения процесса тестирования лучше первоначально автоматизировать тест-кейсы, чтобы потом просто запустить все тесты и посмотреть на результат их выполнения.

Используя инструмент MonkeyRunner, предназначенный для автоматизированного тестирования Android-приложений, мы автоматизировали ряд тест-кейсов для нашего мобильногоприложения «FM» (файловый менеджер для Android).

## Android SDK

MonkeyRunner входит в состав Android SDK.

SDK (от англ. software development kit) — комплект средств разработки, который позволяет специалистам по программному обеспечению создавать приложения для определённого пакета программ, программного обеспечения базовых средств разработки, аппаратной платформы, компьютерной системы, игровых консолей, операционных систем и прочих платформ.

Android SDK представляет собой эмулятор устройства и применяется для тестирования приложений под Android. Кроме того AndroidSDK предлагает набор инструментов для почти мгновенного создания приложений.

## Запись и воспроизведениетеста

Утилита MonkeyRunner предоставляет API для написания скриптов, которые управляют Android устройстами. С помощью MonkeyRunner можно написать скрипт на языке Python, который устанавливает Android приложение, запускает его, имитирует действия пользователя, снимает скриншоты и сохраняет их на компьютер. Утилита MonkeyRunner использует Jython (это реализация языка Python на языке Java) для выполнения скриптов.

С помощью MonkeyRunner можно запустить MonkeyRecorder, с помощью которого можно записывать различные действия, выполняемые на Android-устройстве. Однако MonkeyRunner оперирует координатами для выбора элемента управления, поэтому при изменении размеров экрана придется перезаписывать тест.

Перечень этапов работы:

* Установить Python (например, в c:\Python27)
* Прописать путь к c:\Python27 в переменной среды Path
* Установить Java
* Установить Jython
* Прописать путь в переменных среды к каталогу tools из AndroidSDK
* Скопировать следующие скрипты в папку tools:
* monkey\_recorder.py (это MonkeyRecorder)
* monkey\_playback.py (воспроизведение теста, записанного с помощью MonkeyRecorder)
* Запустить командную строку и ввести (предварительно запустить эмулятор и открыть приложение на нем или подключить реальное устройство (планшет, телефон) к компьютеру):
* monkeyrunner.bat monkey\_recorder.py
* Далее записать действия с помощью MonkeyRecorder и нажать на кнопку Export Actions. Сохранить файл с расширением txt (например, r1.txt).
* Для воспроизведения записанного теста в командной строке необходимо набрать следующее:
* monkeyrunner.bat monkey\_playback.py r1.txt

**Вывод**

При исследовании мы выявили, что в Android-приложениях существует ряд важных моментов, которые необходимо обязательно протестировать (например, мультитач и размер экрана, реагирование на внешние прерывания).

Мы провели ручное тестирование мобильного приложения «FM» (файловый менеджер для Android) по предварительно составленным тест-кейсам. В ходе данного тестирования мы нашли ряд багов, которые задокументировали, чтобы впоследствии данные баги можно было исправить.

Мы также узнали:

* об использовании Android SDK;
* о возможностях утилиты MonkeyRunner при автоматизированном тестировании Android-приложений;
* о возможностях MonkeyRecorder, а также применили его для автоматизации наших тест-кейсов.

Тем не менее, в ходе нашего исследования мы пришли к выводу, что автоматизированное тестирование не всегда целесообразно использовать, так как на написание сложных тестов уходит много времени, а запись действий с помощью MonkeyRecorder основана на координатах элементов, присутствующих в приложении и при изменении размера экрана придется переписывать тест. Поэтому ручное тестирование Android-приложений все чаще используется в современных IT-компаниях.

**Список использованных источников**

1. В.В. Липаев. Тестирование программ. /М.: Радио и связь, 1986. – 296с.

2. Г. Майерс. Искусство тестирования программ. / М.: Финансы и статистика,1982. – 176 с.

3. И.В. Винниченко. Автоматизация процессов тестирования. / СПб.: Питер,2005. – 203 с.

4. Л. Тамре. Введение в тестирование программного обеспечения. / М.: Издательский дом «Вильямс»,2003. – 368 с.

5. Р. Савин. Тестирование Dot COM, или Пособие по жестокому обращению с багами в интернет-стартапах. / М.: Дело,2007. – 312 с.

6. Э. Дастин, Д. Рэшка, Д. Пол. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация. / М.: Издательство «ЛОРИ»,2003. – 567 с.