

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

**АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ МОБИЛЬНОГО  
ПРИЛОЖЕНИЯ ДЛЯ ОС ANDROID**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы

направления 02.03.03 Математическое обеспечение и администрирование  
информационных систем

факультета компьютерных наук и информационных технологий

Улитина Егора Сергеевича

Научный руководитель

доцент кафедры ИиП

Е.В.Кудрина

Зав. кафедрой ИиП,

к.ф.-м.н., доцент

М.В. Огнева

Саратов 2021

## **ВВЕДЕНИЕ**

**Актуальность темы.** В последние годы мобильные устройства вошли в обиход практически каждого человека. В процессе взаимодействия с каким-либо приложением пользователи могут сталкиваться с различными дефектами в его работе, что негативно влияет на оценку качества программного продукта. Поэтому важно, чтобы всевозможные ошибки находились как можно раньше, а именно на этапе тестирования.

Тестирование программного обеспечения – процесс анализа программного средства и сопутствующей документации с целью выявления дефектов и повышения качества продуктов. Тестирование мобильных приложений – проверка работоспособности функционала мобильного приложения на различных мобильных устройствах.

На рынке существует огромное количество мобильных устройств с совершенно разными характеристиками. Но в тоже время практически все устройства используют одну из двух популярных операционных систем – Android или IOS. Согласно статистике statcounter (веб-сайт, который занимается анализом веб-трафика), доля операционной системы Android среди мобильных телефонов составляет 70,43%, а у его главного конкурента IOS - 29,06%. Исходя из приведенных цифр, можно сделать вывод, что Android является самой популярной операционной системой в мире, поэтому важно сделать так, чтобы проведение тестирования мобильных приложений на ОС Android было удобно и как можно более универсально.

**Цель бакалаврской работы** – разработать и реализовать фреймворк для автоматизированного тестирования мобильного приложения для ОС Android.

Поставленная цель определила **следующие задачи:**

1. Рассмотреть теорию развития программного обеспечения, изучить основные методы, виды и типы тестирования программного обеспечения.
2. Обобщить и систематизировать знания в области разработки мобильных приложений под ОС Android, приобретенные в ходе обучения.

3. Изучить современные методы, технологии и инструментальные средства, используемые для реализации автоматизированного тестирования мобильных приложений для ОС Android
4. Провести сравнительный анализ инструментов, используемых при разработке тестового фреймворка для ОС Android.
5. Реализовать тестовый фреймворк для автоматизированного тестирования мобильного приложения для ОС Android
6. Провести тестирование мобильного приложения, используя разработанный фреймворк.

**Методологические основы.** Автоматизированное тестирование мобильного приложения для ОС Android представлены в работах Дэвида Грэффитса [1], Маркуса Полса[2], Филлипа Билла[3] и Святослава Куликова [4].

**Практическая значимость бакалаврской работы.** В ходе выполнения бакалаврской работы был разработан тестовый фреймворк для ОС Android и проведено дымовое тестирование мобильной версии портала обучения информатики и программирования СГУ (<https://school.sgu.ru/>), реализованного на платформе LMS Moodle. Практическая ценность фреймворка заключается в том, что он позволяет проводить автоматизированное тестирование различных мобильных приложений на ОС Android.

**Структура и объём работы.** Бакалаврская работа состоит из введения, 2 разделов, заключения, списка использованных источников и 5 приложений. Общий объем работы – 106 страниц, из них 46 страниц – основное содержание, включая 14 рисунков и 1 таблицы, список использованных источников информации – 28 наименований.

## **КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ**

**Первый раздел «Теоретические основы тестирования программного обеспечения»** посвящен теории развития программного

обеспечения, основным методам, видам и типам тестирования, основным инструментам автоматизированного тестирования и их сравнительному анализу.

На рубеже двух десятилетий, в связи с появлением и бурным ростом популярности смартфонов, возникла необходимость в тестировании мобильных приложений, а также в автоматизации этого процесса, т.к. это позволит создавать мобильные приложения быстрее, качественнее и эффективнее.

Автоматизированное тестирование программного обеспечения – это процесс проверки программного обеспечения, который включает проведение таких основных функций и шагов теста, как запуск, инициализация, выполнение, анализ и выдача результата, автоматически посредством специализированных инструментов.

Дымовое тестирование – проверка наиболее важного и значительного функционала приложения. Без корректного выполнения данного типа тестирования нет необходимости проводить какие-то более глубокие виды тестирования. Приложение, во время дымового тестирования которого обнаружилось большое количество дефектов, является бессмысленным в его использовании. Дымовое тестирование проводится сразу после обновления функционала для того, чтобы была возможность определить уровень качества приложения и принять решение о необходимости проведения расширенного тестирования и тестирования критического пути. Поскольку тест-кейсов на уровне дымового тестирования относительно немного, а сами они достаточно просты, но при этом очень часто повторяются, они являются приоритетными при автоматизации.

Исходя из принятой методологии на проекте во время проведения тестирования разрабатывается определенное количество артефактов (документов). При тестировании принято составлять следующие артефакты: тест-план, тест-кейсы, чек-листы, тестовые наборы, баг-репорты, отчет о результатах тестирования.

Android – операционная система, разработанная Google, с открытым исходным кодом, преимущественно используемая в мобильных устройствах.

Один из самых главных плюсов автоматизации мобильного тестирования – быстрое выполнение регрессионного тестирования. Регрессионное тестирование – это вид тестирования направленный на проверку изменений, сделанных в приложении или окружающей среде, для подтверждения того факта, что существующая ранее функциональность работает как и прежде. Регрессионное тестирование автоматизируют чаще всего, что избавляет тестировщика от повторения одних и тех же тестов перед каждым релизом.

При реализации собственного тестового фреймворка необходимо правильно выбрать средство автоматизации, что является достаточно сложной задачей, потому что существует огромное множество инструментов. Поэтому важно оценить каждый инструмент по некоторым критериями, и прийти к выводу какой из них является более подходящим для использования во фреймворке. Сравнительный анализ инструментов автоматизированного тестирования мобильных приложений представлен в таблице 1

Таблица 1 – Сравнение инструментов автоматизированного тестирования мобильных приложений

Критерий сравнения	Robotium	UIAutomator	Appium	Calabash
Языки программирования	Java	Java, Kotlin	Ruby, Python, Java, JavaScript, PHP, C#, Objective-C, Kotlin	Ruby, JRuby
Тип	Драйвер	Драйвер	Надстройка	Надстройка
Возможность тестирования на эмуляторе	Да	Да	Да	Да
Возможность тестирования на реальном устройстве	Да	Да	Да	Да

Поддержка компании-разработчика	Да	Да	Да	Нет
Активность сообщества	Низкая	Средняя	Высокая	Средняя
Документация	Плохая	Хорошая	Хорошая	Средняя
Кросс-платформенность	Нет	Нет	Да	Да
Необходимость доступа к коду	Нет	Нет	Нет	Нет
Возможность использования программной клавиатуры	Нет	Да	Да	Да
Скорость выполнения тестов	Быстрая	Быстрая	Медленная	Средняя.
Множественный выбор элементов	Нет	Нет	Да	Нет
Доступ к клавишам специального назначения	Частично	Да	Да	Нет
Возможность тестирования гибридных приложений	Частично	Частично	Да	Да
Универсальность	Нет	Да	Да	Нет

Из всех средств был выделен Appium, который предоставляет широкий функционал, способен выполнять тестирование гибридных приложений, а также активно поддерживается сообществом и компанией-разработчиком. Он и был выбран в качестве основного инструмента для разработки собственного фреймворка.

**Второй раздел «Реализация собственного тестового фреймворка для тестирования мобильного приложения на ОС Android»** посвящен созданию тестового фреймворка для автоматизированного тестирования для ОС Android и проведению тестирования мобильного приложения.

В рамках тестирования необходимо было выполнить дымовое тестирование приложения через GUI приложение Moodle, используя платформу school.sgu.ru. Тестирование проводилось только роли обучающегося.



Рисунок 1 - Вход в личный кабинет

Функционал приложения, который был подвергнут тестированию:

- Ввод сайта на стартовой страничке.
- Вход пользователя в аккаунт.
- Регистрация.
- Сброс пароля.
- Нахождение курса через категории и через механизм поиска.
- Сообщения.
- Новости сайта.
- Запись на открытый и закрытый курс.
- Отправка ответа на задание.
- Просмотр разделов курса.

Для корректного проведения тестирования приложения были составлены следующие артефакты: тест-план, тест-кейсы, баг-репорты, отчет о результатах тестирования.

Созданный тест-план содержит в себе цель тестирования, описание областей, подвергаемых и не подвергаемых тестированию, критерии качества

и критического успеха тестирования, тестовую стратегию, описание команды тестирования, ресурсы, расписание, риски, планируемую документацию, классификацию багов и метрики.

Все созданные тест-кейсы были разбиты на несколько модулей: подключение к сайту, регистрация и вход личный кабинет, курс, новости, сообщения. Некоторые модули также содержат подмодули. В общей сложности было создано около 30 тест-кейсов.

По результатам проведенного тестирования было найдено 5 дефектов, которые были занесены в баг-репорт.

Отчет о результатах тестирования содержит краткое описание выполненной работы, описание процесса тестирования, расписание, статистику по новым дефектам, список найденных багов, рассчитанные значения метрики.

Был создан тестовый фреймворк, используя следующие инструменты тестирования:

1. Надстройка: Appium.
2. Язык программирования: Java
3. Среда разработки: IntelliJ IDEA
4. Драйвер: Selenium Webdriver
5. Тестовый фреймворк: TestNG
6. Отчетность: Allure
7. Сборщик проекта: Maven
8. Логирование: log4j

Один из самых важных критериев качественного фреймворка является его гибкость, в случае создания фреймворка для мобильного тестирования необходимо, чтобы была возможность легко изменить тестируемое приложение и используемые мобильные устройства. Для этого был создан файл `test.properties`, в котором хранятся параметры приложения, эмулятора и реального устройства.



Создание объектов происходит в классе «DriverManager» из пакета «Driver» . «DriverManager» создает и закрывает рабочую сессия, эмулятор и драйвер. Класс написан на основе паттерна singleton, который позволяет создать объект драйвера только в случае его необходимости.

Драйвер имеет тип AppiumDriver, но инициализируется AndroidDriver. Это позволяет нам абстрагироваться от конкретной реализации драйвера, для того, чтобы в будущем, если потребуется такая задача, легко добавить нужный тип драйвера во фреймворк.

Класс «TestListener» формирует оболочку драйвера, имплементируя методы интерфейса «ITestListener». Это позволяет добавить некоторые действия при определенных результатах тестов, т.е. персонализировать их.

В созданном фреймворке логирование происходит практически на всех уровнях работы, т.е. открытие и закрытие порта, сервера, создание и уничтожение драйвера, эмулятора, запуск тестов, успешное и неуспешное выполнение тестов, информация о различных видах ошибок сохраняется в логе с помощью библиотеки log4j2.

Всего было реализовано 14 классов-страниц и 11 классов с тестами.

Запуск тестов осуществляется средствами Maven. Также создан файл smoke.xml, который позволяет определять запускаемые тесты вне кода и объявляет оболочку драйвера. Данный файл добавляет гибкости во фреймворк, т.к. становится возможным легко добавлять новые оболочки драйвера, если это необходимо.

Во время проведения тестирования, благодаря разработанному фреймворку, были обнаружены дефекты в приложении. Сам фреймворк получился гибким и универсальным для разных типов Android-приложений.

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения выпускной квалификационной работы бакалавра были решены все поставленные задачи, в частности изучены основные виды, типы тестирования, систематизированы знания в области разработки

приложений для ОС Android, изучены современные методы, технологии, используемые для реализации автоматизированного тестирования мобильных приложений под ОС Android, проведен сравнительный анализ основных инструментов мобильного тестирования на Android, разработан тестовый фреймворк, проведено тестирование приложения. Таким образом, поставленная цель достигнута, а именно был разработан и реализован фреймворк для автоматизированного тестирования мобильного приложения для ОС Android.

В заключение следует сделать вывод, что самым удобным и универсальным инструментом для проведения мобильного тестирования является Appium, но также стоит сказать, что другие рассмотренные средства могут быть более подходящими под какие-либо задачи.

Таким образом, получилось создать гибкий фреймворк для мобильного тестирования. Данный фреймворк является универсальным для многих типов приложений и устройств, он дает возможность легко менять устройство, на котором запускаются тесты, позволяет определять запускаемые тестовые комплекты вне кода, создает отчетность и выполняет логирование, не допускает создание лишних драйверов, открыт для расширения.

#### **Основные источники информации:**

1. Дэвид Гриффитс, Дон Гриффитс, Head First. Программирование для Android. 2-е изд. – СПб.: Питер, 2018. – 912 с.
2. Маркус Полс, Retrofit: Love Working with APIs on Android. – Future Studio, 2016. – 80 с.
3. Филиппс Б, Стюарт К, Марсикано К. – Android. Программирование для профессионалов. – Санкт-Петербург: Питер, 2019 – 688с.
4. Куликов С.С. – Тестирование программного обеспечения. Базовый курс. – Минск: Четыре четверти, 2017 – 312 с.