**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Институт цифры**

**ОТЧЕТ**

**О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №7**

тестирование программного обеспечения

«Автоматизация Mobile»

Студента 4 курса, МОА-211 группы

**Киселева Константина Евгеньевича**

Направление 02.03.03 – «Математическое обеспечение и администрирование информационных систем»

Руководитель:

ассистент кафедры цифровых технологий

Гайдай А.М.

Работа защищена

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_2024г.

Кемерово 2024

Оглавление

[Введение 3](#_Toc185883804)

[Основная часть 4](#_Toc185883805)

[Appium Studio 4](#_Toc185883806)

[IntelliJ IDEA 5](#_Toc185883807)

[Заключение 7](#_Toc185883808)

# Введение

**Цель**

Научиться писать автоматизированные тесты мобильных приложений с использованием Appium Studio.

**Задачи**

1. Изучить работу в Appium Studio
2. Научиться писать тесты используя встроенные возможности Appium Studio
3. Научиться писать тесты на языке Java, с использованием Appium Studio

# Основная часть

## Appium Studio

В Appium Studio присутствует возможность записи тестовых сценариев. Для того, чтобы записать сценарий, нужно выбрать приложение в списке и нажать на кнопку record.

После начала записи Appium отображает экран выбранного устройства, на котором можно выполнять какие-либо действия прямо с компьютера, каждое действие записывается как команда, действия, выполненные напрямую со смартфона, не записываются.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 1 - Библиотека команд

Также, Appium предоставляет режим «Object spy», который позволяет просмотреть параметры виджетов, отображаемых на экране мобильного устройства.

## IntelliJ IDEA

Хоть Appium Studio и позволяет генерировать код, надежнее написать его самостоятельно.

Для написания тестов необходимы следующие библиотеки:

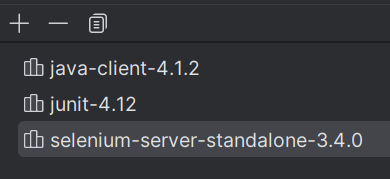


Рисунок 2 - Добавление библиотек

* «java-client» - необходима для работы с Appium;
* «junit» - необходима для использования тегов @Before, @After, @Test;
* «selenium» - необходима для работы с xPath и элементами на экране.

В ходе работы лабораторной работы мы написали следующие универсальные методы:  
Изображение выглядит как текст, Шрифт, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, Шрифт, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Они позволяют нам находить любые элементы на экране и взаимодействовать с ними.

Далее эти методы были помещены в класс, представляющий собой объект для работы с какой-либо узкой функциональностью (MainPageObject). Затем были определены производные от этого класса, в которых конкретизировались объекты взаимодействия:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Пример одного из таких классов:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Данный класс предназначен для взаимодействия с поиском по википедии, что определено полями.

# Заключение

Итак, IntelliJ IDEA и Appium Studio работают в паре: в Appium Studio в режиме «Object spy» мы просматриваем интересующие нас элементы (можем сразу скопировать xPath), а в IntelliJ IDEA прописываем алгоритм действий, который мы хотим совершить в ходе теста.