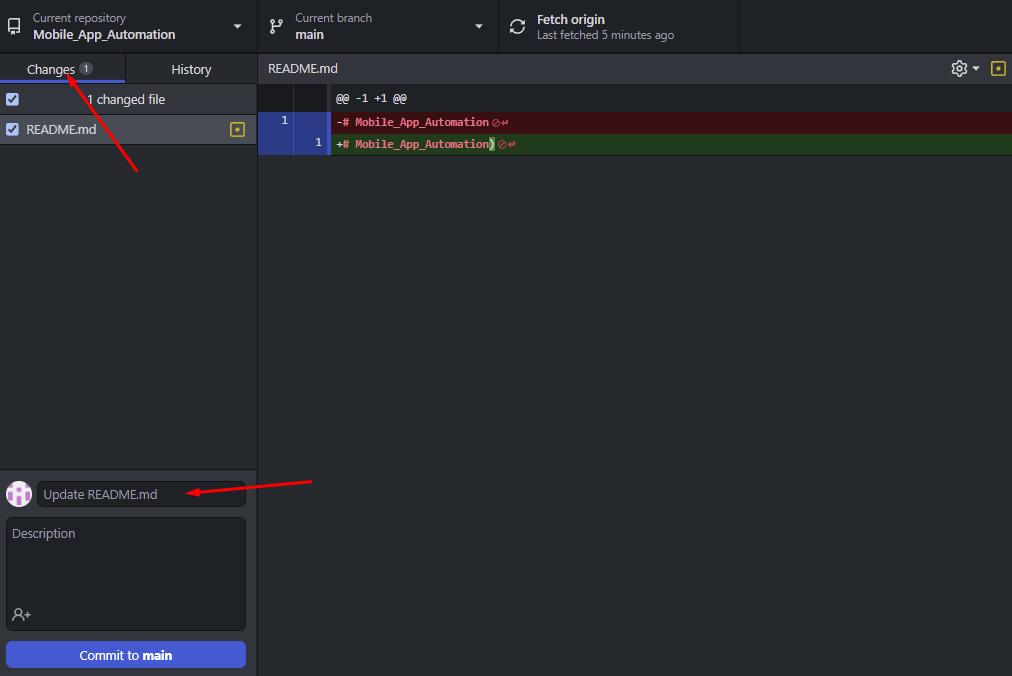
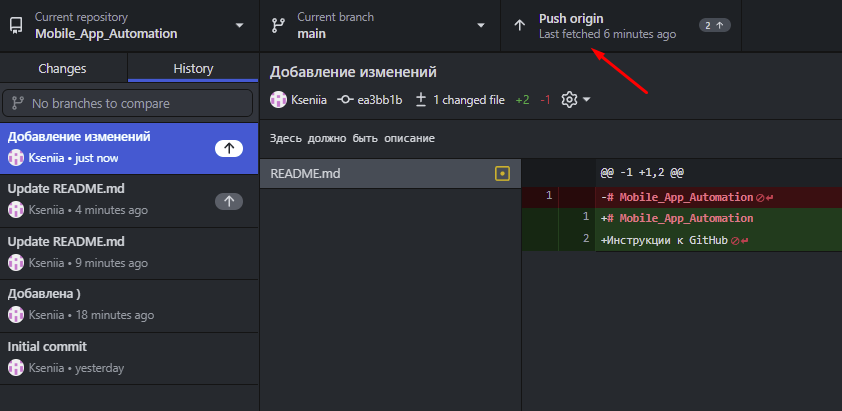
# Урок 0

## Инструкции к GitHub

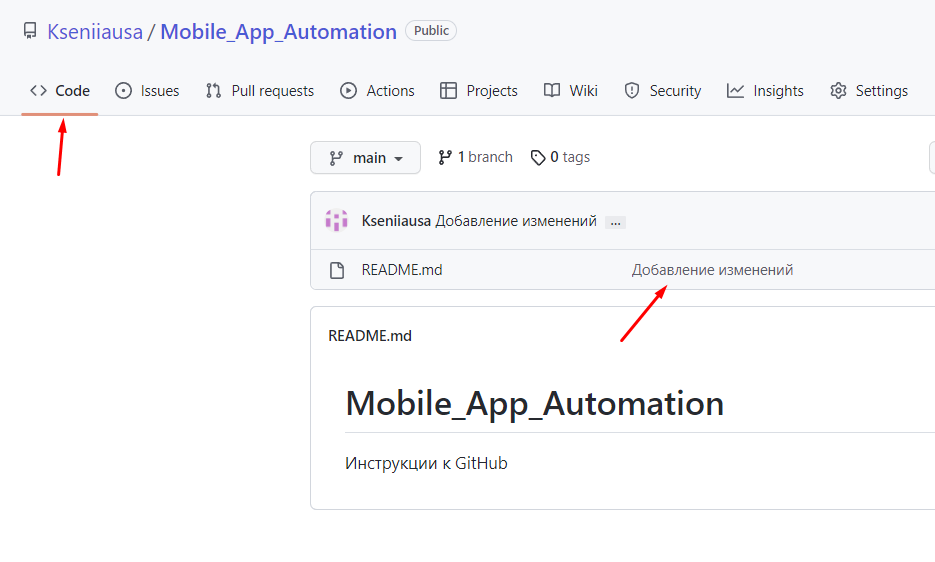
1. При внесении изменений можно посмотреть изменения и добавить комментарий:



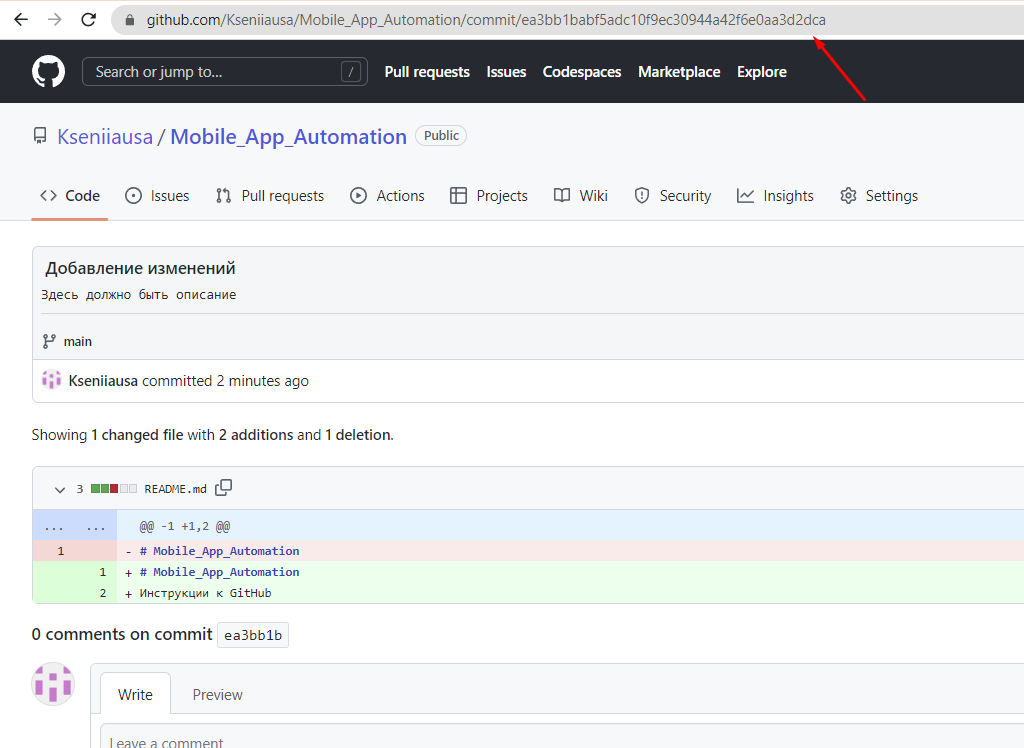
1. Добавляем комментарий и жмем кнопку коммит
2. Изменения сохранены локально, просмотр вкладка History
3. Для отправки на удаленную машину нажать PUSH – двухсторонняя синхронизация.



1. Изменения отображаются на удаленном доступе

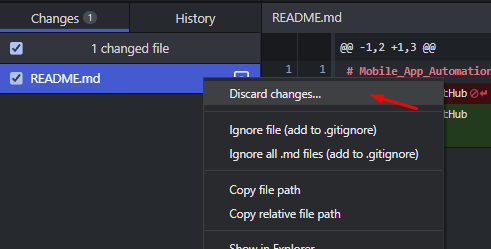


1. При переходе по коммиту открываются изменения, ссылку на которых необходимо отправлять на проверку



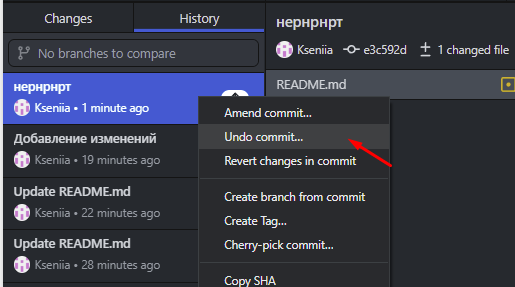
### Вернуть изменения

1. Внесли изменения в файл и сохранить
2. Перейти в ГитД и выбрать:



### Вернуть изменения с коммитом

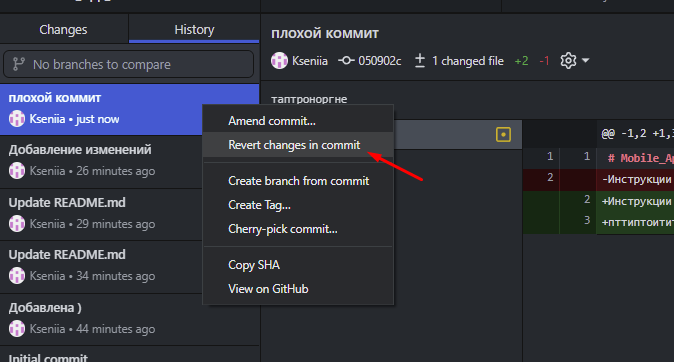
1. Внесли изменения в файл и сохранить
2. Отправить коммит
3. Перейти в ГитД и выбрать:



1. Для отмены изменений выполнить шаги [выше](#_Вернуть_изменения)

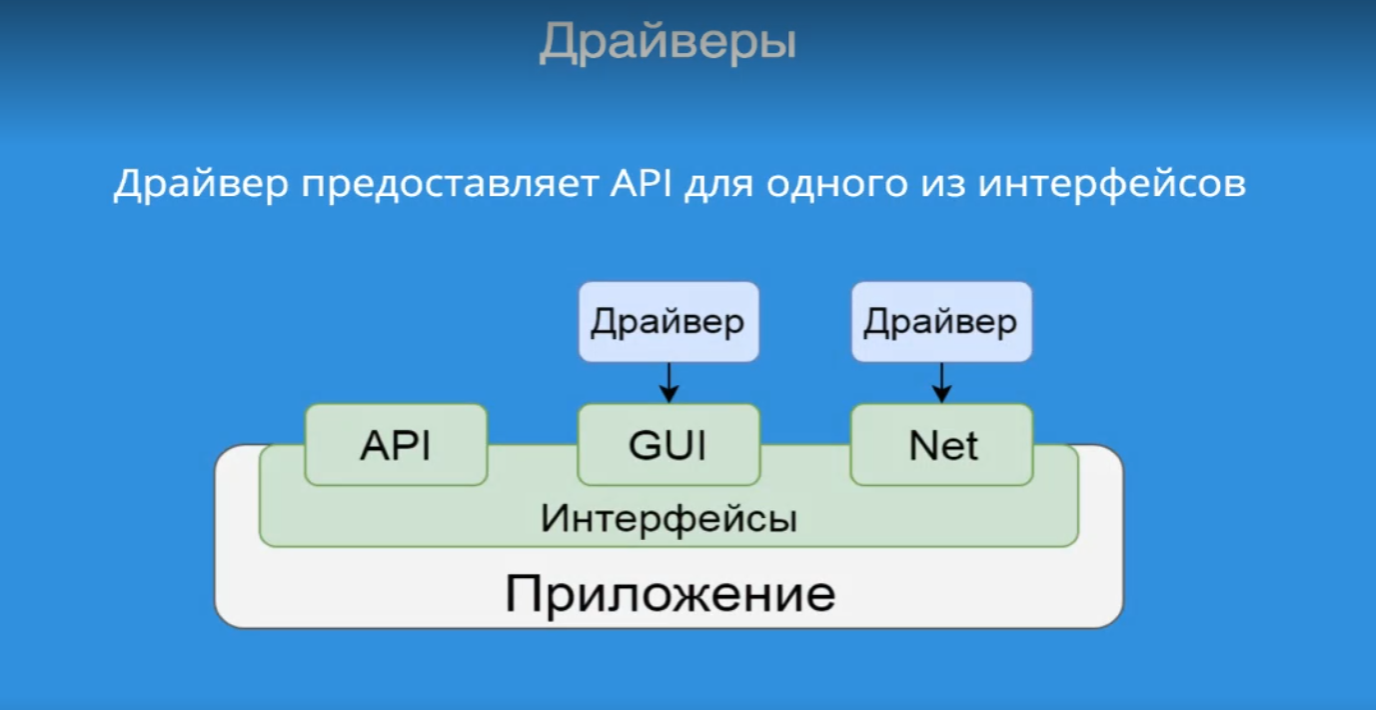
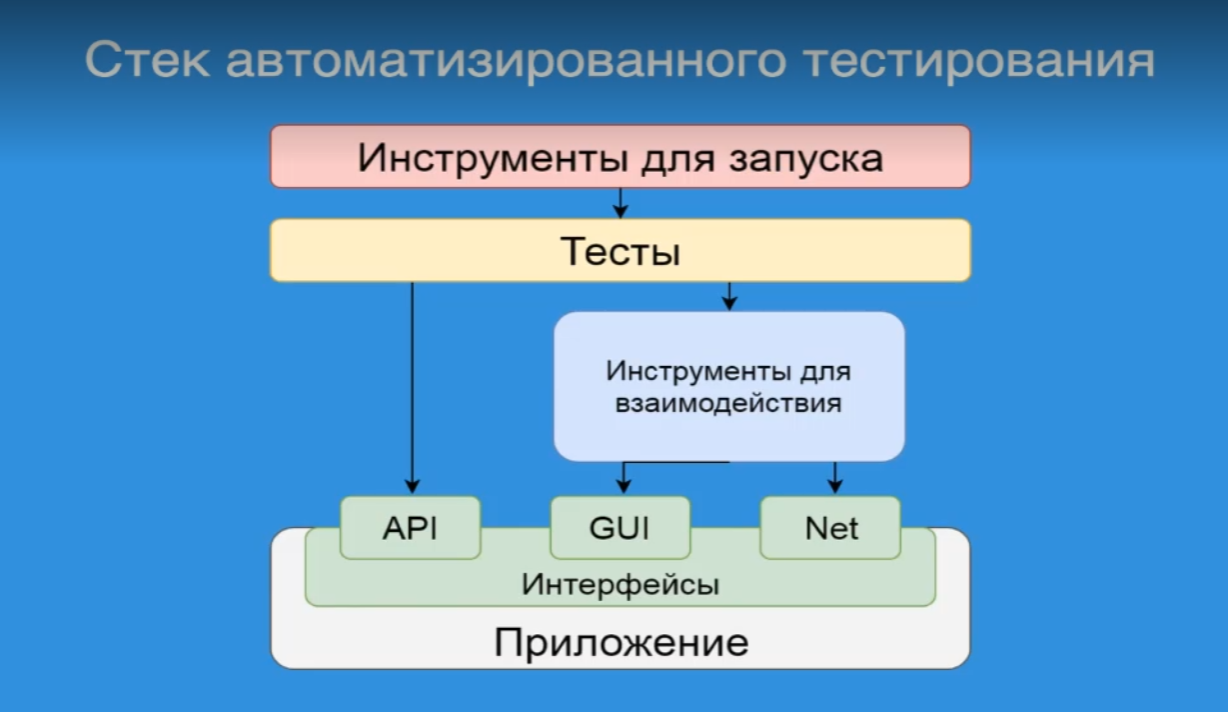
### Вернуть изменения с коммитом и уд. синхронизацией

1. Внесли изменения в файл и сохранить
2. Отправить коммит
3. Выполнить синхронизацию
4. Перейти в ГитД и выбрать Revert – локально изменения отменены
5. Для отправки на удаленную машину нажать PUSH

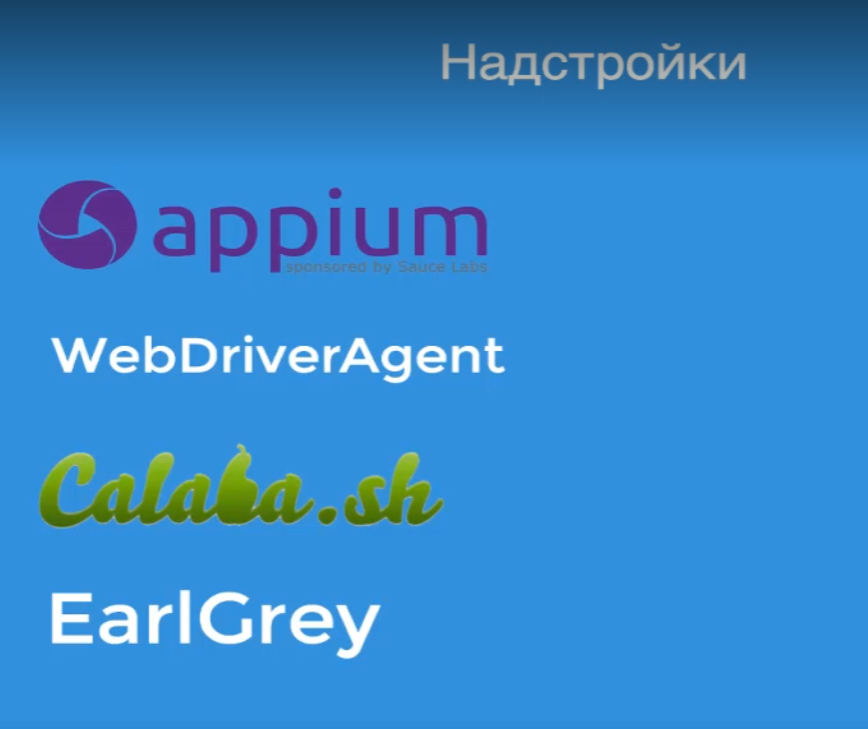
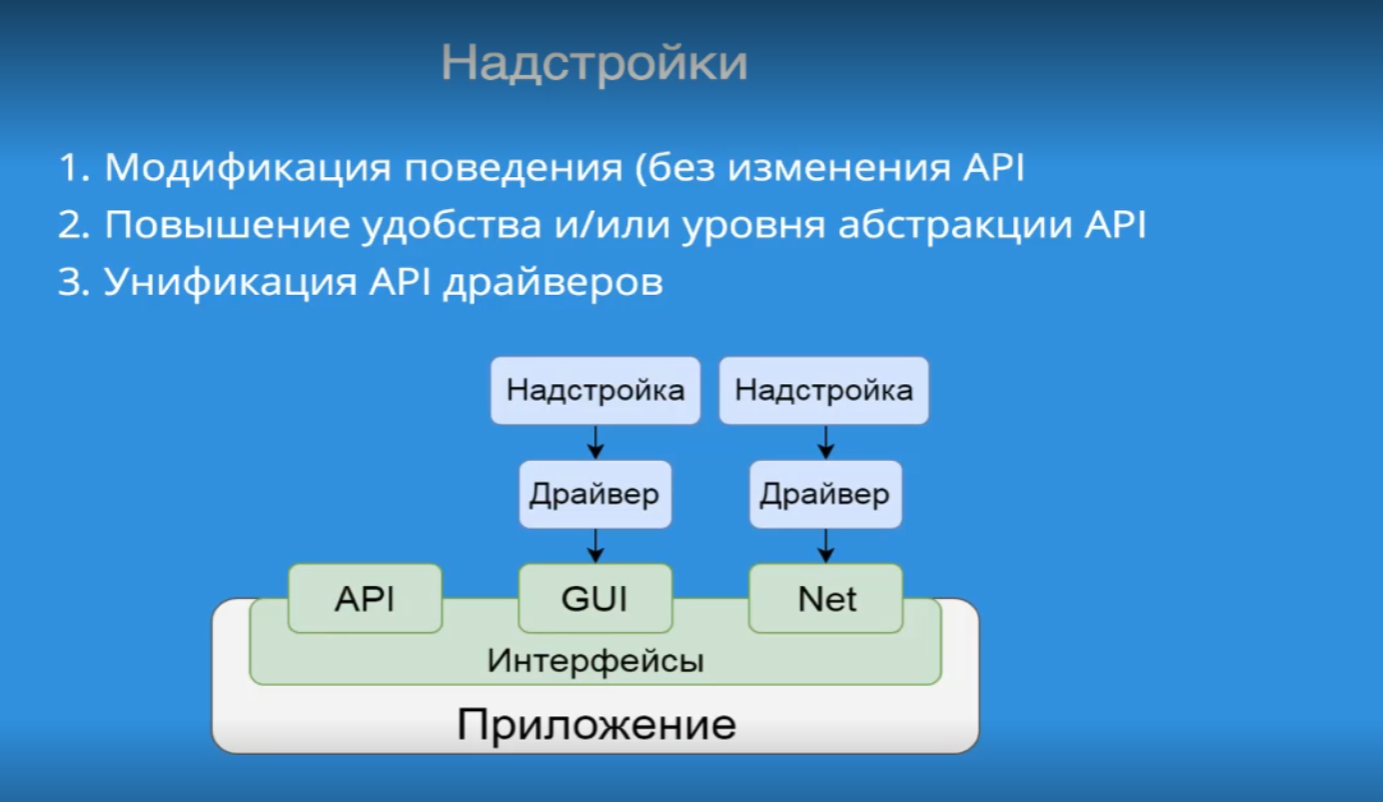


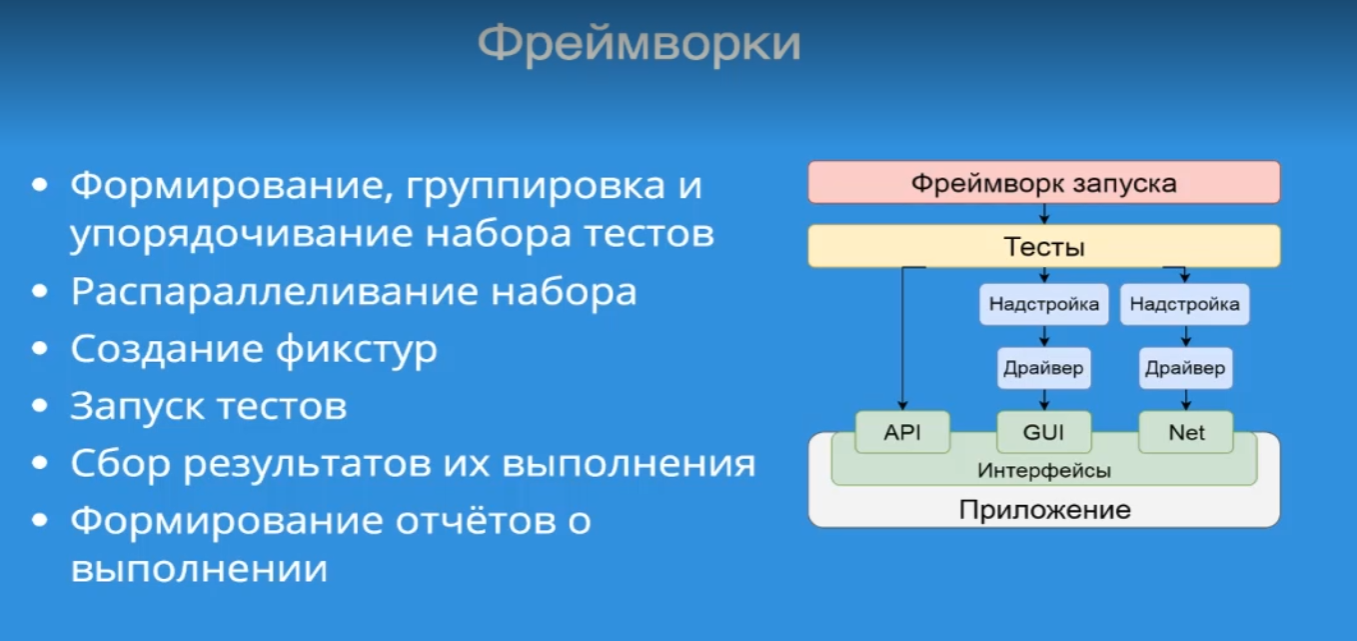
# Урок 1

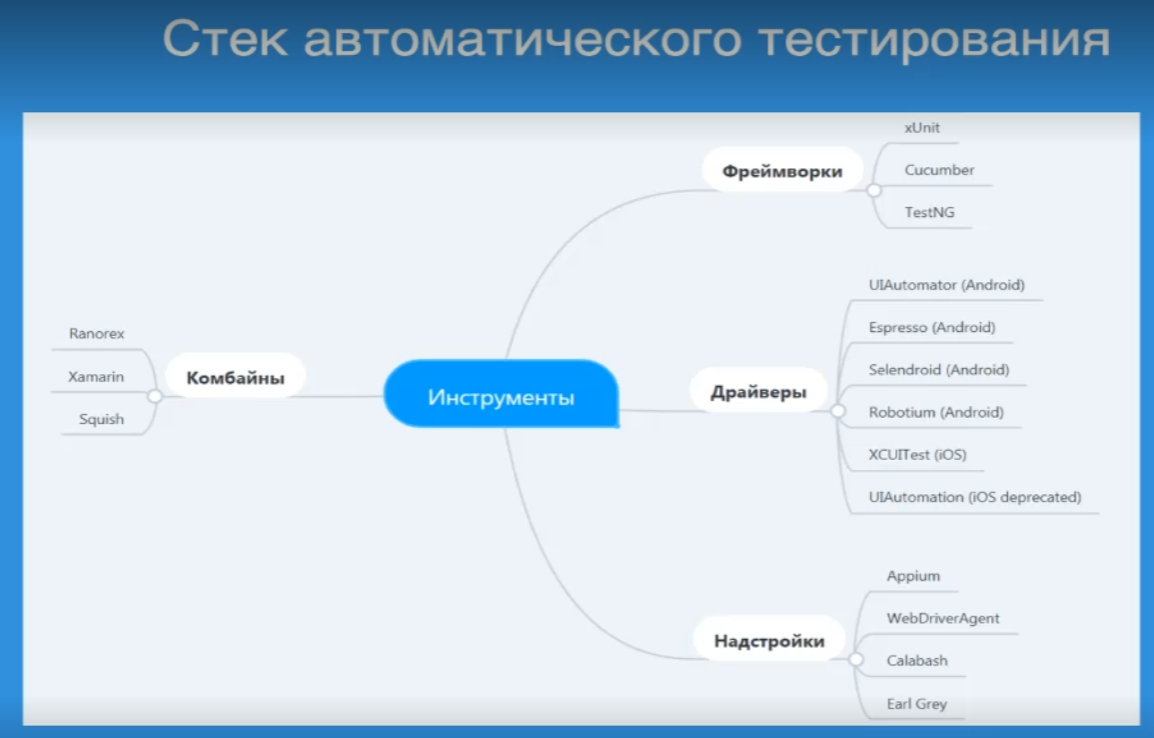
## Инструменты



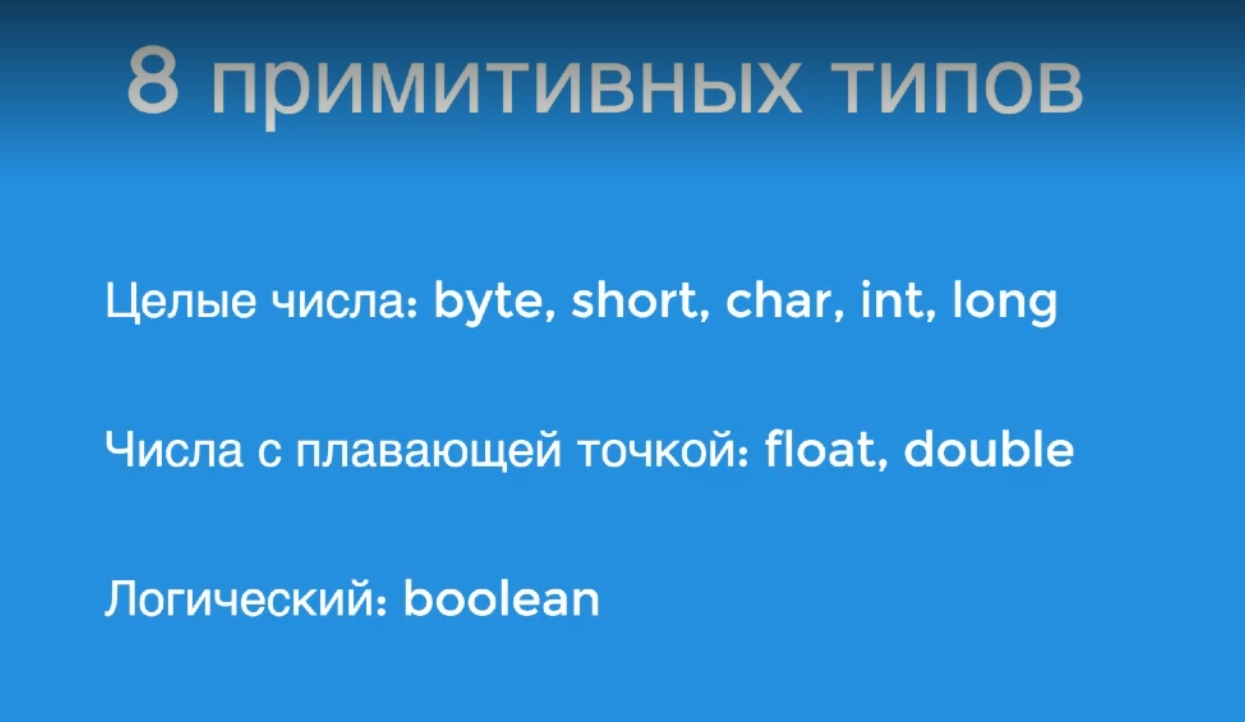




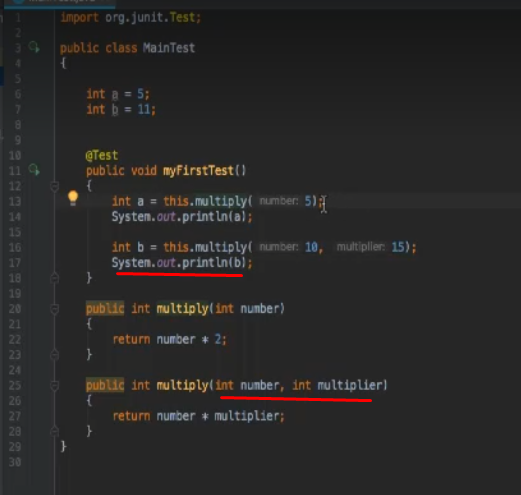
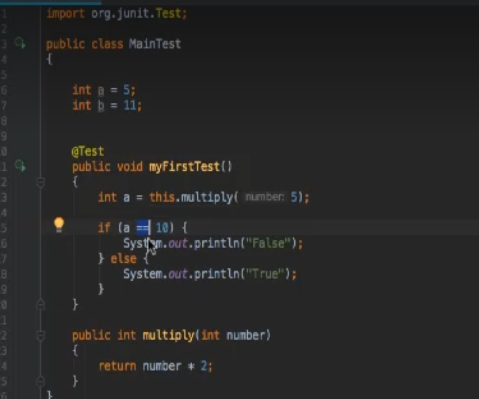




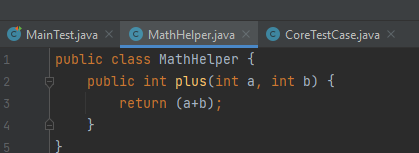
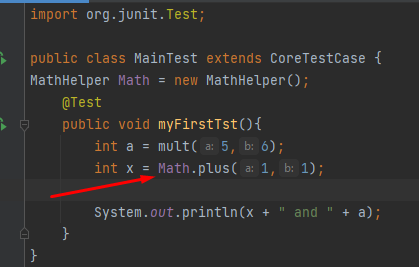
## Java



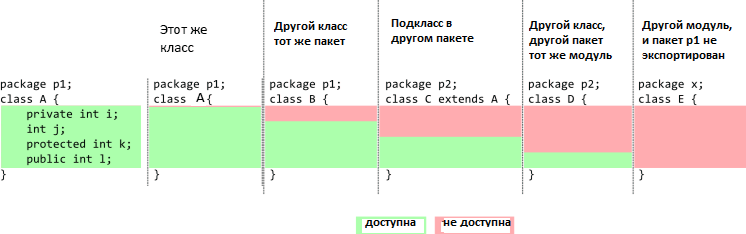
### Функции и this



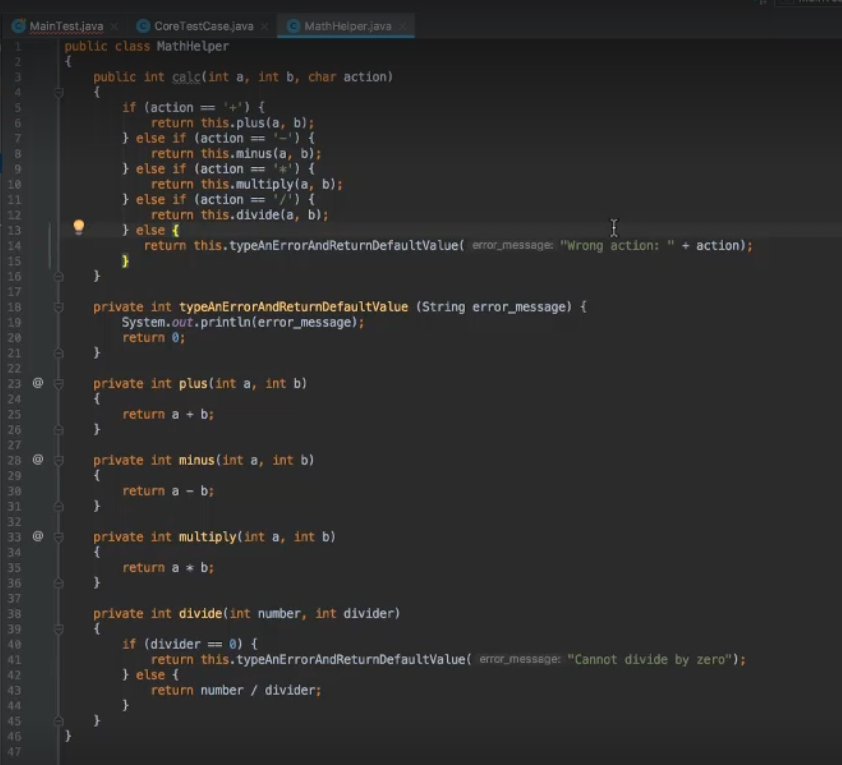
### Родительские классы



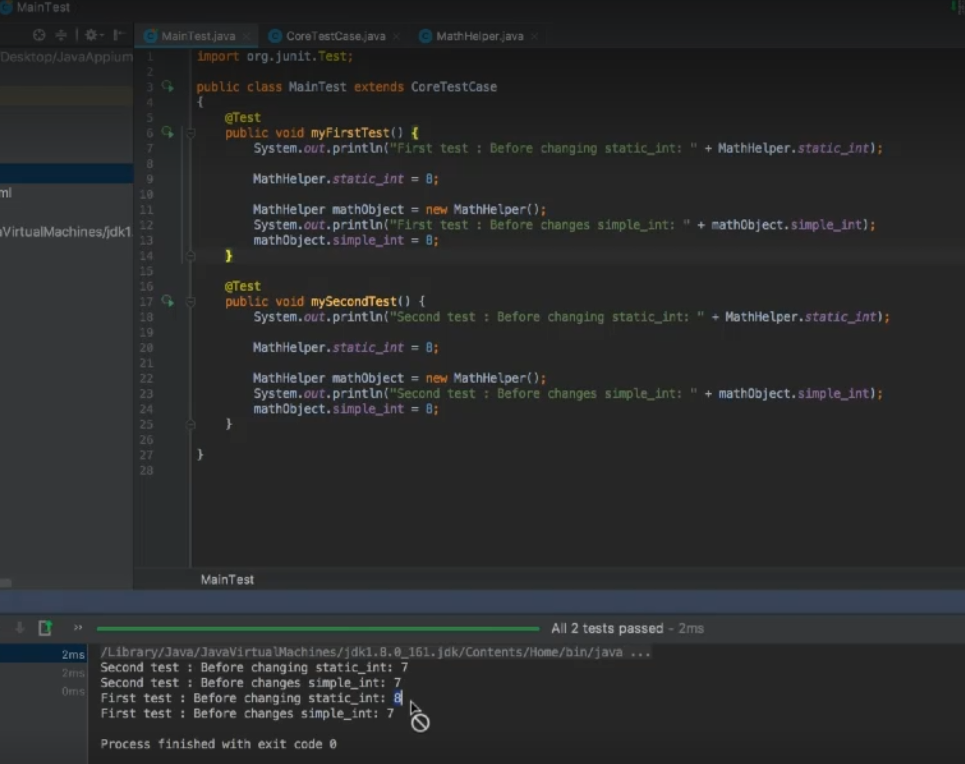
### Модификатор доступа



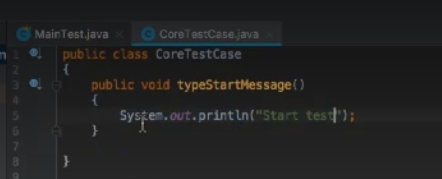
Пример:



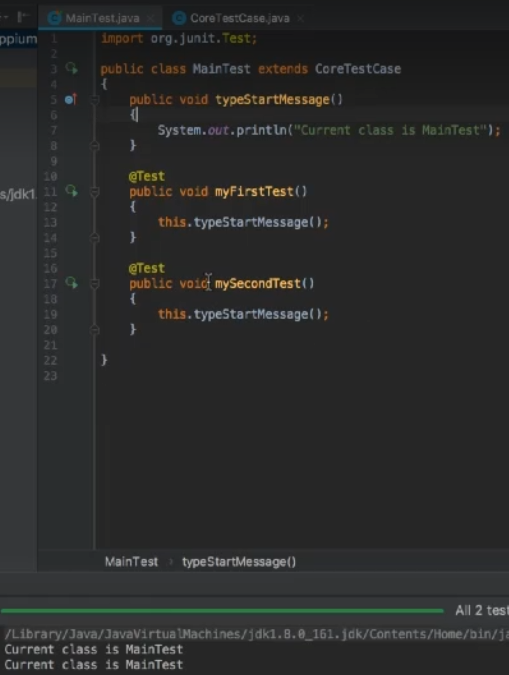
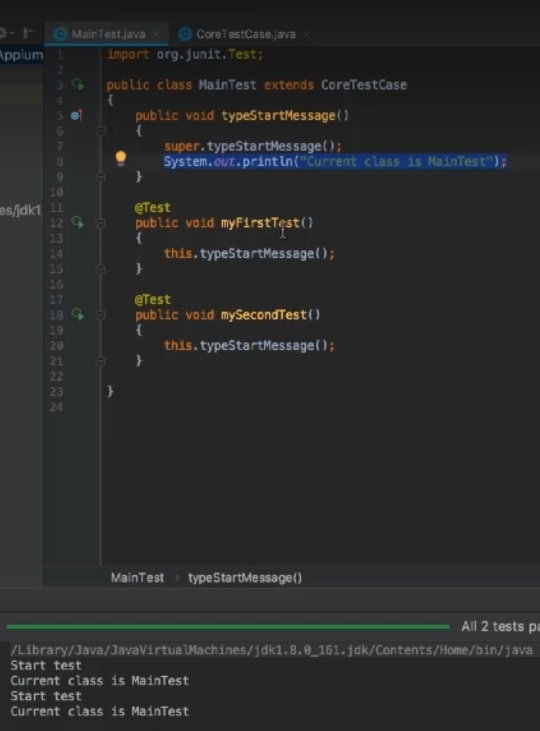
### Статические методы и поля?



### Super



При Super сначала родитель потом текущий. Для переопределения – убрать.



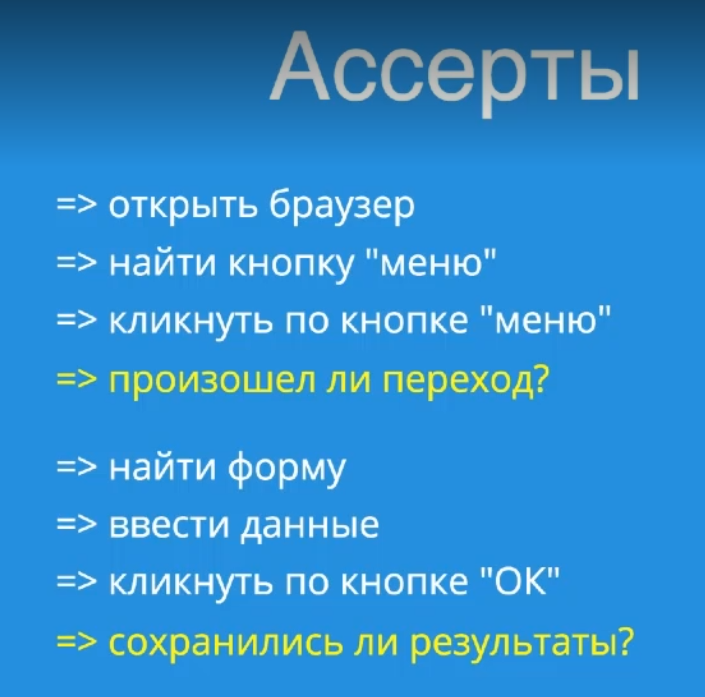
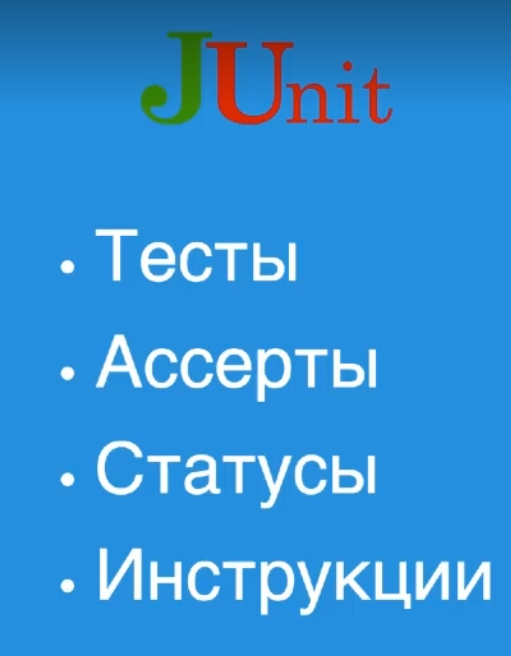
### ООП



### Junit

Ассерт - соответствие значения параметру предустановленного значения

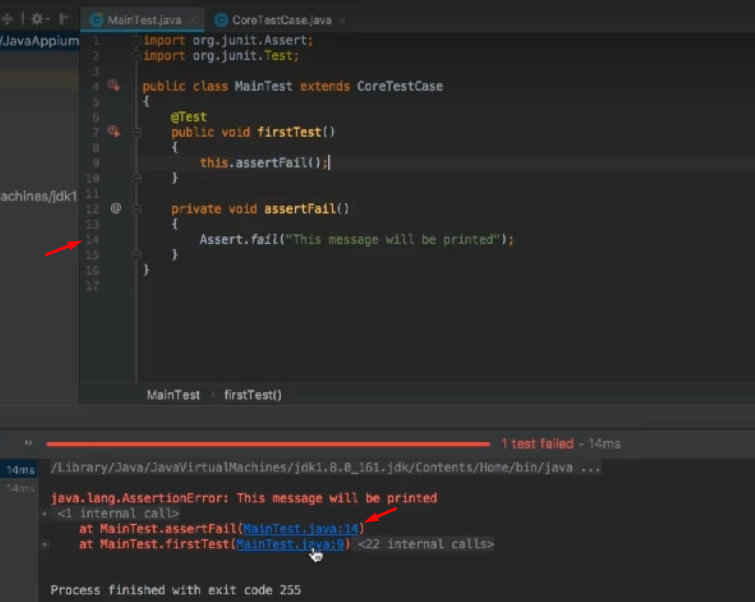
Инструкции – before, after



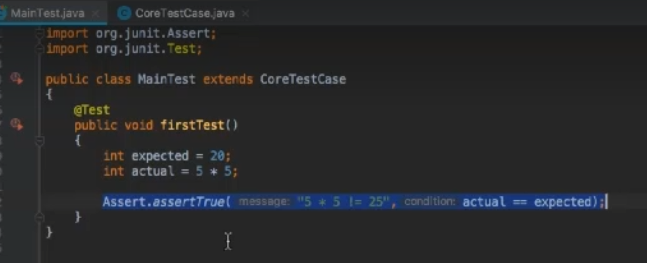
### Test

Alt(option) + enter – выбор

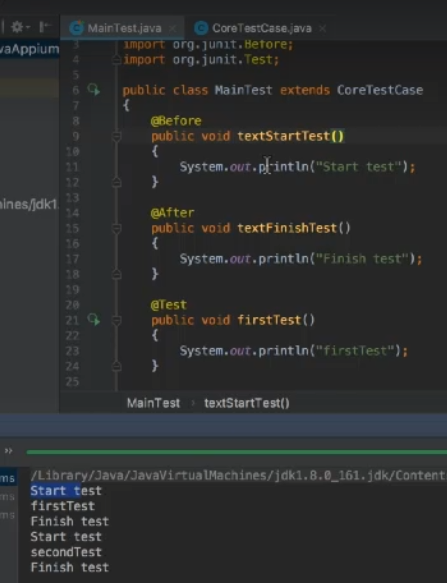
Trace показывает где ошибка



Пример успешного теста (справа как нужно писать):



Before – перед каждым тестом, After - после каждого теста:



# Урок 2

## Запуск эмулятор из Cmd

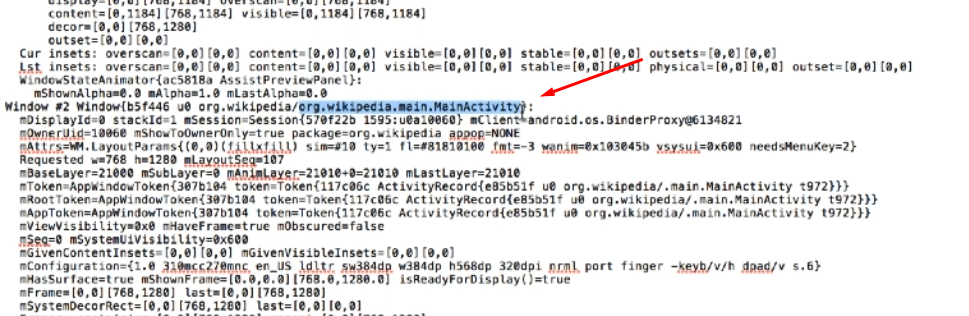
emulator @and80

## Packages and Capabilities

1. Запустить приложение
2. Проверить подключение : adb devices
3. Список всех пакетов всех приложений на устройстве: adb shell pm list packages
4. Найти пакет: adb shell pm list packages | grep "wiki"
5. Выдает активность на открытом экране приложения и записывает в файл:

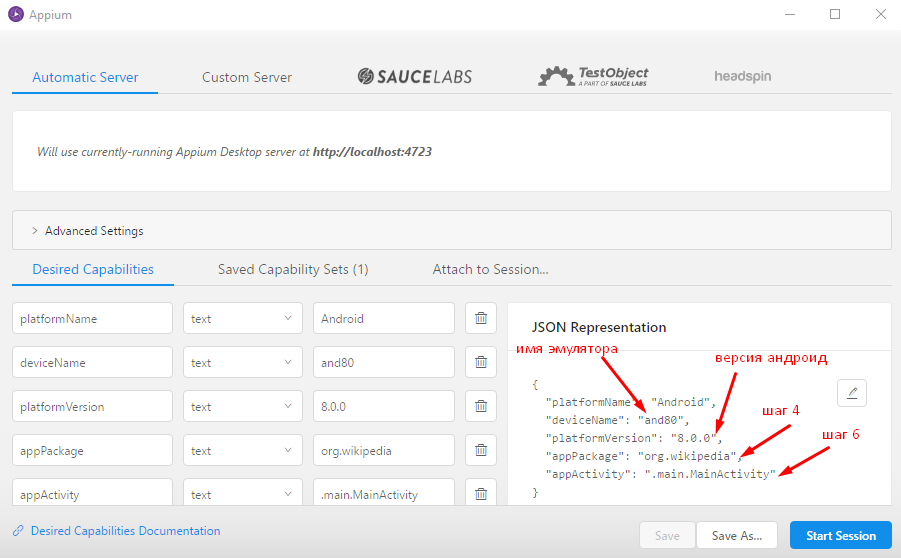
adb shell dumpsys window windows >> Desktop\activity.txt

1. Найти в файле по строке – шаг 4.



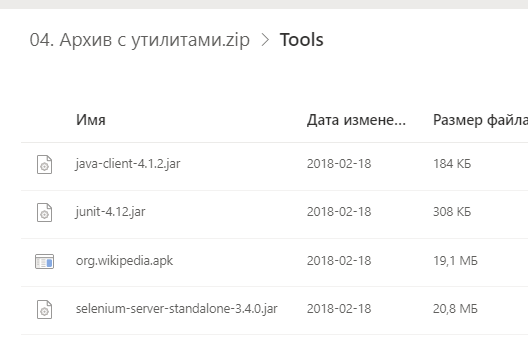
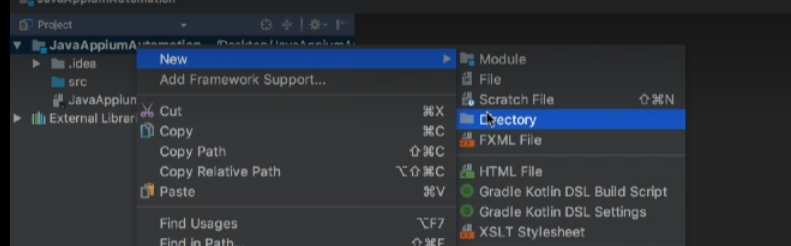
1. Перенести параметры в Appium

## Appium

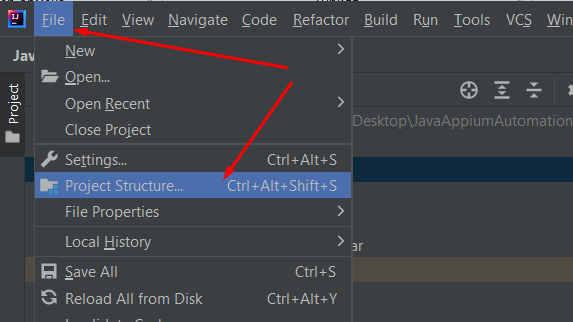
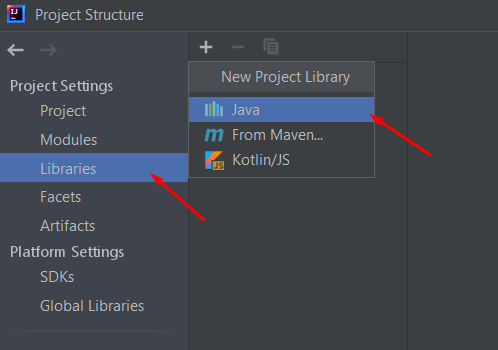


## Создание нового проекта

1. Скачать и импорт в проект в новую папку libs (фреймворки ниже)

1. Перейти в структуру и добавить их

1. Создать еще папку apks – перенести приложение

## Первый тест

В классе FirstTest

import io.appium.java\_client.AppiumDriver;  
import io.appium.java\_client.android.AndroidDriver;  
import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;  
import org.openqa.selenium.remote.DesiredCapabilities;  
  
import java.net.URL;  
  
public class FirstTest {  
 private AppiumDriver driver;  
 @Before  
 public void setUp() throws Exception {  
 DesiredCapabilities capabilities = new DesiredCapabilities();  
 capabilities.setCapability("platformName","Android");  
 capabilities.setCapability("deviceName","and80");  
 capabilities.setCapability("platformVersion","8.0.0");  
 capabilities.setCapability("automationName","Appium");  
 capabilities.setCapability("appPackage","org.wikipedia");  
 capabilities.setCapability("appActivity",".main.MainActivity");  
 capabilities.setCapability("app","C:\\Users\\User\\Desktop\\JavaAppiumAutomation\\apks\\org.wikipedia.apk");  
  
 driver = new AndroidDriver(new URL("http://127.0.0.1:4723/wd/hub"),capabilities);  
 }  
  
 @After  
 public void tearDown ()  
 {  
 driver.quit();  
 }  
 @Test  
 public void firstTest(){  
 System.*out*.println("First run");  
 }  
}