# 1. Язык программирования Java

**Java** – строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle).

Программы на Java транслируются в байт-код Java, выполняемый виртуальной машиной Java (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор.

Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина.

Другой важной особенностью технологии Java является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером), вызывают немедленное прерывание.

Почему именно Java?

Прочитайте статью про выбор языка программирования для автотестов <https://www.software-testing.ru/library/testing/testing-automation/2308-automation-language-is-not-necessarily-your-development-language>

## 1.1. Где взять Java?

На сайт Oracle <https://www.oracle.com/java/technologies/javase-downloads.html>

## 1.2. Какую версию Java устанавливать?

Релизы Java выходят раз в полгода (не считая мелких обновлений). Но не стоит гнаться за новизной. Если вы установите Java последней версии, приготовьтесь к тому, что не все приложения будут хорошо работать с ней. Новые недавно добавленные возможности иногда приводят к проблемам совместимости.

В сентябре 2019 вышла версия 13, в 2020 вышла версия 15, но при этом большинство продолжает использовать Java 8! Поэтому если вы хотите максимальной стабильности и совместимости — в 2020 году берите Java 8.

Второе место по популярности занимает Java 11, это так называемый релиз с долгосрочной поддержкой (Long Term Support, LTS), ориентированный на корпоративных пользователей, для которых стабильность важнее новых фич.

Поддержка Java 8 официально прекращается в декабре 2020 года, к этому времени все корпоративные пользователи будут вынуждены перейти на Java 11. Но вы уже сейчас, не дожидаясь конца 2020 года, можете смело брать эту версию, она обеспечит комфортное соотношение достаточной новизны и не очень высокого риска. А если что-то не будет работать — можно установить рядом Java 8 для использования со старыми приложениями.

Выбирая из 32-битной и 64-битной версий, берите 64-битную, если ваша операционная система это позволяет.

## 1.3. Что устанавливать, JRE или JDK?

**Java Development Kit, или JDK** — это набор инструментов для разработки программ на языке программирования Java (компилятор, архиватор, генератор документации и прочие), среду выполнения (Java Runtime Environment), стандартную библиотеку языка, примеры и документацию.

**Java Runtime Environment, или JRE** — это виртуальная машина, позволяющая запускать приложения, написанные на языке программирования Java.

Правило очень простое: если вы собираетесь что-нибудь писать на языке программирования Java, значит вам потребуется JDK. А если только запускать готовые программы — тогда достаточно JRE.

## 1.4. Как устанавливать Java?

### 1.4.1. Установка Java Development Kit

Шаги:

1. Перейти на сайт Oracle <https://www.oracle.com/java/technologies/javase-downloads.html> и загрузите JDK для вашей платформы.
2. Запустить exe файл.
3. В процессе установки выбрать «Development Tool» и нажать «Next».

Спустя некоторое время установка завершится.

Итак, Java Development Kit установлен, но это еще не всё. Необходимо настроить его для системы.

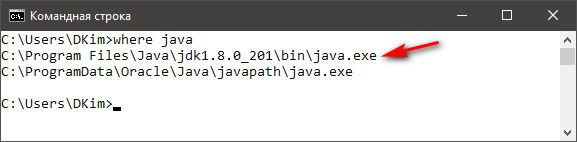
### 1.4.2. Настройка переменных сред

Инсталлятор Java выполняет минимальную необходимую настройку окружения, в том числе он добавляет в переменную среды PATH путь к директории, которая содержит исполняемые файлы Java (в операционной системе Windows), либо создаёт в стандартной директории для исполняемых файлов символические ссылки на установленные исполняемые файлы Java (в других операционных системах).

Но некоторые программы вместо этого используют переменную среды JAVA\_HOME, которая должна указывать на директорию, в которую установлена Java. Поэтому на всякий случай можно сразу установить эту переменную.

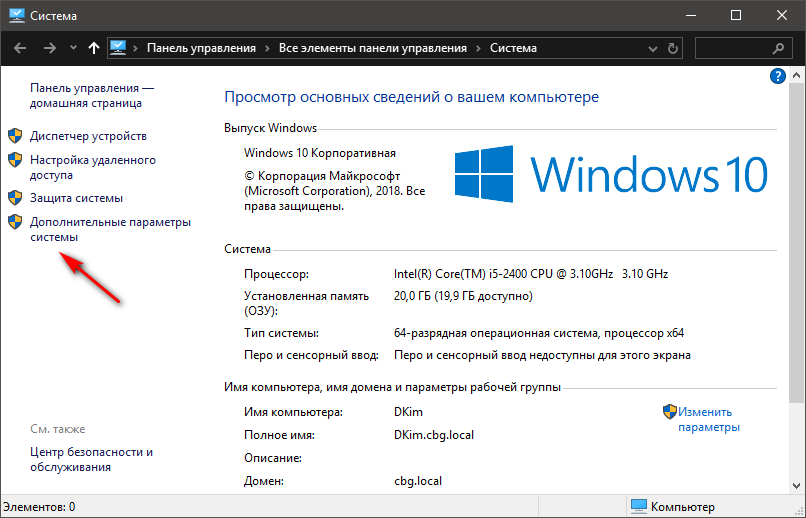
Шаги:

1. Выполнить команду where java в консоли

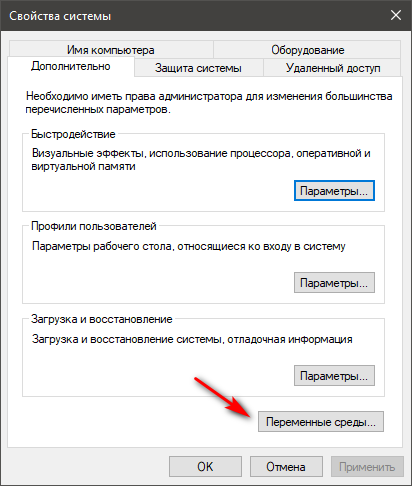


Эта команда вернёт путь к директории установки Java

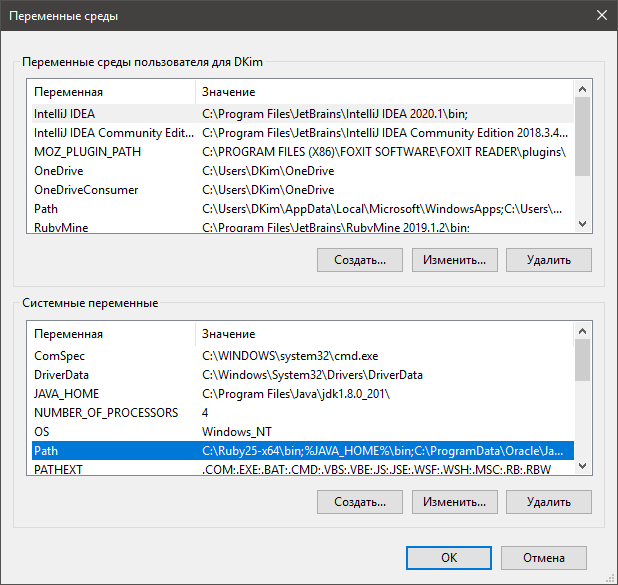
1. Скопировать пути к директории установки Java
2. Открыть Панель управления -> Система, выбрать Дополнительные параметры системы



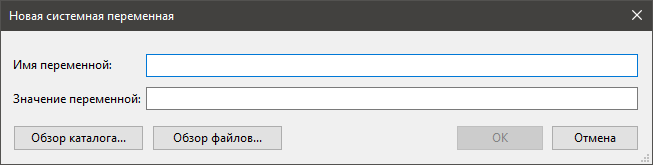
1. В появившемся окне нажать кнопку Переменные среды.



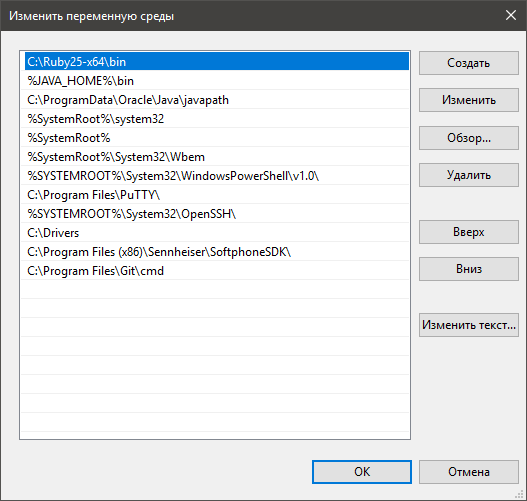
1. В открывшемся окне в блоке Системные переменные нужно проверить существует ли переменная PATH



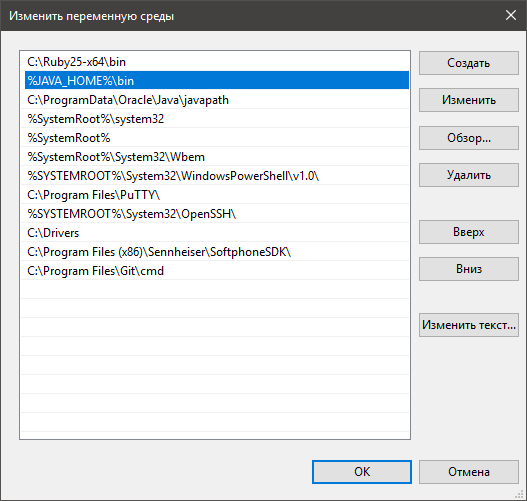
1. Если переменная PATH не существует, то ее нужно создать, нажав на кнопку Создать



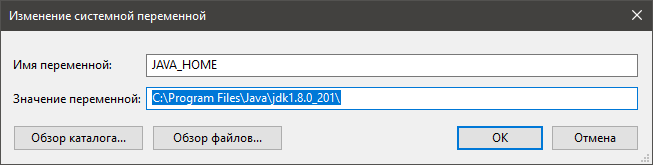
1. Если переменная PATH уже существует, то ее нужно изменить, нажав на кнопку Изменить



1. Для переменной PATH нужно установить Значение переменной — скопированный ранее путь к каталогу (%JAVA\_HOME%\bin), в который была установлена Java (из шага 1)



1. Подобным образом создается/изменяется переменная JAVA\_HOME. Значение этой переменной — путь к каталогу, в который была установлена Java (как для PATH), но уже без поддиректории bin



То есть будет примерно так

* Значение в **PATH**: C:\Program Files\Java\jdk1.7.0\_25\bin
* Значение в **JAVA\_HOME**: C:\Program Files\Java\jdk1.7.0\_25

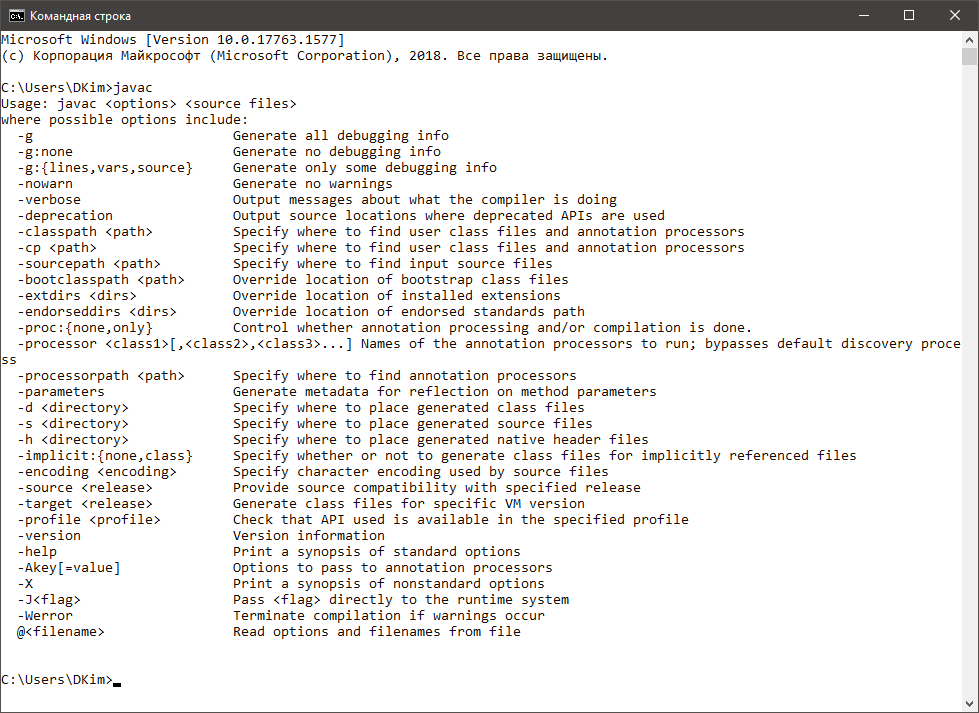
После того, как были прописаны переменные, установку Java можно считать завершенной!

## 1.5. Как проверить правильность установки?

Теперь самое важное – необходимо убедиться, что Java установлена и настроена корректно.

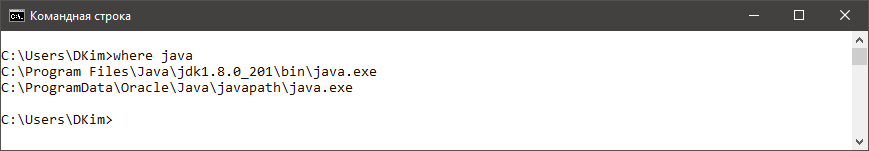
Шаги:

1. Открыть командную строку
2. Выполнить команду javac, чтобы убедиться, что JRE установлен и настроен.

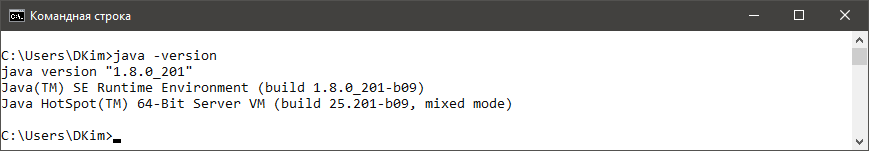


Если выдаётся список аргументов к команде javac, то успешно установлено и настроено все, что необходимо для использования Java!

1. Выполнить команду where java, которая должна показать правильный путь до исполняемого файла java.exe



1. Выполнить команду java -version для проверки того, что это именно та версия, которую устанавливали:

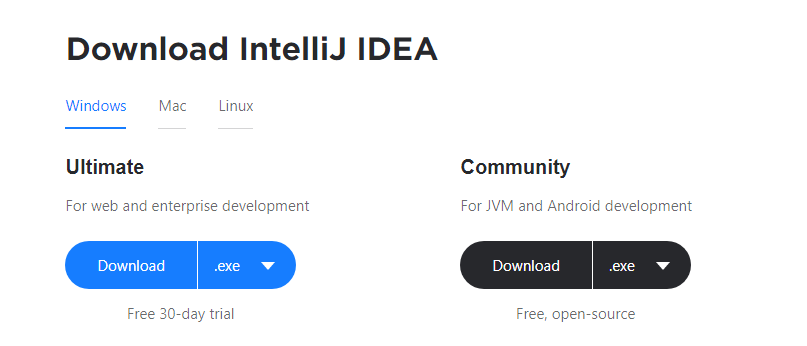


Ссылка: <http://barancev.github.io/how-to-install-java/>

# 2. Среда разработки IntelliJ

**IntelliJ IDEA** — известная IDE для Java, написанная, как ни странно, на Java. Укомплектована уникальными инструментами и позволяет без проблем ориентироваться в программе. Нахождение ошибок и отладка кода никогда не были такими легкими, как с IntelliJ IDEA.

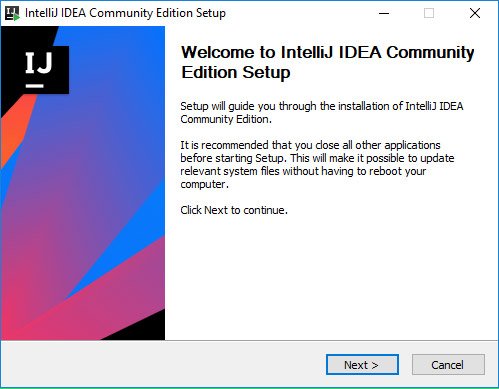
Существует две версии IntelliJ IDEA – Community и Ultimate. Первая – бесплатная, но ограниченная. В ней не поддерживаются фреймворки Spring, Vaadin, GWT, языки JavaScript и TypeScript, SQL и многое другое. Вторая включает в себя всевозможные фичи для коллективной разработки, но бесплатна будет только 30 дней пробного периода. Потом вам придётся заплатить минимум 500 долларов США



## 2.1. Установка IntelliJ IDEA

Шаги:

1. Скачать исходник с расширением exe. с официального сайта <https://www.jetbrains.com/idea/download/#section=windows>
2. Запустить его и следовать инструкциям установщика



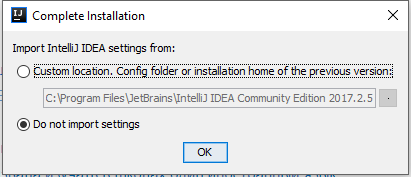
## 2.2. Предварительная настройка IntelliJ IDEA

Шаги:

1. Указать путь до настроек

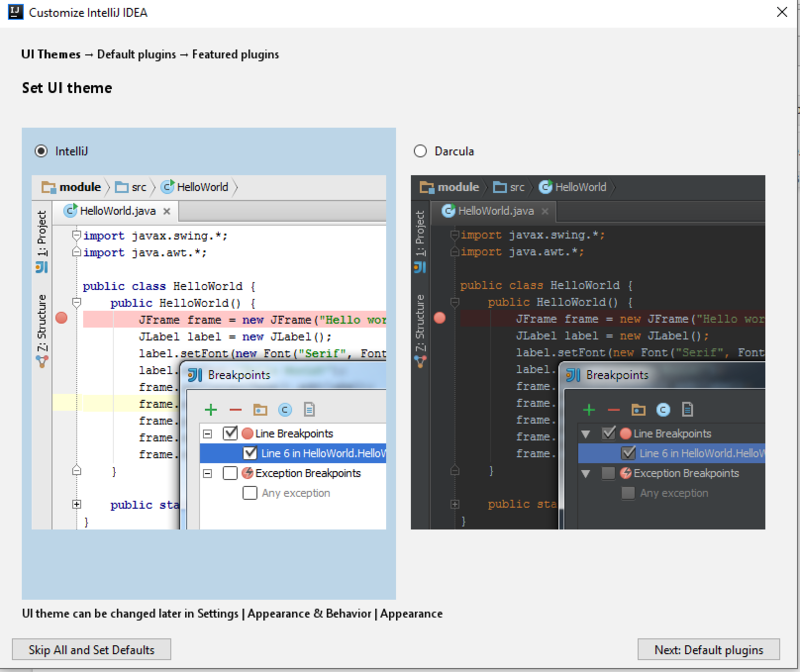
При первом запуске IntelliJ IDEA перед вами выскочит диалоговое окно с требованием указать путь до файла с настройками. Так как это первый опыт знакомства с IDE, то выбрать пункт «Do not import settings».

Если данное окно вылезло после обновления или переустановки – выбрать исходную директорию.



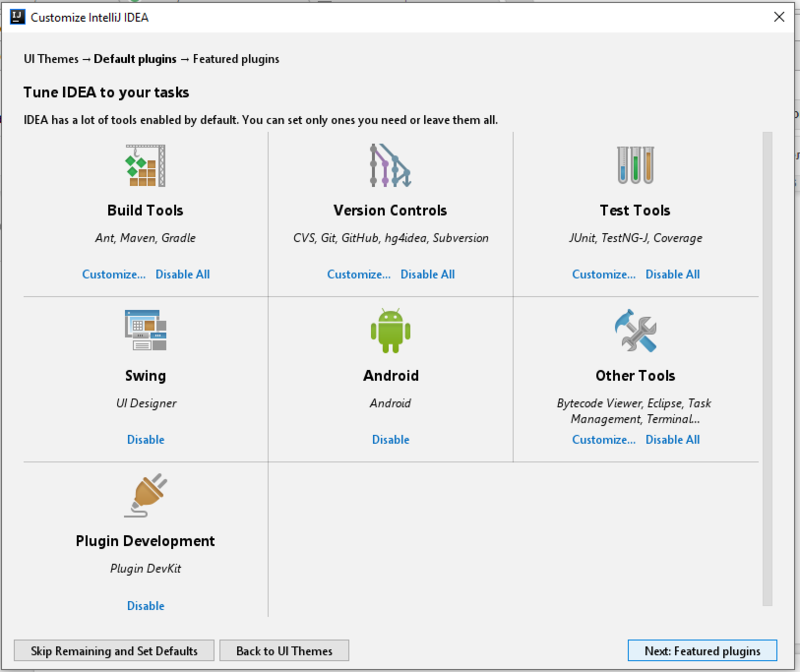
1. Выбрать тему

Следующим шагом настройки IntelliJ IDEA будет выбор темы. ПО умолчанию их две – стандартная (светлая) и Darcula (тёмная).



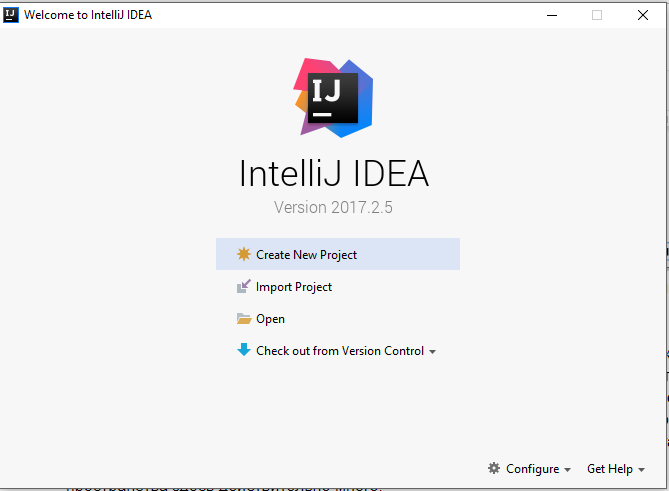
1. Отключить вспомогательные инструменты

Как уже было написано выше, IDEA поддерживает огромное количество вспомогательных инструментов. Разумеется, новичку они все не нужны. Поэтому следующим шагом будет их отключение для увеличения производительности. Для этого в очередном диалоговом окне выбрать «Disable All». Не переживайте, их можно подключить позднее в меню настроек.



1. Завершение установки и настройки

Теперь, когда с установкой и настройкой IntelliJ IDEA покончено, можно приступить к работе. Начинается она, разумеется, с создания IntelliJ IDEA проекта.



Ссылка <https://www.surgebook.com/JunMidSen/book/put-java-programmista/ustanovka-intellij-idea>

# 3. Сборщик Maven

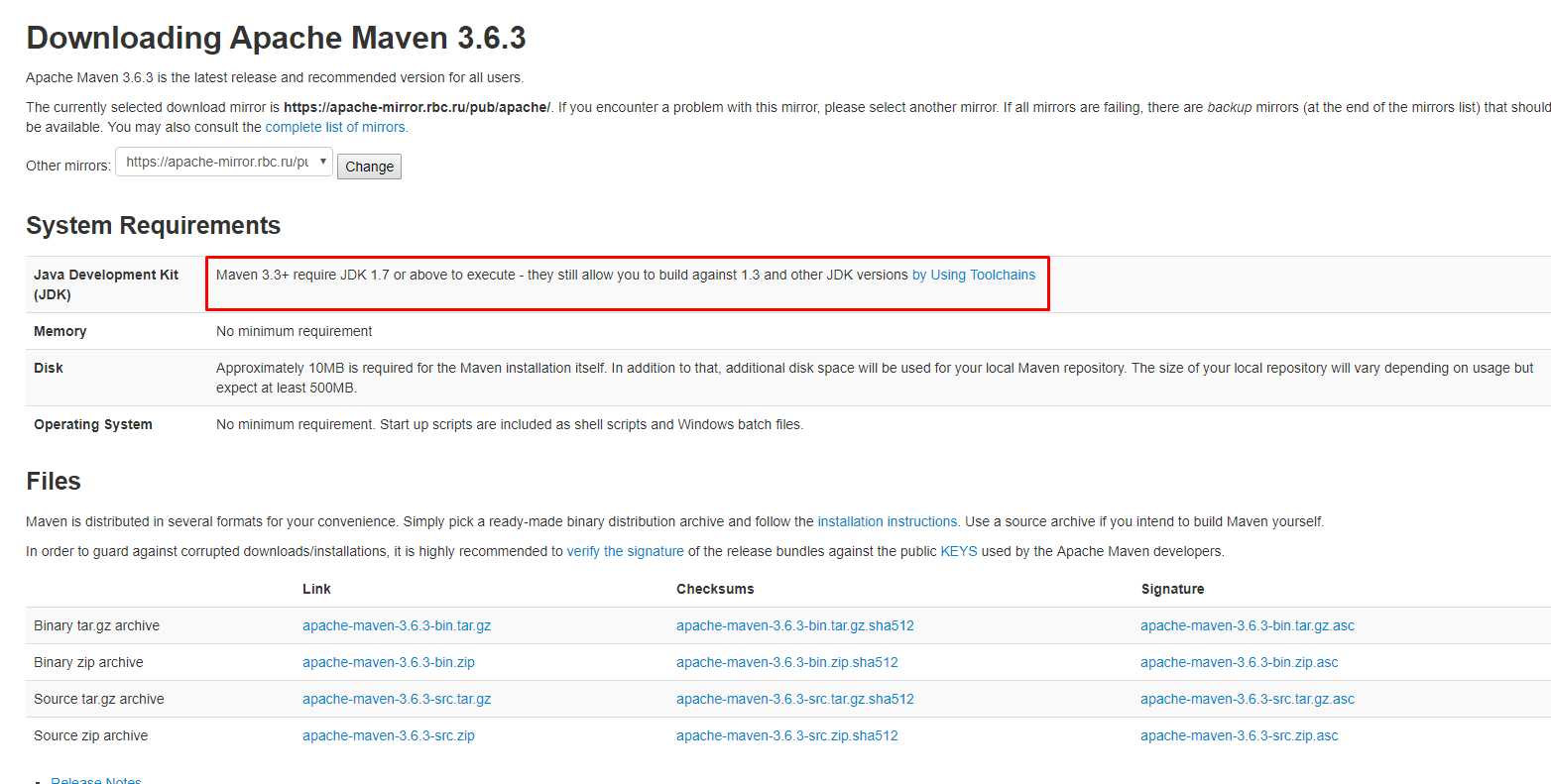
**Apache Maven** — система управления жизненным циклом проекта, фреймворк для автоматизации сборки проектов на основе описания их структуры в файлах на языке POM (англ. Project Object Model), являющемся подмножеством XML. Проект Maven издаётся сообществом Apache Software Foundation, где формально является частью Jakarta Project.

Maven обеспечивает декларативную, а не императивную (в отличие от средства автоматизации сборки Apache Ant) сборку проекта. В файлах описания проекта содержится его спецификация, а не отдельные команды выполнения. Все задачи по обработке файлов, описанные в спецификации, Maven выполняет посредством их обработки последовательностью встроенных и внешних плагинов.

## 3.1. Установка Maven

Шаги:

1. Установить Java
2. Скачать архив с официального сайта <https://maven.apache.org/download.cgi>



1. Распаковать скачанный архив в папку
2. Установить переменной окружения M2\_HOME ссылку на местонахождение распакованного архива
3. Установить еще одну переменную M2 со значением %M2\_HOME%\bin
4. Установить или добавить в переменную Path значение %M2%, чтобы наша папочка с исполняемым файлом Maven была видна из командной строки

## 3.2. Проверка установки

Шаги:

1. Выполнить команду в командной строке mvn –v
2. Если отобразится информация о Maven, то значит, что процесс установки завершен успешно

Все готово к работе

# 4. Система контроля версий Git

# 5. Веб драйвер Selenium WebDriver

# 5. Веб драйвер менеджер WebDriverManager

# 6. Тестовый фреймворк Junit

# 7. Логирование Log4J

# 8. Конфигурирование Owner

# 9. Домашнее задание

# Предусловия

Требования к знаниям

- Знание теории тестирования

- Знание языка программирования (предпочтительнее Java, Ruby, Python)

- Знание систем контроля версий (Git)

# Урок 01. Вводное занятие

## Цели

Рассказать про автоматизацию тестирования.

## План

1. Что такое автоматизация тестирования?

2. Какие есть преимущества и недостатки у автоматизации тестирования?

3. В каких случаях полезна автоматизация тестирования?

4. Какие существуют направления автоматизации тестирования?

5. Из каких этапов состоит автоматизация тестирования?

6. Как выглядит процесс прогона автоматизированных тестов?

7. В каком объеме тестировать автоматизированными тестами?

8. Как оценить результат выполнения автоматизированных тестов?

9. Какие существуют инструменты автоматизации тестирования?

## Результаты

Понимание сущности автоматизации тестирования.

# Урок 02. Старт проекта

## Цели

объяснить, как выбрать тестовый фреймворк;

настроить логирование, интеграцию с GIT и параметризированный запуск тестов.

## План

Язык программирования - Java (JDK)

Сборщик - Maven

Веб Драйвер - ChromeDriver

IDE - IntelliJ IDEA

Система контроля версий – Git

- Добавление библиотек

Тестовый фреймворк - JUnit

Логирование - Log4J

WebDriverManager

Owner

- Первый тест

## Результаты

Заготовка для автоматизации UI-тестирования.

# Урок 03. Веб драйвер

## Цели

научиться конфигурировать драйвер при старте,

настраивать его под специфичные задачи;

научиться задавать настройки, специфичные для конкретного браузера.

## План

WebDriver

- Конфигурирование драйвера

--Capabilities

DesiredCapabilities caps = new DesiredCapabilities()

setCapability()

Shared capabilities

Driver specific capabilities

--Options

ChromeOptions chromeOptions = new ChromeOptions()

FirefoxOptions firefoxOptions = new FirefoxOptions()

addArguments()

Browser Arguments:

–headless To open browser in headless mode. Works in both Chrome and Firefox browser

–start-maximized To start browser maximized to screen. Requires only for Chrome browser. Firefox by default starts maximized

–incognito To open private chrome browser

–disable-notifications To disable notifications, works Only in Chrome browser

setPageLoadStrategy()

PageLoadStrategy.NORMAL

PageLoadStrategy.EAGER

PageLoadStrategy.NONE

- Запуск и остановка

driver.get("https://devqa.io");

driver.getCurrentUrl();

driver.getTitle();

driver.getPageSource();

driver.close();

driver.quit();

- Запуск разных браузеров

WebDriver driver = new ChromeDriver();

WebDriver driver = new FirefoxDriver();

WebDriver driver = new EdgeDriver();

WebDriver driver = new InternetExplorerDriver();

## Результаты

Проект готовый к запуску тестов на разных браузерах и с разными настройками.

# Урок 04. Локаторы

## Цели

проанализировать все типы локаторов;

научиться строить сложные локаторы;

проанализировать какие возникают ошибки при работе с элементами;

рассмотреть подходы к хранению локаторов.

## План

- Команды поиска

driver.findElement()

driver.findElements()

Find Element From Element

Find Elements From Element

- Стратегии поиска

By.id

By.className

By.name

By.tagName

By.linkText

By.partialLinkText

By.xpath

By.cssSelector

- Виды локаторов

CSS,

XPATH,

JQUERY

сравнение CSS и xPath;

- Правила

валидация локаторов в браузере;

подходы к хранению локаторов в проекте.

## Результаты

Оптимальные локаторы элементов на странице.

# Урок 05. Веб элементы

## Цели

проанализировать какие свойства элементов страницы доступны для чтения средствами автоматизации и какие проверки можно на них строить;

объяснить какие элементы Selenium считает interactable и clickable.

## План

Webelement

click()

sendKeys()

clear()

submit()

isDisplayed()

isEnabled()

isSelected()

getText()

getTagName()

getCssValue()

getSize()

getRect()

getLocation()

getAttribute(“”)

driver.switchTo().activeElement()

- Текстбоксы

- Кнопки

- Чекбоксы

- Радиобаттоны

- Выпадающие списки

Select select = new Select(<WebElement object>);

selectByVisibleText()

select.selectByIndex()

selectByValue()

deselectByVisibleText()

deselectByIndex()

deselectByValue()

deselectAll()

getOptions()

getAllSelectedOptions()

getFirstSelectedOption()

- Таблицы

## Результаты

Умение работать с элементами

# Урок 06. Ожидания

## Цели

применять явные и неявные ожидания для повышения стабильности тестов и осуществления сложных проверок.

## План

- Implicit Waits Неявные ожидания

driver.manage().timeouts().implicitlyWait()

- Explicit Waits Явные ожидания

WebDriverWait wait = new WebDriverWait()

wait.until()

- Expected Conditions

ExpectedConditions:

alertIsPresent()

elementSelectionStateToBe()

elementToBeClickable()

elementToBeSelected()

frameToBeAvaliableAndSwitchToIt()

invisibilityOfTheElementLocated()

invisibilityOfElementWithText()

presenceOfAllElementsLocatedBy()

presenceOfElementLocated()

textToBePresentInElement()

textToBePresentInElementLocated()

textToBePresentInElementValue()

titleIs()

titleContains()

visibilityOf()

visibilityOfAllElements()

visibilityOfAllElementsLocatedBy()

visibilityOfElementLocated()

- Fluent Waits

Wait wait = new FluentWait()

- Thread.sleep Задержка

## Результаты

Умение работать с ожиданиями

# Урок 07. Окна

## Цели

поработать с нативными элементами браузера.

## План

-Переходы

driver.navigate().to("https://devqa.io");

driver.navigate().back();

driver.navigate().forward();

driver.navigate().refresh();

- Размеры окна

driver.manage().window().getSize().getWidth()

driver.manage().window().getSize().getHeight()

driver.manage().window().setSize()

driver.manage().window().getPosition().getX()

driver.manage().window().getPosition().getY()

driver.manage().window().setPosition()

driver.manage().window().maximize()

driver.manage().window().fullscreen()

- Alert

- Prompt,

- Confirm

driver.switchTO().alert.getText()

driver.switchTO().alert.accept()

driver.switchTO().alert.dismiss()

driver.switchTO().alert.sendKeys(“Text”)

- iFrame

driver.switchTo.frame()

driver.switchTo().defaultContent();

- Tabs

getWindowHandle()

getWindowHandles()

driver.switchTo().window()

// Opens a new tab and switches to new tab

driver.switchTo().newWindow(WindowType.TAB);

// Opens a new window and switches to new window

driver.switchTo().newWindow(WindowType.WINDOW);

- BasicAuth

Переключение

Размеры окна

## Результаты

Умение работать с окнами

# Урок 08. Куки, логи, скриншоты

## Цели

## План

- Куки

addCookie(arg)

getCookies()

getCookieNamed()

deleteCookieNamed()

deleteCookie()

deleteAllCookies()

- Логирование

driver().manage().logs().getAvailableLogTypes()

-- Типы логов

browser Логи от javascript консоли браузера

client Логи от клиентской части протокола Webdriver (например, Java bindings)

driver Внутренние логи драйвера (например, логи FirefoxDriver)

performance Логи относящиеся к производительности на странице (тайминги загрузки ресурсов)

server Логи от Selenium сервера

-- Уровни логгирования

OFF: Логгирование отключено

SEVERE: Сообщения об ошибках. К примеру, при неизвестной команде.

WARNING: Предупреждения о том, что могло быть неверным, хоть ситуация и было успешно обработано. Например, перехваченное исключение.

INFO: Сообщения информативного характера. Например, о полученных командах.

DEBUG: Сообщения для дебаггинга. Например, информация о состоянии драйвера.

ALL: Все сообщения. Это способ получить все сообщения независимо от его уровня.

-- Получение логов

Logs logs = driver.manage().logs();

LogEntries logEntries = logs.get(LogType.DRIVER);

for (LogEntry logEntry : logEntries) {

System.out.println(logEntry.getMessage());

}

-- Конфигурация логгирования

- Скриншоты

TakesScreenshot screenShot =(TakesScreenshot)driver;

FileHandler.copy(screenShot.getScreenshotAs(OutputType.FILE), new File("path/to/destination/folder/screenshot.png"));

- Chrome Devtools

## Результаты

Умение работать с куками, логами браузера, скриншоты

# Урок 09. Нестандартные задачи

## Цели

объяснить, как решать нестандартные задачи.

## План

- Цвета

private final Color HEX\_COLOUR = Color.fromString("#2F7ED8");

private final Color RGB\_COLOUR = Color.fromString("rgb(255, 255, 255)");

private final Color RGB\_COLOUR = Color.fromString("rgb(40%, 20%, 40%)");

private final Color RGBA\_COLOUR = Color.fromString("rgba(255, 255, 255, 0.5)");

private final Color RGBA\_COLOUR = Color.fromString("rgba(40%, 20%, 40%, 0.5)");

private final Color HSL\_COLOUR = Color.fromString("hsl(100, 0%, 50%)");

private final Color HSLA\_COLOUR = Color.fromString("hsla(100, 0%, 50%, 0.5)");

private final Color TRANSPARENT = Color.fromString("transparent");

- Действия мыши и клавы

Actions

keyboard events

keyDown()

keyUp()

sendKeys()

mouse events

clickAndHold()

contextClick()

doubleClick()

dragAndDrop(source,target)

dragAndDropBy(source,xOffset,yOffset)

moveByOffset(xOffset,yOffset)

moveByElement()

release()

build()

perform()

Канвас элемент

Перетягивание объекта

Скроллинг

- Выполнение JS

executeAsyncScript()

executeScript()

Скроллинг

Загрузка файлов

Видеоплеер

Веб сторадж

## Результаты

Умение решать нестандартные задачи