

Пример:



“~” – в командах обозначает директорию текущего пользователя

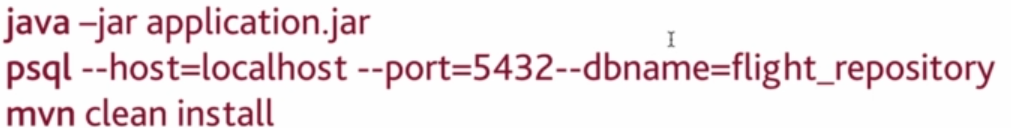
cat first.json – отобразить содержимое файла first.json на консоли

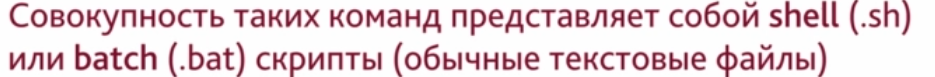
 - скопировать из директории текущего пользователя папки Download файл apache-maven.zip в директорию текущего пользователя в папку maven.

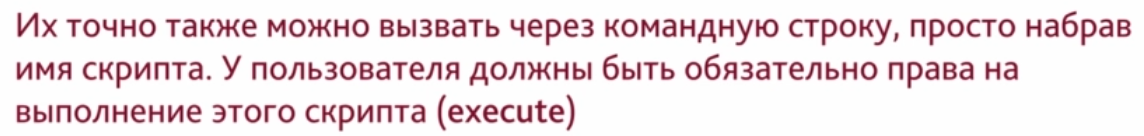
 - меняем текущую директорию на Work/test-project

Это стандартные команды, которые отличаются от того это shell или batch.

Но есть команды и для установленных приложений

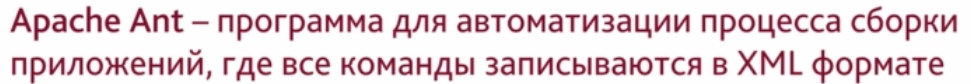


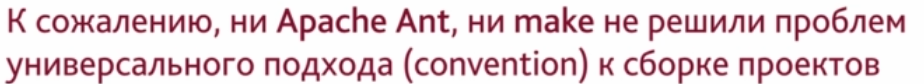


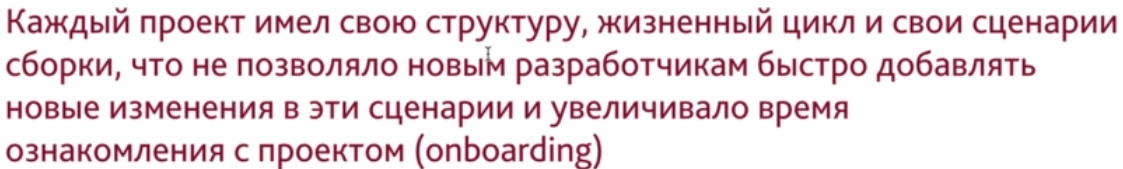


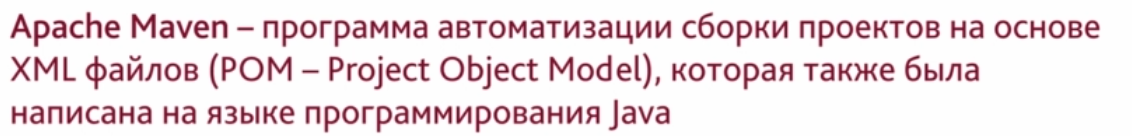
Обычно, чтобы выполнить какой-либо скрипт вызвают файл из командной строки, в котором указаны несколько таких команд.

Shell скрипты используются только для linux-систем, batch – для windows.

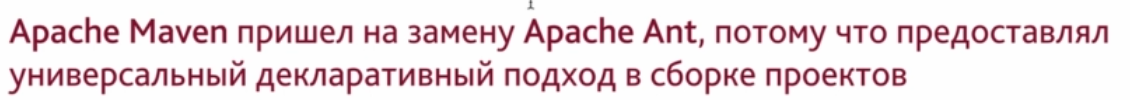


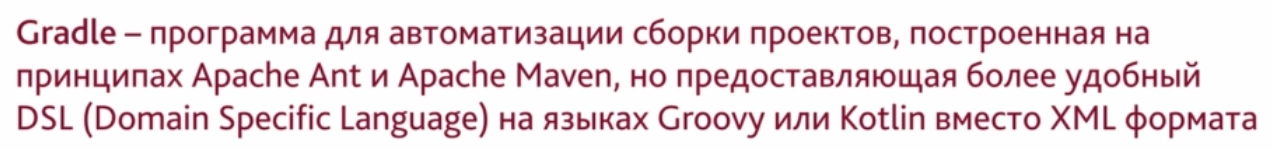














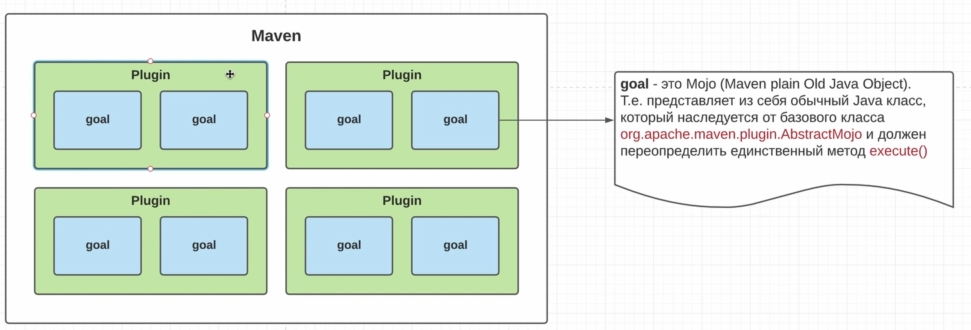
Maven – совокупность плагинов, плагин – основная часть как ядро(core) и при общении с maven мы общаемся с этими плагинами.

core – ядро maven

goal – цели

Плагин – java-проект состоящий из набора классов, и главные из этих классов это goal-ы, от 1 до нескольких. При этом каждый плагин содержит обязательно одну цель это help(служит для описания плагина)

Кроме того, goal-ы в maven называются MOJO



Mojo java класс который наследуется от AbstractMojo, который в свою очередь реализует интерфейс Mojo c методами execute(), setLog() и getLog()



А реализация метода execute() это и есть то, что выполняет goal в плагине.

При работе с плагинами вызывается тот или иной goal или их совокупность.

В core содержаться плагины которые всегда используются в java-приложениях:

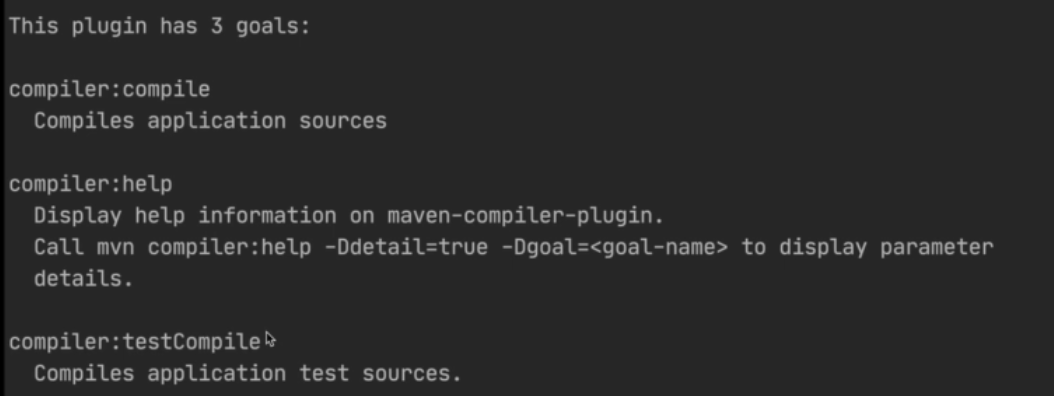
* Clean
* Compiler
* Deploy
* Install
* Resources
* Site
* Verifier
* Surefire

Чтобы набрать команду для maven пишем

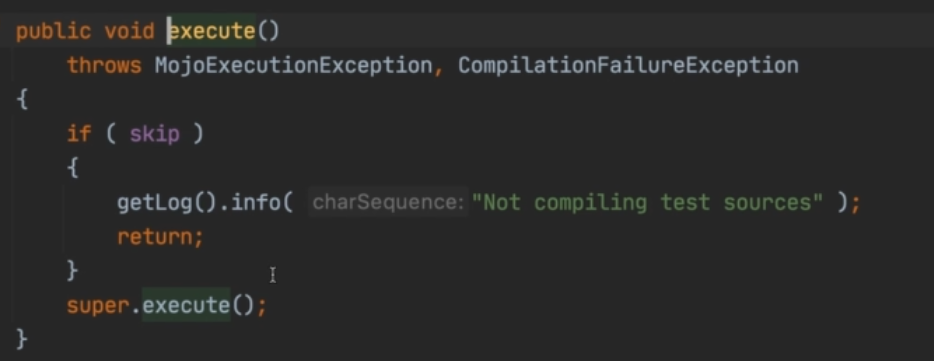
mvn – основная команда, далее идет набор параметров

Чтобы обратиться к плагину нужно набрать его имя compiler и далее после двоеточия набираем цель(goal), которая нас интересует

**mvn compiler: help** - таким образом вызывается необходимый goal в нужном плагине



Пример переопределения метода execute()



Во время запуска нашего приложения мы можем дополнительно передавать какие-то аргументы чтобы настраивать процесс запуска нашего проекта

Чтобы передавать пользовательские значения настраиваются VM Options со значком -Dключ=значение,

Чтобы передавать системные значения, они именуются -Xзначение, -XXзначение, но получить на стороне приложения мы эти значения не можем, потому что они нами не могут использоваться.

Также можно передавать Program arguments dво время запуска приложения

А через класс System можем получать JVM options пользовательские или environment