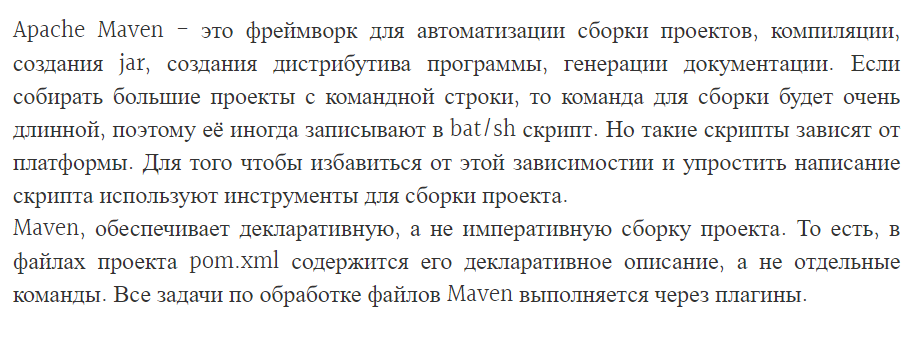
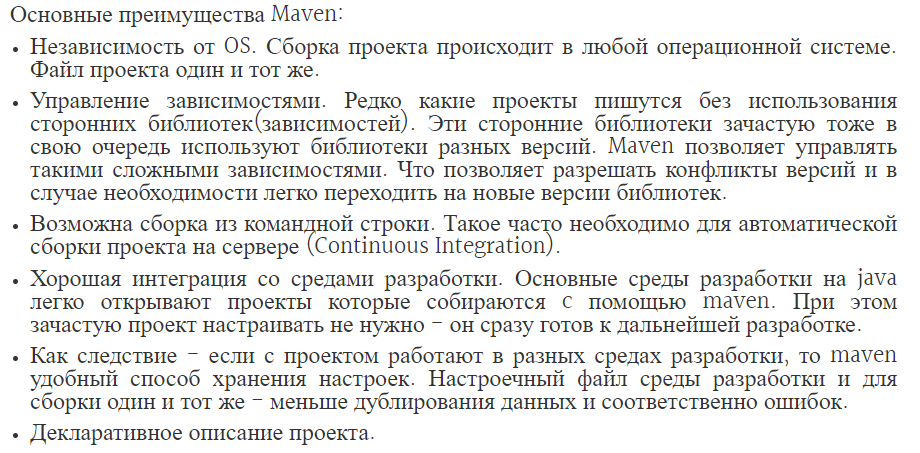
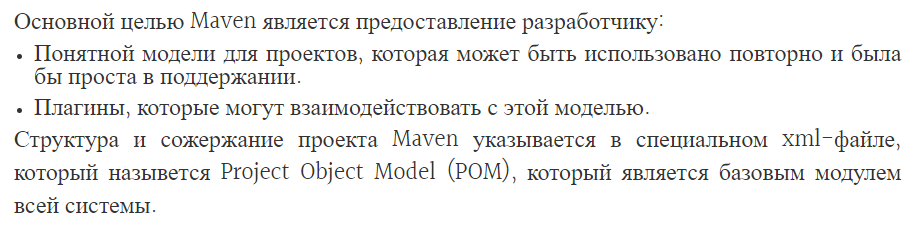
1. **Что такое Maven? Для чего создан Maven?**



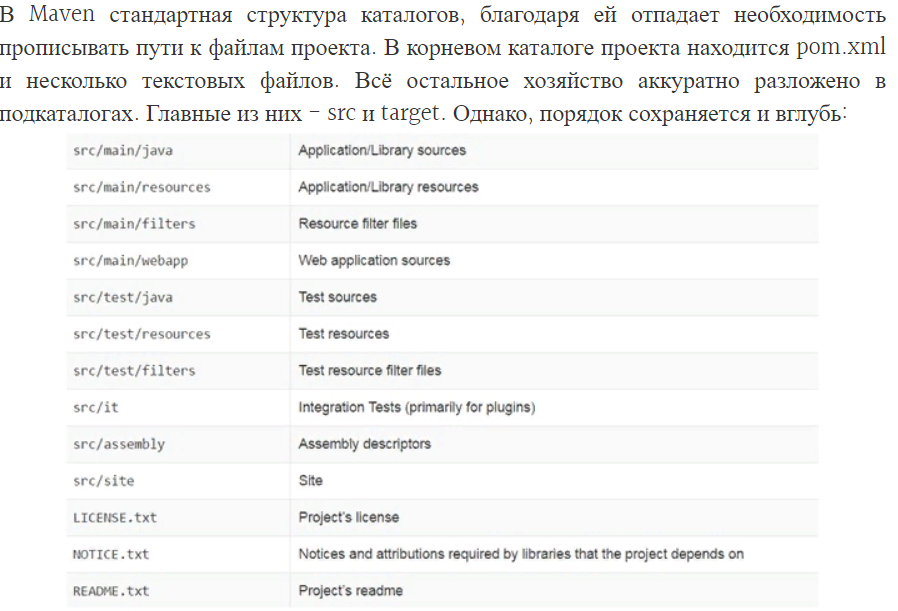




1. **Как узнать какую версию Maven вы используете?**

cmd + mvn --version

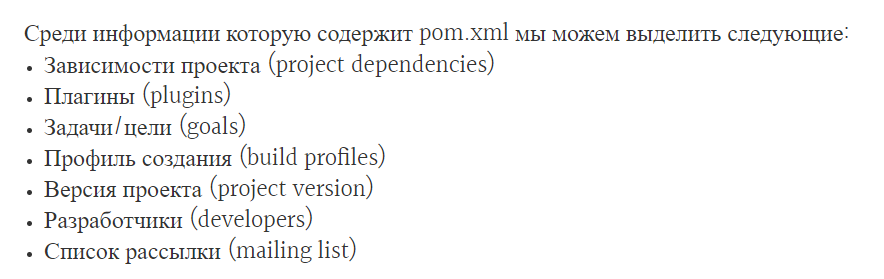
1. **Какая структура каталогов в Maven?**



**4. Что такое pom.xml?**

Информация для программного проекта, поддерживаемого Maven, содержится в XML-файле с именем pom.xml (от Project Object Model). При исполнении Мавен проверяет прежде всего, содержит ли этот файл все необходимые данные и все ли данные синтаксически правильно записаны.

1. **Какую информацию содержит pom.xml?**



Корневой элемент <project>, в котором прописана схема облегчающая редактирование и проверку, и версия POM. Внутри тэга project содержится основная и обязательная информация о проекте.

В Maven каждый проект идентифицируется парой groupId, artifactId.

Во избежание конфликта имён, groupId - наименование организации или подразделения и обычно действуют такие же правила как и при именовании пакетов в Java - записывают доменное имя организации или сайта проекта.

artifactId - название проекта.

Внутри тэга version хранится версия проекта.

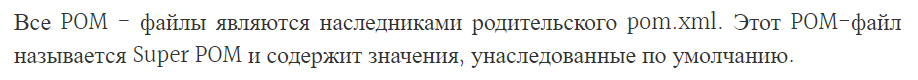
Тройкой groupId, artifactId, version (далее - GAV) можно однозначно идентифицировать jar файл приложения или библиотеки. Если состояние кода для проекта не зафиксировано, то в конце к имени версии добавляется "-SNAPSHOT" что обозначает, что версия в разработке и результирующий jar файл может меняться.

Тэг <packaging> определяет какого типа файл будет создаваться как результат сборки. Возможные варианты pom, jar, war, ear. Тэг является необязательным. Если его нет, используется значение по умолчанию - jar.

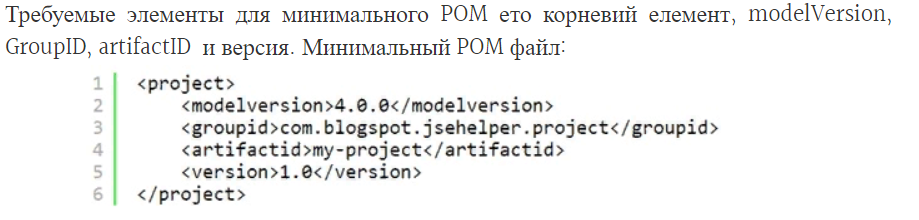
Зависимости - следующая очень важная часть pom.xml - тут хранится список всех библиотек (зависимостей) которые используются в проекте. Каждая библиотека идентифицируется также как и сам проект - тройкой groupId, artifactId, version (GAV). Объявление зависимостей заключено в тэг <dependencies>...</dependencies>.

Кроме GAV при описании зависимости может присутствовать тэг <scope>. Он задаёт, для чего библиотека используется. В данном примере говорится, что библиотека с GAV junit:junit:4.4 нужна только для выполнения тестов.

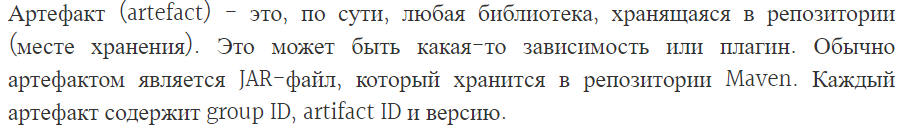
**6. Что такое супер POM?**



**7. Какие элементы необходимы для минимального POM?**



**8. Что такое артефакт? Что является полным именем артефакта?**



**9. Что такое зависимости в Maven?**

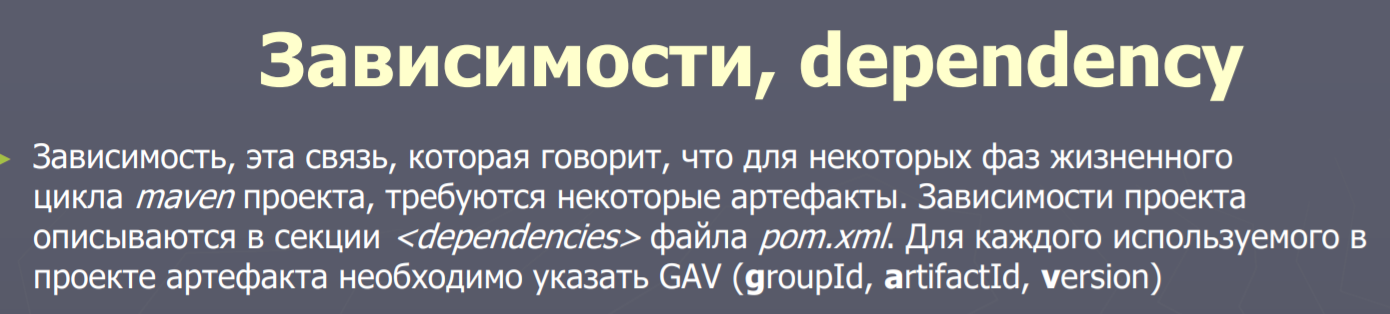
Редко когда какой-либо проект обходится без дополнительных библиотек. Как правило, используемые в проекте библиотеки необходимо включить в сборку, если это не проект OSGi или WEB (хотя и для них зачастую приходится включать в проект отдельные библиотеки). Для решения данной задачи в maven-проекте необходимо использовать зависимость dependency, устанавливаемые в файле pom.xml, где для каждого используемого в проекте артефакта необходимо указать:

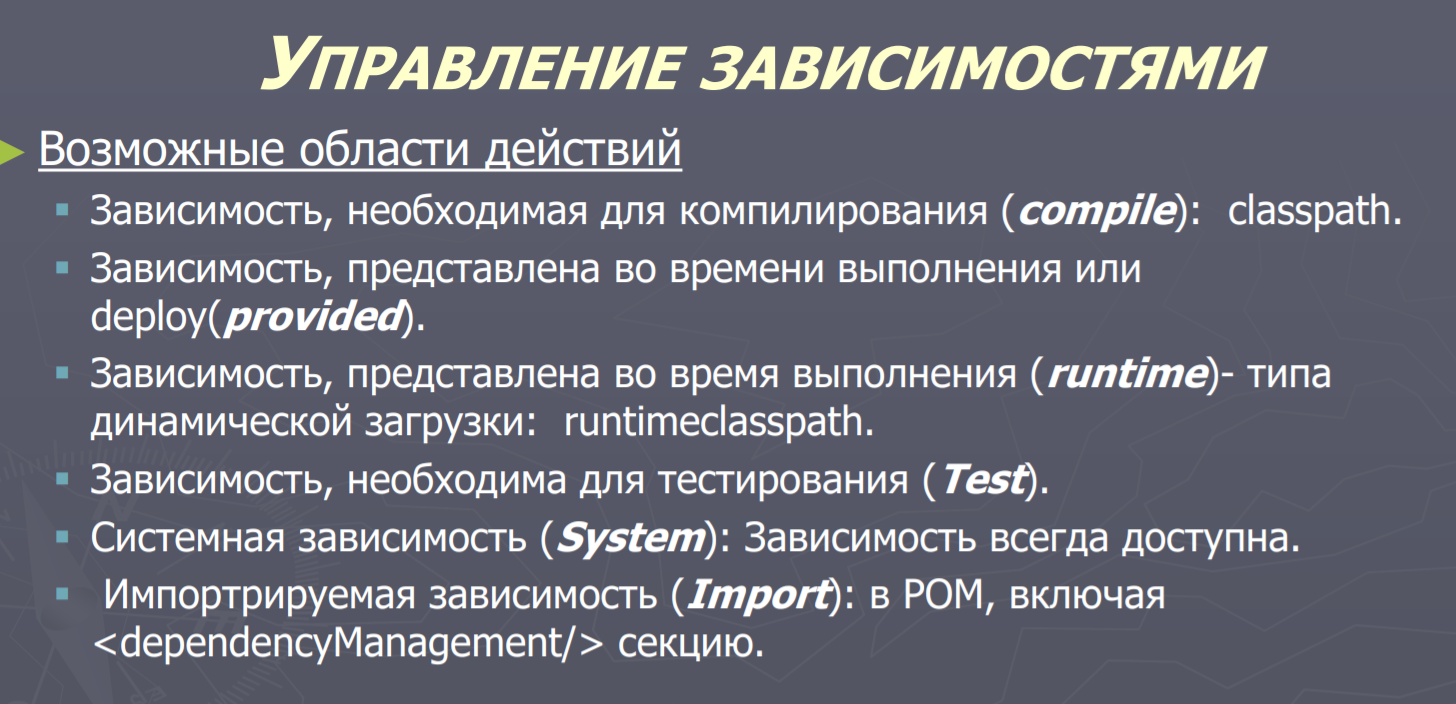
* параметры GAV (groupId, artifactId, version) и, в отдельных случаях, «необязательный» классификатор classifier;
* области действия зависимостей scope (compile, provided, runtime, test, system, import);
* месторасположение зависимости (для области действия зависимости system).

Параметры GAV

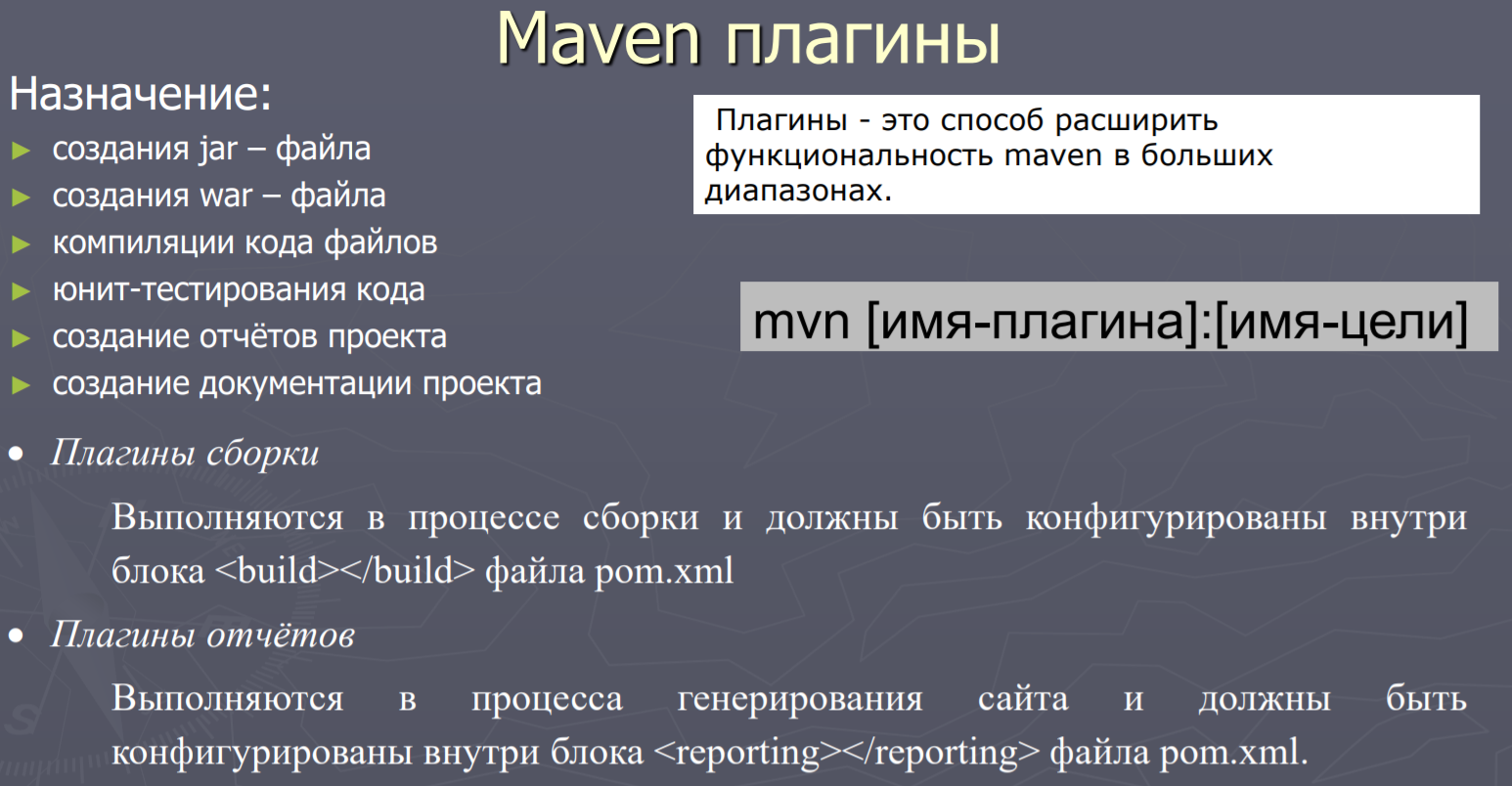
* groupId - идентификатор производителя объекта. Часто используется схема принятая в обозначении пакетов Java. Например, если производитель имеет домен domain.com, то в качестве значения groupId удобно использовать значение com.domain. То есть, groupId это по сути имя пакета.
* artifactId - идентификатор объекта. Обычно это имя создаваемого модуля или приложения.
* version - версия описываемого объекта. Для незавершенных проектов принято добавлять суффикс SNAPSHOT. Например, 1.0.0-SNAPSHOT.

Значения идентификаторов groupId и artifactId подключаемых библиотек практически всегда можно найти на сайте [www.mvnrepository.com](http://www.mvnrepository.com).

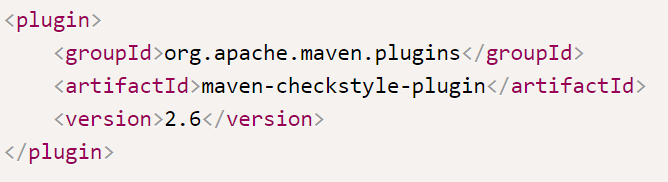




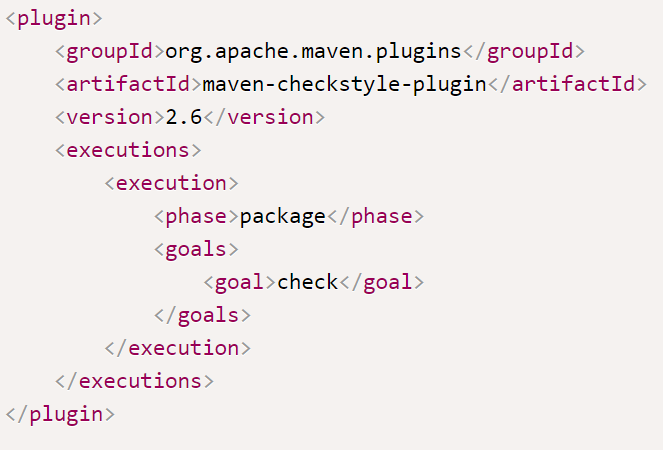
**10. Что такое плагин в Maven?**



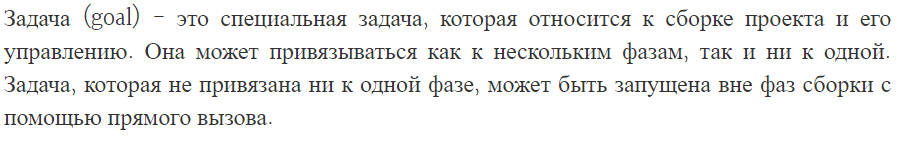
Объявление плагина похоже на объявление зависимости. Также, как и зависимости плагины идентифицируется с помощью GAV (groupId, artifactId, version). Объявление плагина в pom.xml позволяет зафиксировать версию плагина, задать ему необходимые параметры, привязать к фазам.



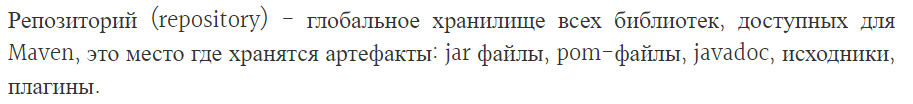
После того как плагин объявлен, его можно настроить так, чтобы он автоматически запускался в нужный момент. Это делается с помощью привязки плагина к фазе сборки проекта. В данном примере плагин запустится в фазе проекта package:

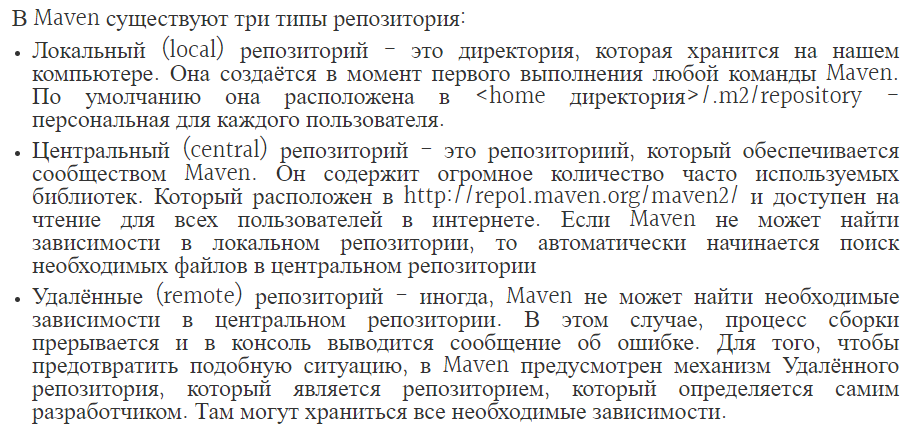


**11.Что такое задача в Maven?**

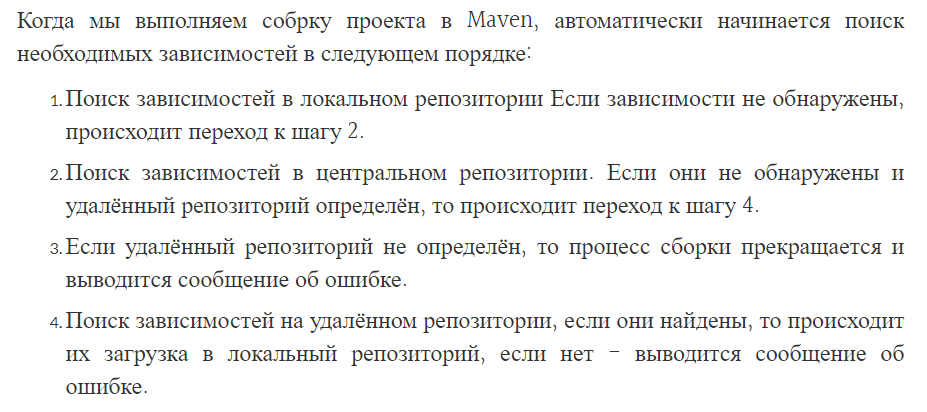


**12.Что такое репозиторий в Maven? Какие типы репозитория существуют в Maven?**

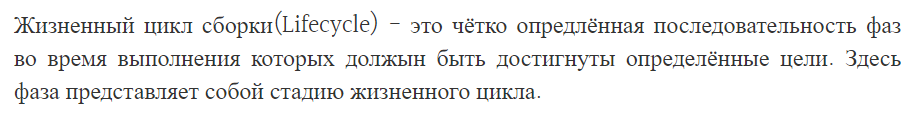




**13.Какой порядок поиска зависимостей Maven?**



**14.Назовите основные фазы жизненного цикла сборки Maven?**

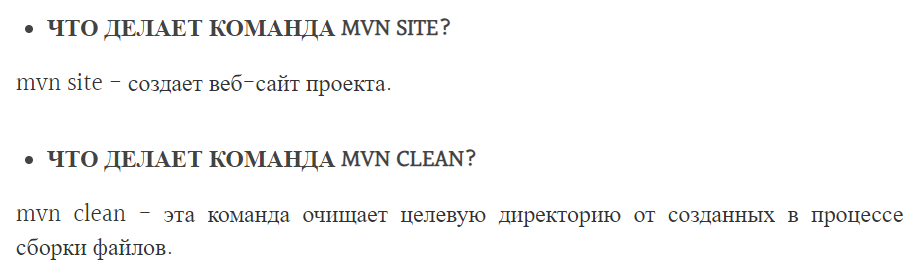
****

Процесс построения приложения называют жизненным циклом Maven-проекта, и состоит он из фаз (phase). Посмотреть на них ты можешь в IDEA, нажав на Maven>example>Lifecycle в правом верхнем углу.

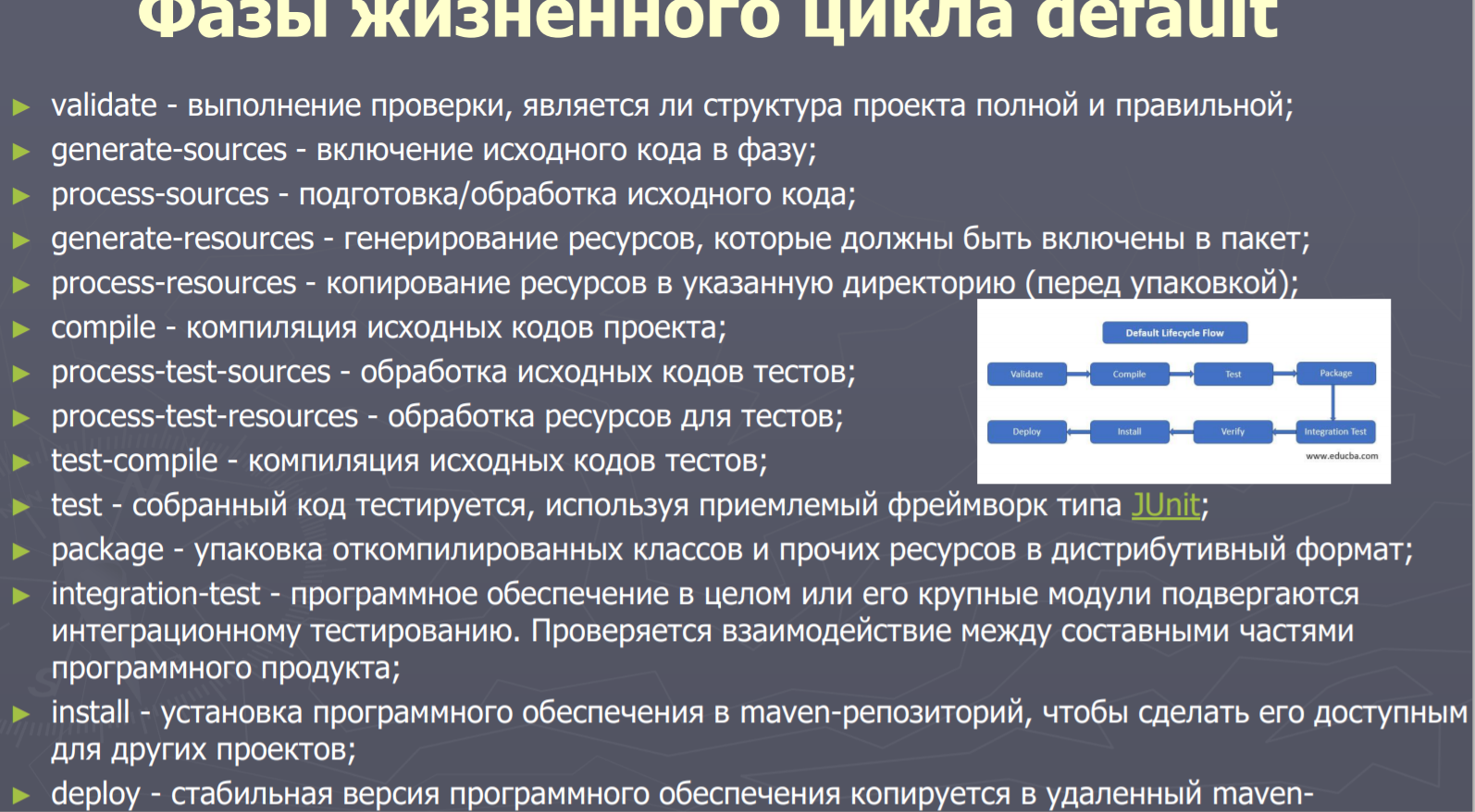
Существует 9 фаз:

* clean — удаляются все скомпилированные файлы из каталога target (место, в котором сохраняются готовые артефакты);
* validate — идет проверка, вся ли информация доступна для сборки проекта;
* compile — компилируются файлы с исходным кодом;
* test — запускаются тесты;
* package — упаковываются скомпилированные файлы (в jar, war и т.д. архив);
* verify — выполняются проверки для подтверждения готовности упакованного файла;
* install — пакет помещается в локальный репозиторий. Теперь он может использоваться другими проектами как внешняя библиотека;
* site — создается документация проекта;
* deploy — собранный архив копируется в удаленный репозиторий.

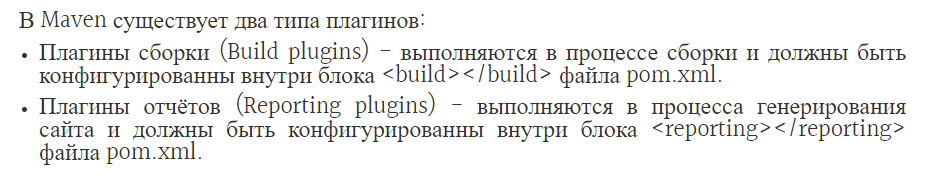
**15.Что делает команда mvn site? Что делает команда mvn clean?**



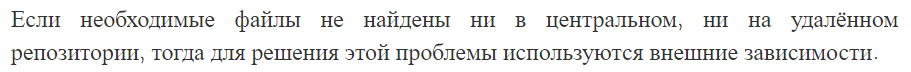
**16.Из каких фаз состоит жизненный цикл сборки Default (Build)?**



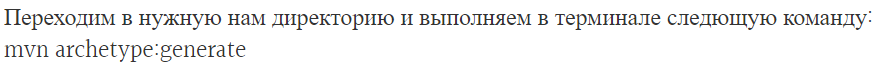
**17.Какие типы плагинов существуют в Maven?**



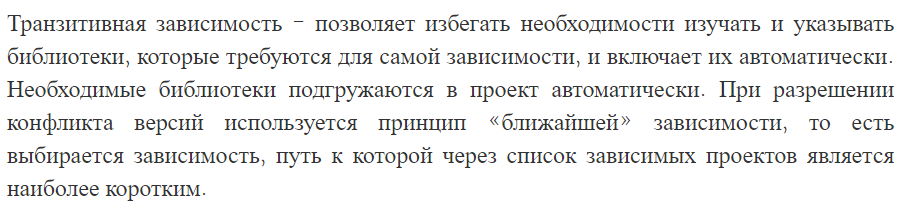
**18.Когда Maven использует внешние зависимости?**



**19.Какая команда создает новый проект на основе архетипа?**



**20.Что такое транзитивная зависимость в Maven?**

****

**21.Перечислите теги pom.xml.**

