01.03.25. Говорим о сборке в Java

JAR -> \*.JAR – типа зипник, который можно потом распаковать и потыкаться

\/

Make – есть .txt файлик, в котором лежит описание всех этапов РУКАМИ, например А зависит от Б, Б зависит от С, и тд. Проблема – миллион этапов, очень много, не читабельно, легко допустить ошибку

Ant – начался где-то в 2000 и умер в 2005

Использует в себе .xml, а не txt. Но проблема зависимостей как в make осталась и надо было что-то делать…

Потом появился *хороший* инструмент – **Maven**

Тоже использует xml – там реализовано декларативная сборка (говорим что хотим делать, Maven сам запускает процессы создания, оптимизации и тд)

Maven – целый фреймворк (накладывает ограничения (главное отличие от библиотек) на структуру директорий)

SRC

\\_ Main строгая иерархия

\\_ java

\\_recourse

\\_test

\\_ …

Pom.xml – файл, который и используется в maven

У maven есть своя структура:

1. clean(удаляет исходники, jar и тд)
2. validate (проаерка корректности структуры)
3. compile (компилируется? Круто, го некст)
4. test (тестируем (самый часто отключаемый этап))
5. package -> JAR
6. verife (проверяет 5-й этап, все ли в норме, ничего не потеряли)
7. **install** (\*)
8. site (генерация документации)
9. deploy (ныне не используется(раньше лежал конфиг который можно залить на сервер и скриптом из линукса стартануть))

\* - Maven Central – единый сайт, на котором есть все зависимости любых версий. Если там чего-то нет – его либо нет, либо это не opensource

Теперь не надо ползать по зависимостям, просто обращаемся в Maven Central и кидаем это в сборщик. Он сам все выполняет, если чего-то не хватает – сам скачивает и дополняет проект

При установке Maven появляется папка .m2 – папка, в которую скачиваются все новые зависимости. Это как бы локальный репозиторий мавена (локалка от Maven Central)

После **install** jar кидается в m2, типа как своя библиотека, своя зависимость (для других проектов своя библиотека и тд)

**Maven Central**

<groupid> - перевернутый домен компании (ru.sber)

<artifactid> - название проекта

<versionid> - версия проекта

Зависимости в pom.xml

<dependensies>

<dependency>

//подключаем сюда объектв из MC

</…>

</…>

Минусы Maven

* Это все еще xml
* При введении интересных частей проекта, pom.xml быстро расширется в размерах
* Maven – медленный сборщик. Он все заново считает, собирает и тд

**Что было после Maven?**

Сделали Gradle

* Полностью сохраняет структуру данных
* Все зависимости как тянутся из MC так и продолжают браться оттуда
* Жизненный цикл изменен, но суть та же
* Папка вместо m2 – gradle
* Градл умный, если изменили 1 файл из 500 он пересоберет один файл, а не все 500

Для описания конфигов – придется учить новый язык (groovy/kotlin)

Один из небольших минусов градла – в последних версиях для андроид разработки сборщик стал сильно многословным (3 файла, которые ссылаются друг на друга -> неудобства)

А что там с Ant? В 2020 году получил реборн и получился Ivy. Взяли идею анта под xml, и подключили туда MC