‹epam›

Build: Maven

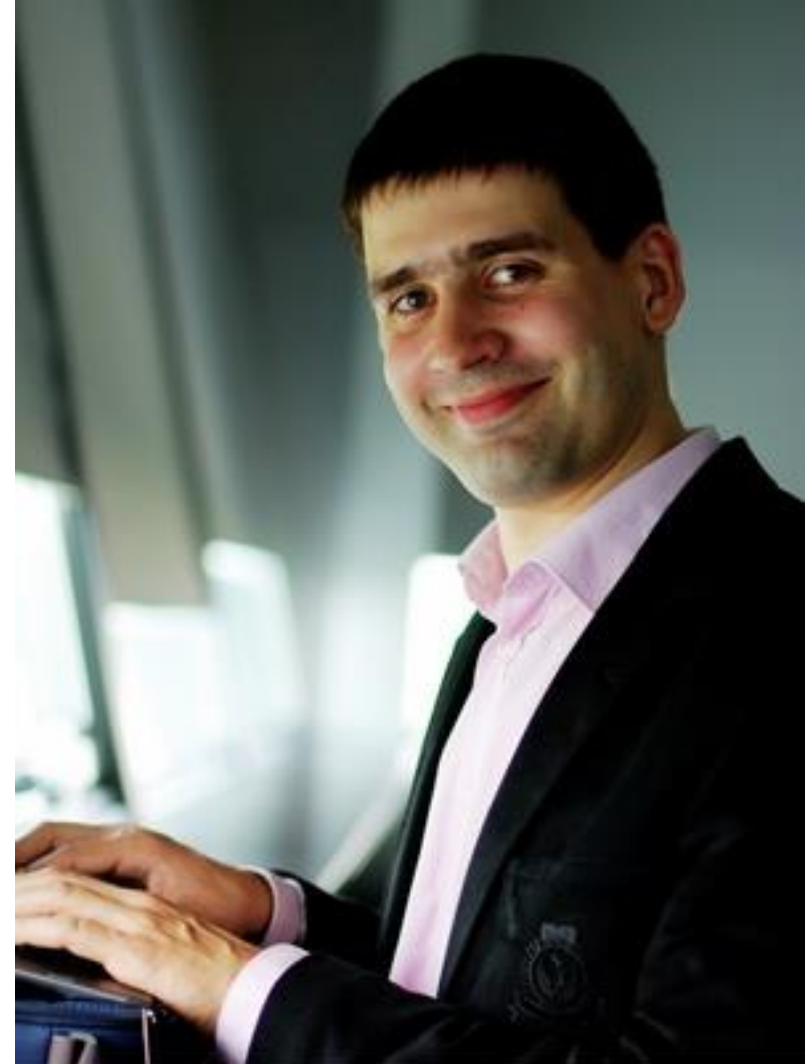**Ivan Spresov**

Start at 10.05

Dec 12, 2016

Several words about us...

Ivan Spresov

Developer, Trainer, Manager

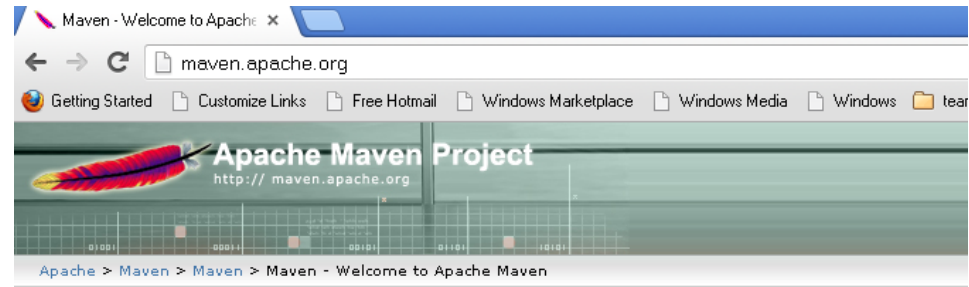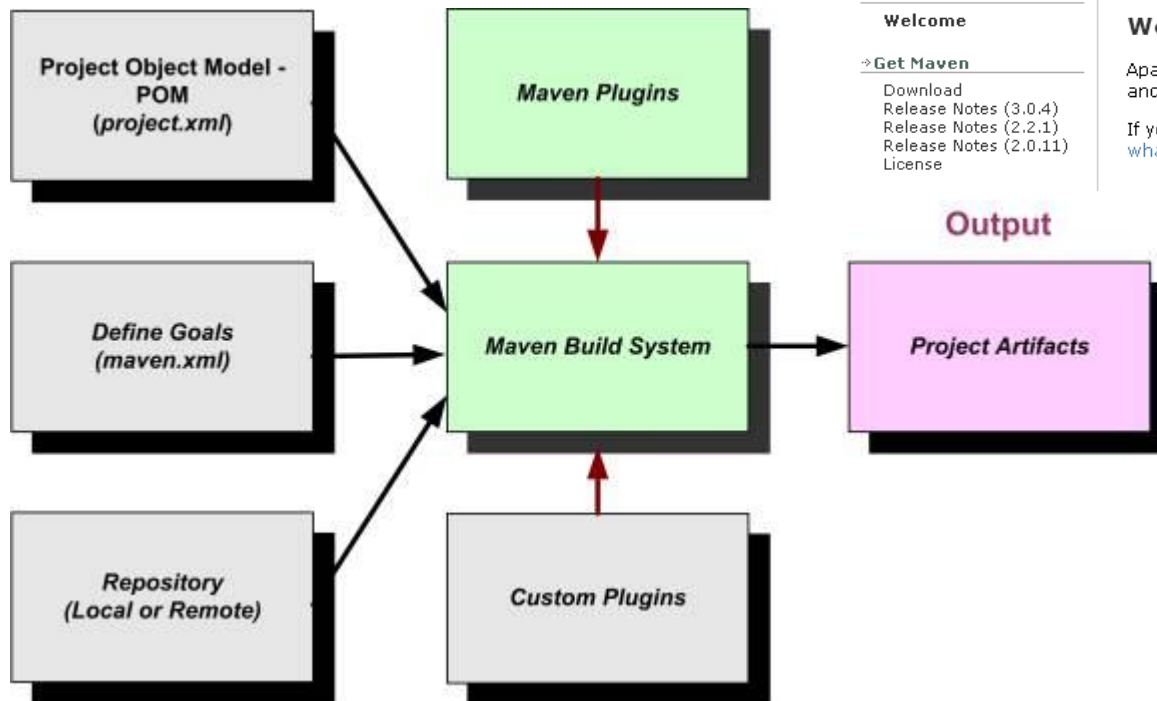# РЕГЛАМЕНТ

- Вопросы в чат
- Unmute All
- Будет перерыв

AGENDA

- **What is it?**
- Concepts
- Configuration
- Lifecycles and phases
- Profiles
- Dependencies
- Multi-Module projects
- How to …

WHAT IS IT?

standardize and simplify the build process
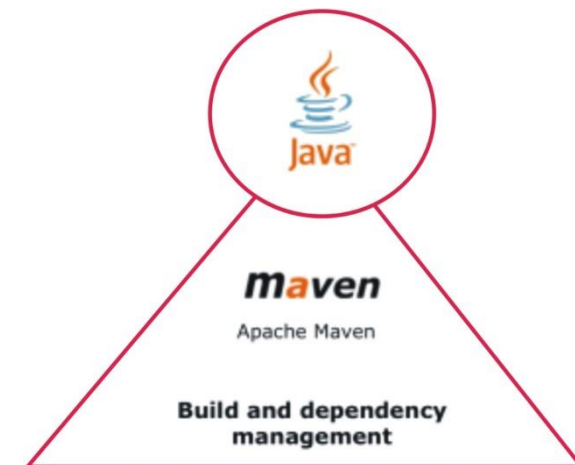
CLEAR CONCEPTS

*WHAT? - the project-specifics are **described HOW?** - implemented in reusable **plugins WHERE?** - everything is in the **repository***

WHY MAVEN?

AGENDA

- What is it?
- **Concepts**
- Configuration
- Lifecycles and phases
- Profiles
- Dependencies
- Multi-Module projects
- How to ...

PROGRAM OBJECT MODEL

Inheritance

super pom
org parent pom
project root pom
module pom

effective pom

Universal reuse through **plugins**

REPOSITORIES

Public
Org internal
Local

binaries and
metadata

DEPENDENCIES

Reuse   Made   Easy

Structure

multi-module projects
aggregation

IDE INDEPENDENCE

AGENDA

- What is it?
- Concepts
- **Configuration**
- Lifecycles and phases
- Profiles
- Dependencies
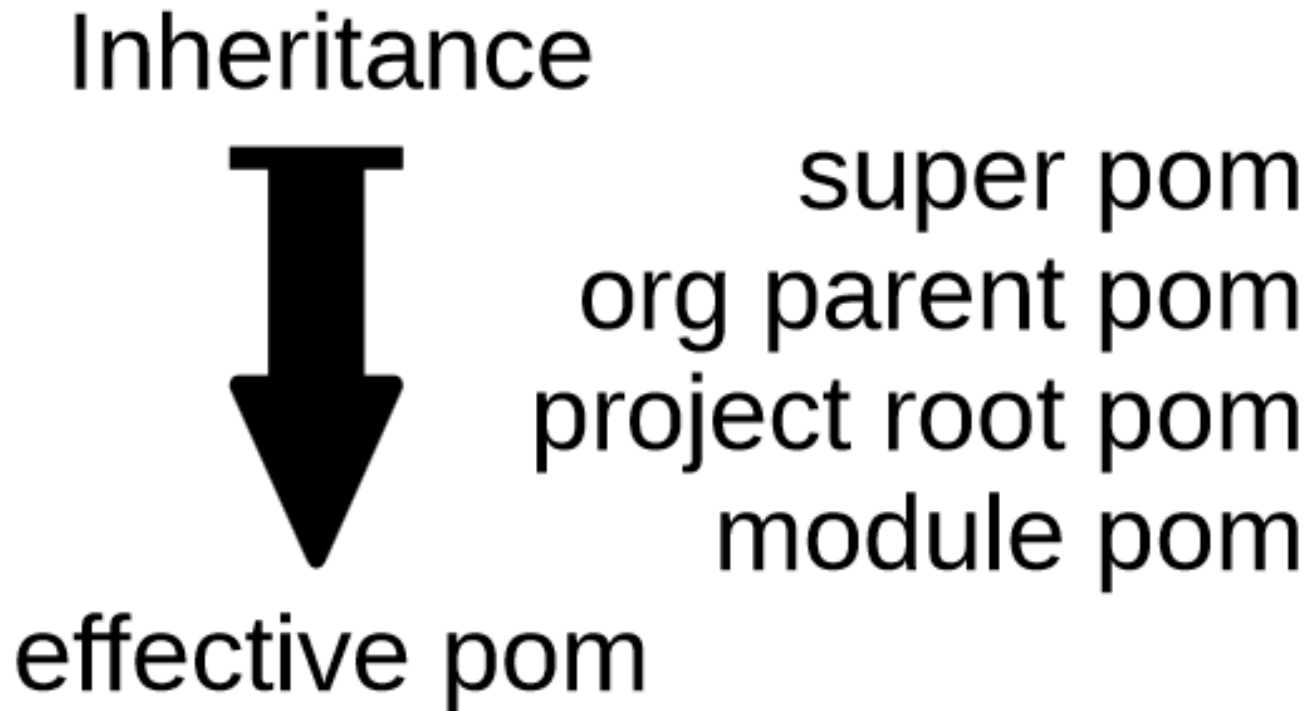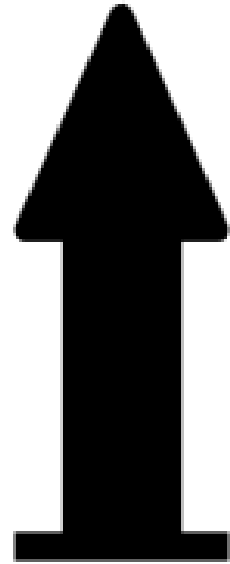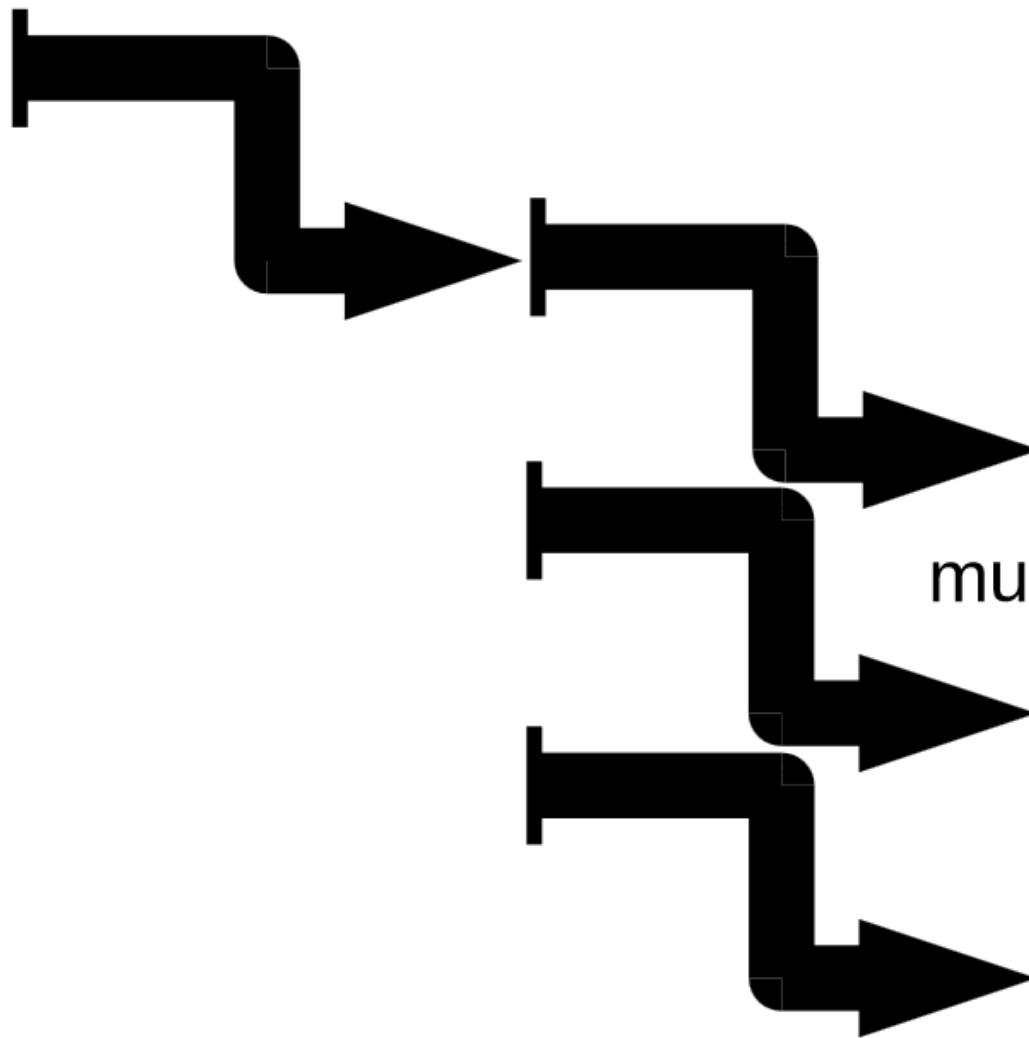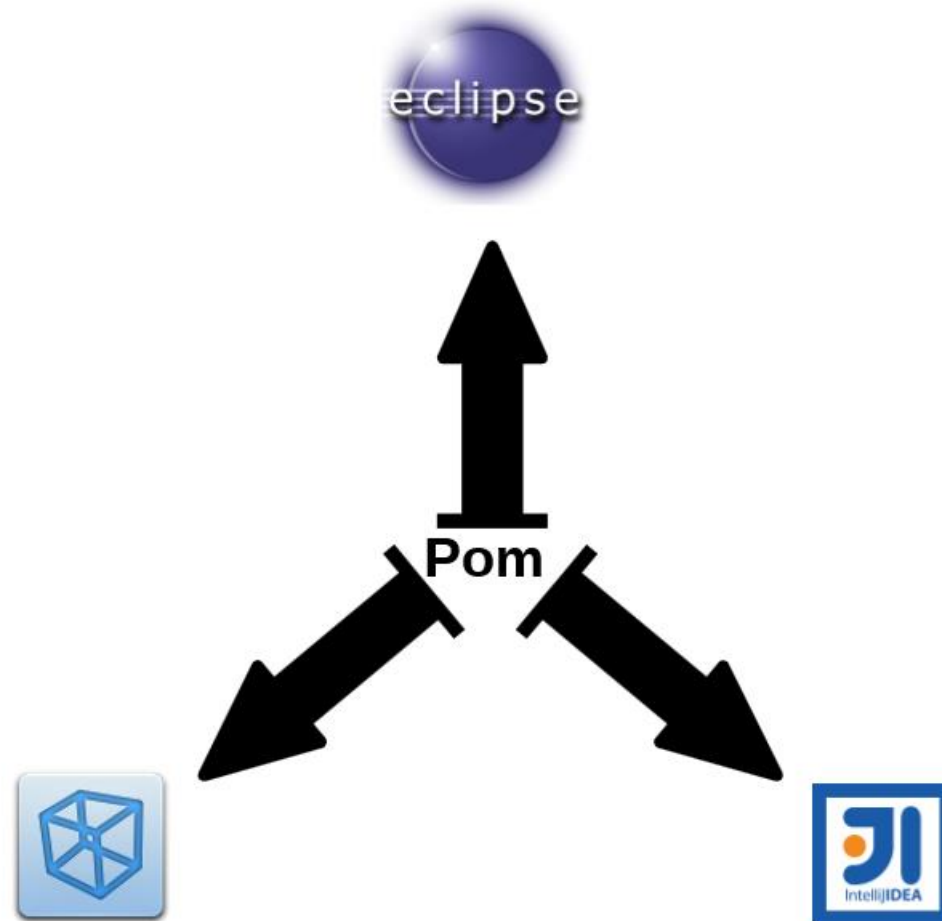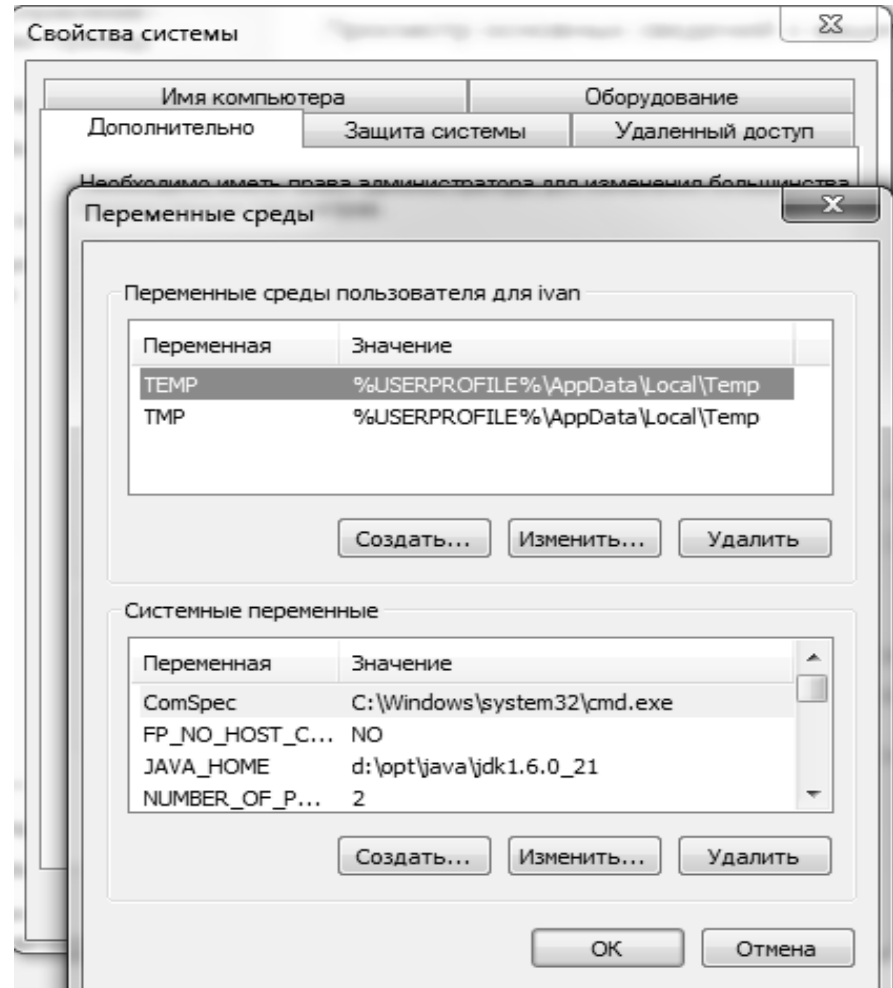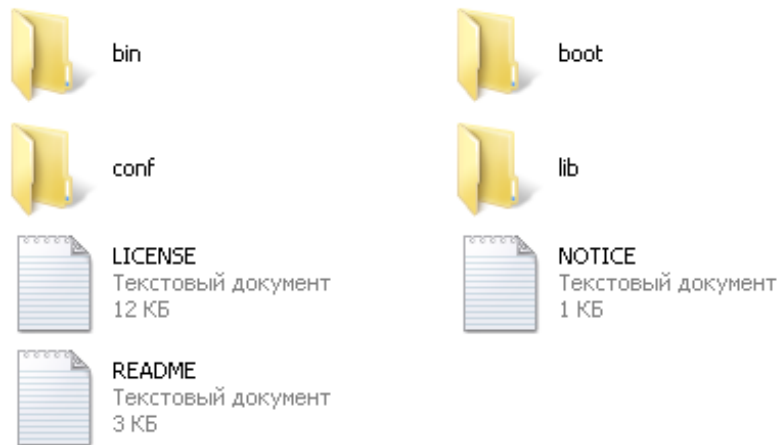- Multi-Module projects
- How to ...

MAVEN INSTALLATION

apache-maven-3.x.bin.zip

MAVEN INSTALLATION
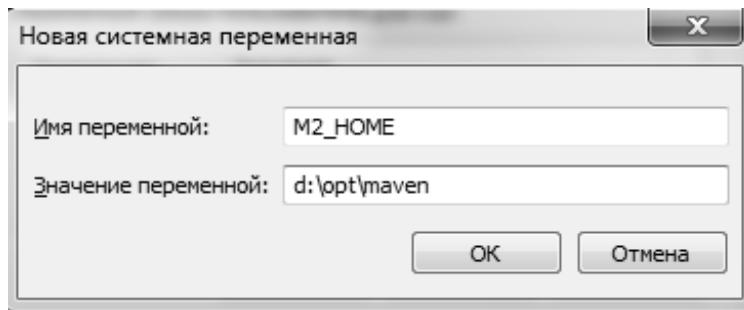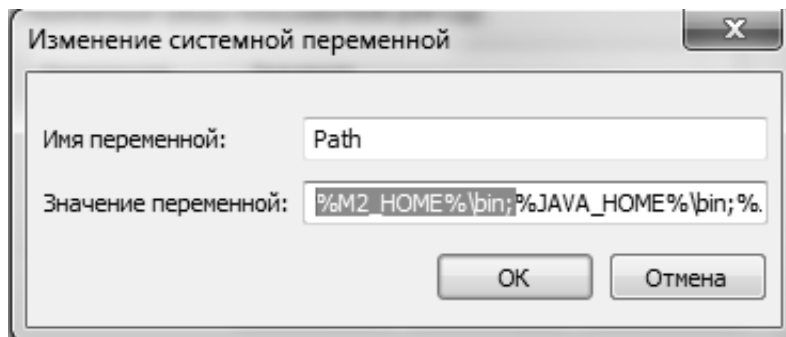
mvn -version

MAVEN INSTALL

c:\Documents and Settings
\<user>\.m2\settings.xml

```xml
<settings xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                        http://maven.apache.org/xsd/settings-1.0.0.xsd">

  <localRepository>h:\opt\globoforce\repository</localRepository>
  <profiles>
    <profile>
      <id>default</id>
      <repositories>
        <repository>
            <id>java.net</id>
          <url>http://download.java.net/maven/2</url>
        </repository>
        <repository>
          <id>JBoss repository</id>
            <url>http://repository.jboss.com/maven2/</url>
      </repository>
      <repository>
      <id>JBoss repository1</id>
      <url>http://repository.jboss.org/nexus/content/groups/public/</url>
    </repository>
     <repository>
        <id>Repo2</id>
        <url>http://repo2.maven.org/maven2/</url>
    </repository>
    <repository>
        <id>apache.snapshots</id>
        <url>http://repository.apache.org/content/groups/snapshots-group/</url>
        <releases>
          <enabled>false</enabled>
        </releases>
        <snapshots>
          <enabled>true</enabled>
        </snapshots>
     </repository>
     </repositories>
    </profile>
  </profiles>
  <servers>
    <server>
        <id>myserver</id>
        <username>managerGui</username>
        <password>abc123</password>
    </server>
  </servers>
  <activeProfiles>
    <activeProfile>default</activeProfile>
  </activeProfiles>
</settings>
```

PROJECT COORDINATES


```
<groupId>
  org.apache.maven.plugins
</groupId>
<artifactId>
  Maven-install-plugin
</artifactId>
<version>
  2.3.1
</version>
<packaging>
  Maven-plugin
</packaging>
```

groupId : artifactId : version : packaging : classifier
groupId : artifactId : version is the unique key

Versioning

0.7.1   [0.9.1]   [0.9.2,   0.9.9]   1.0.0   1.2.1   2.0.0-SNAPSHOT
                                                        2.0.0-20081117.213112-16
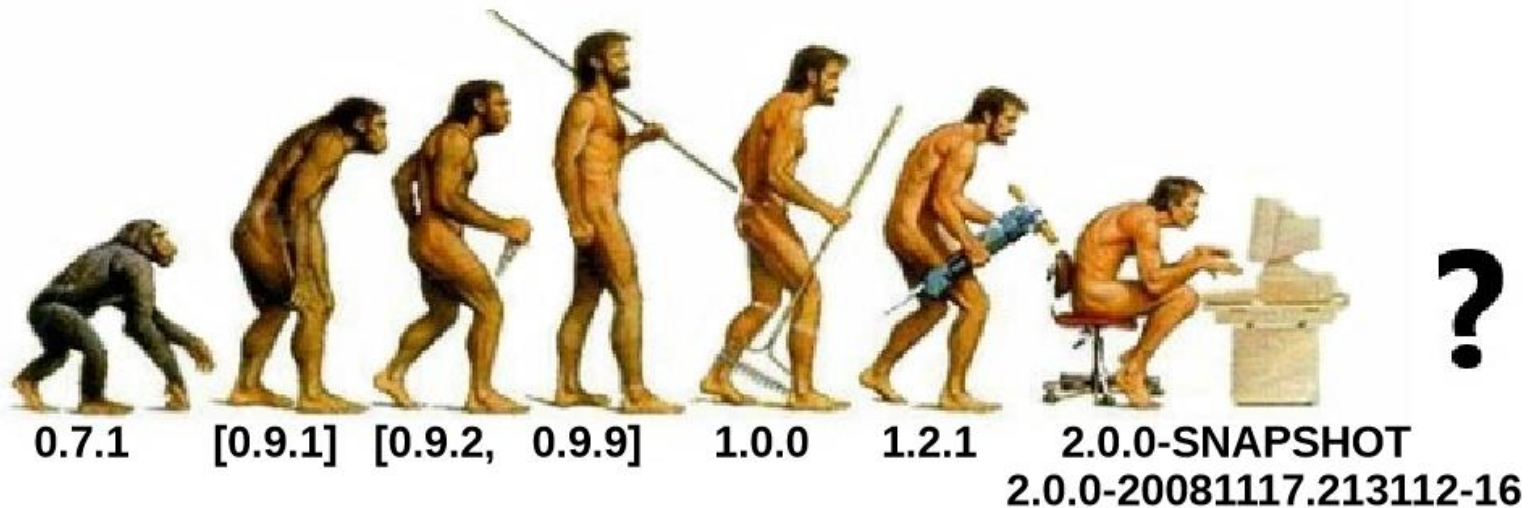
Version inheritance

Versions plugin

versions:set
versions:use-releases
versions:lock-snapshots

```xml
<parent>
    <artifactId>openjpa-parent</artifactId>
    <groupId>org.apache.openjpa</groupId>
    <version>2.0.0</version>
</parent>
<artifactId>openjpa-all</artifactId>
```

CREATION OF THE NEW PROJECT

mvn archetype:generate

Choose a number or apply filter (format: [groupId:]artifactId, case sensitive contains): 225: 225
Define value for property 'groupId': : by.academy
Define value for property 'artifactId': : app
Define value for property 'version': 1.0-SNAPSHOT: :
Define value for property 'package': by.academy: :
[INFO] ---
[[INFO] project created from Old (1.x) Archetype in dir:
H:\Work\maven\app\app
[INFO] ---
[INFO] BUILD SUCCESS
[INFO] ---
[INFO] Total time: 4:36.218s
[INFO] Finished at: Thu Jan 12 00:06:07 GMT+03:00 2012
[INFO] Final Memory: 9M/26M

CREATION OF THE NEW PROJECT

✓ *GroupId* – hierarchical location of the project in Maven repository

✓ *ArtifactId* – project identificatio

✓ *Version* – Version of the project
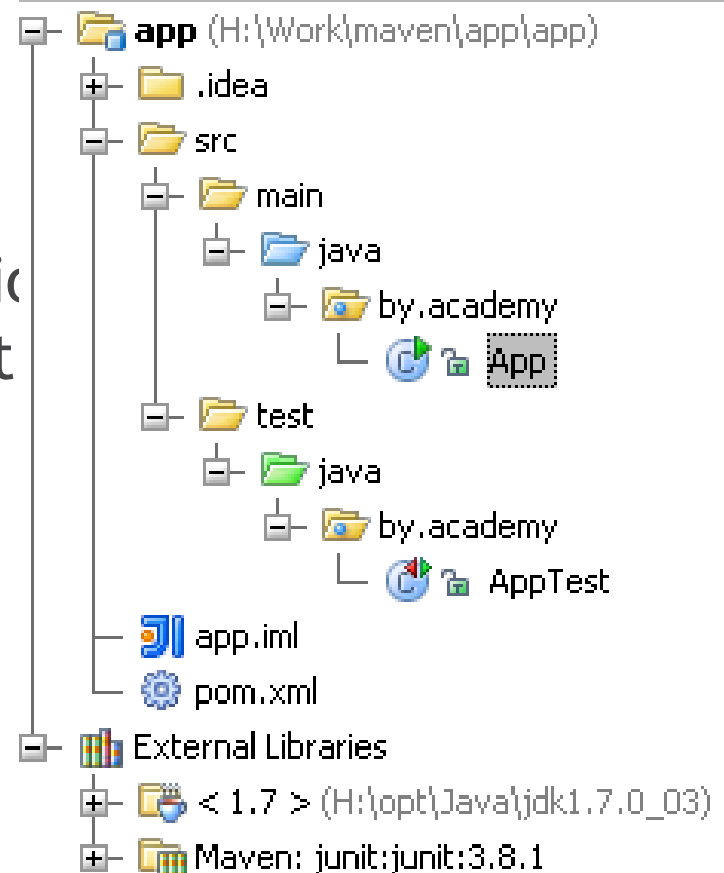
AGENDA

- What is it?
- Concepts
- Configuration
- **Lifecycles and phases**
- Profiles
- Dependencies
- Multi-Module projects
- How to …

COMPILING AND TESTING OF THE PROJECT

DEFAULT LIFECYCLE
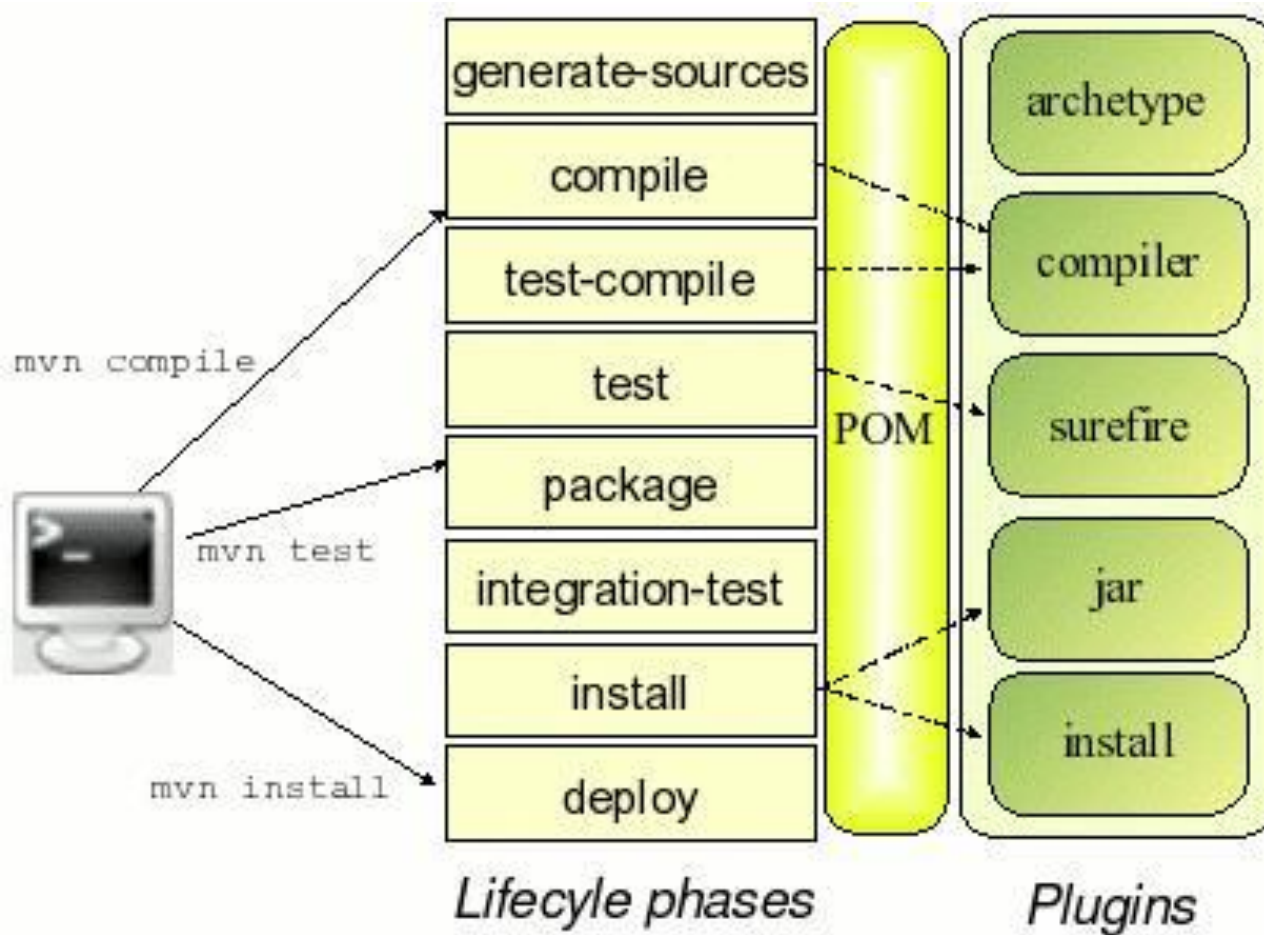
The Maven default lifecycle

COMPILE

validate → compile → test → package → integration-test → verify → install → deploy
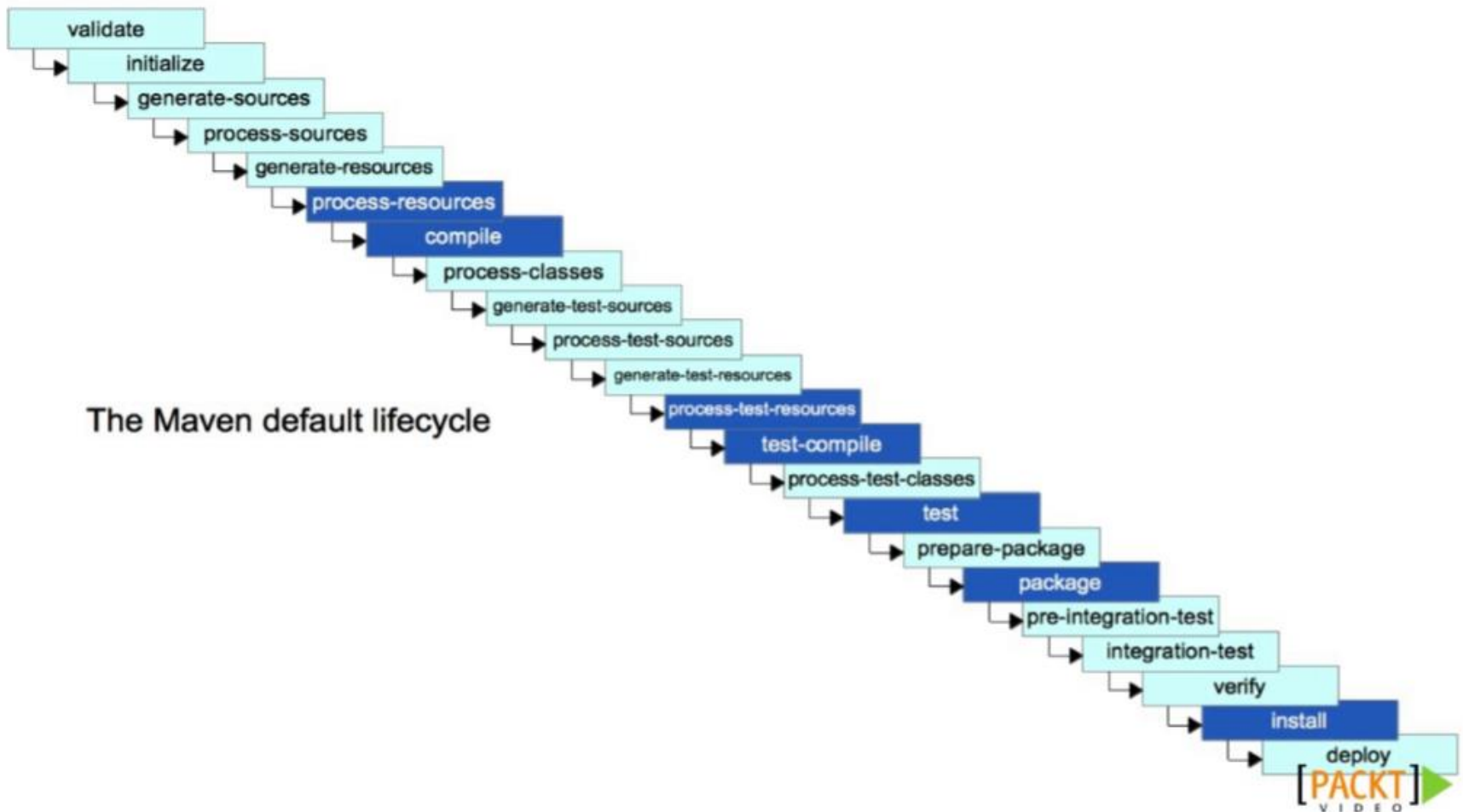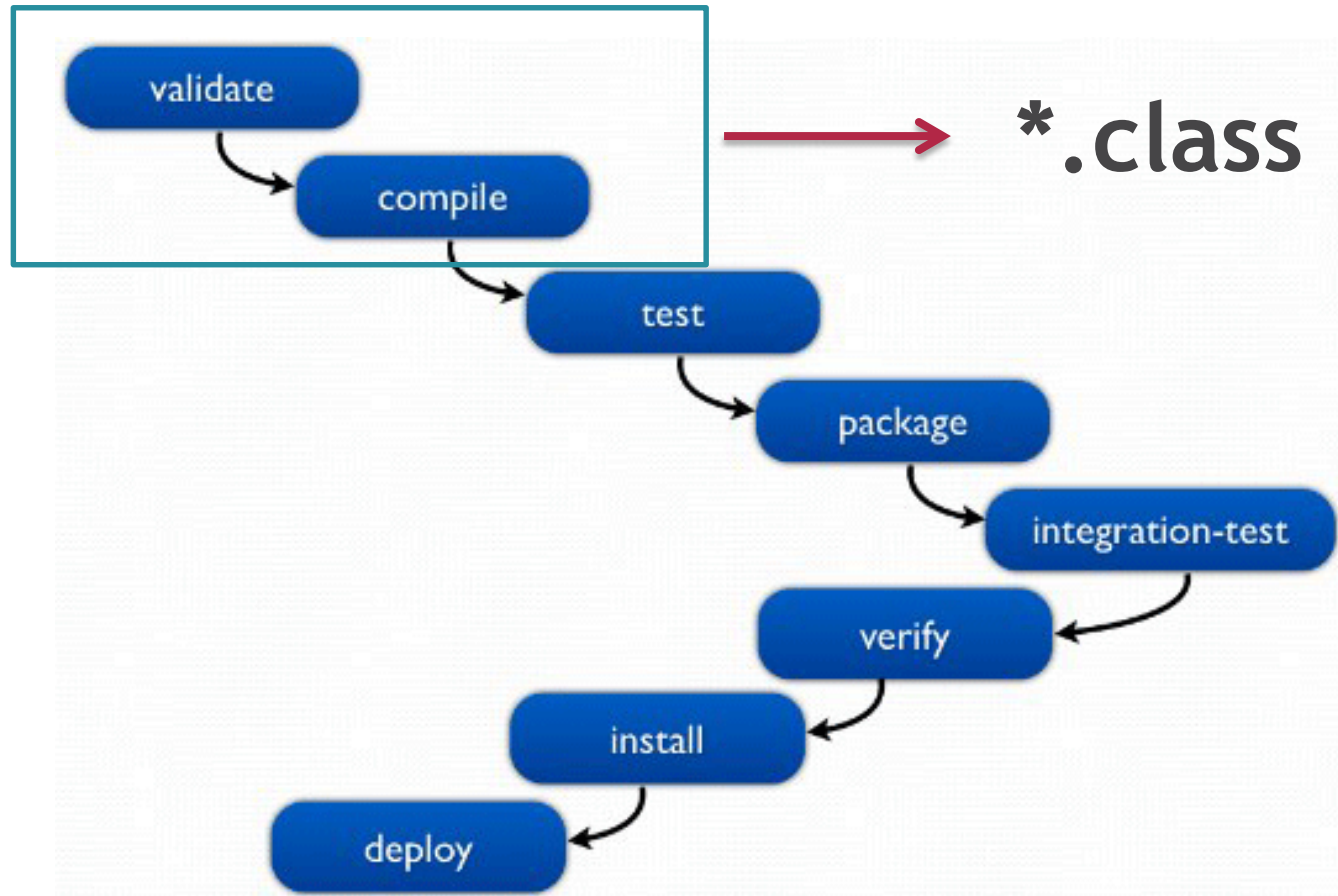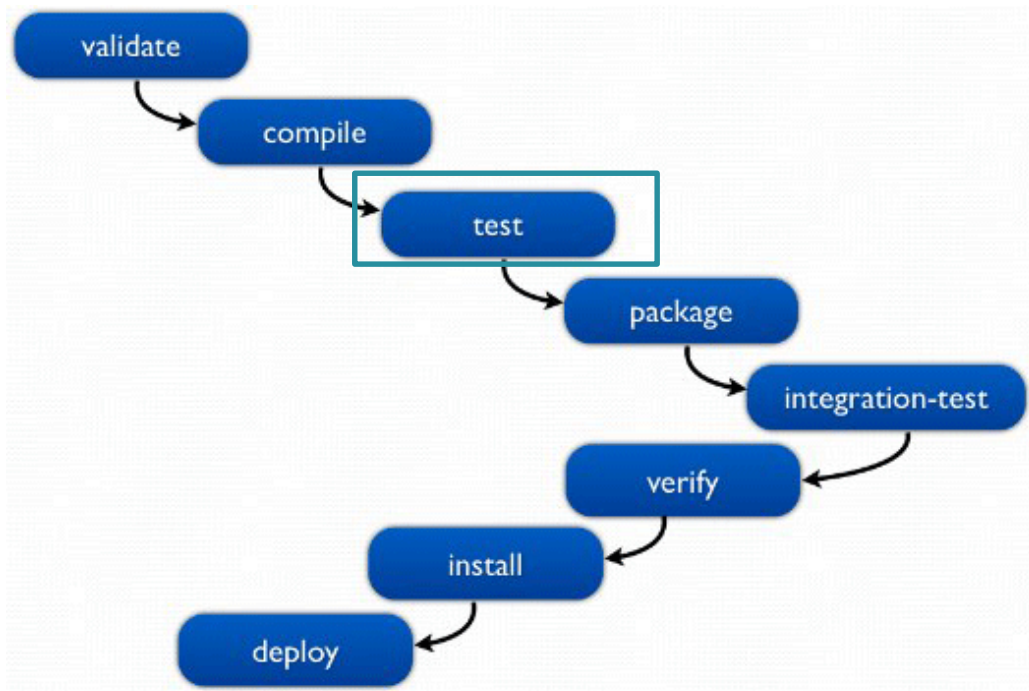
→ *.class

TEST


```java
package by.academy;

import ...

/**
 * Unit test for simple App.
 */
public class AppTest
    extends TestCase
{
    /**
     * Create the test case
     *
     * @param testName name of the test case
     */
    public AppTest( String testName )
    {
        super( testName );
    }

    /**
     * @return the suite of tests being tested
     */
    public static Test suite()
    {
        return new TestSuite( AppTest.class );
    }

    /**
     * Rigourous Test :-)
     */
    public void testApp()
    {
        assertTrue( true );
    }
}
```

PACKAGE

validate → compile → test → **package** → integration-test → verify → install → deploy

classes

surefire

test-classes

maven-archiver

surefire-reports

app-1.0-SNAPSHOT
Executable Jar File
3 КБ
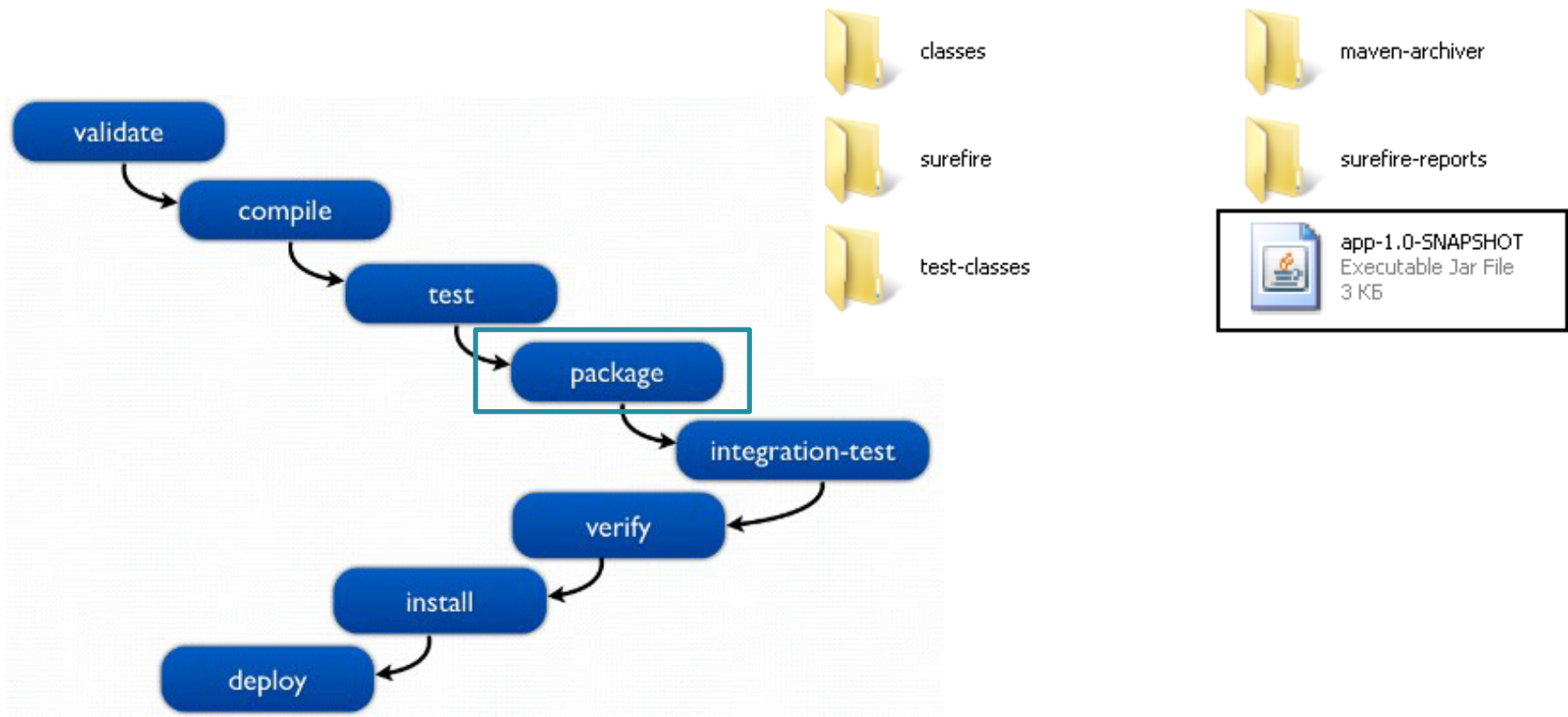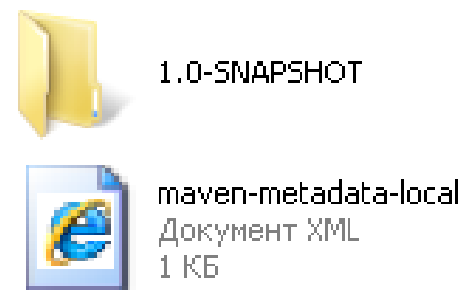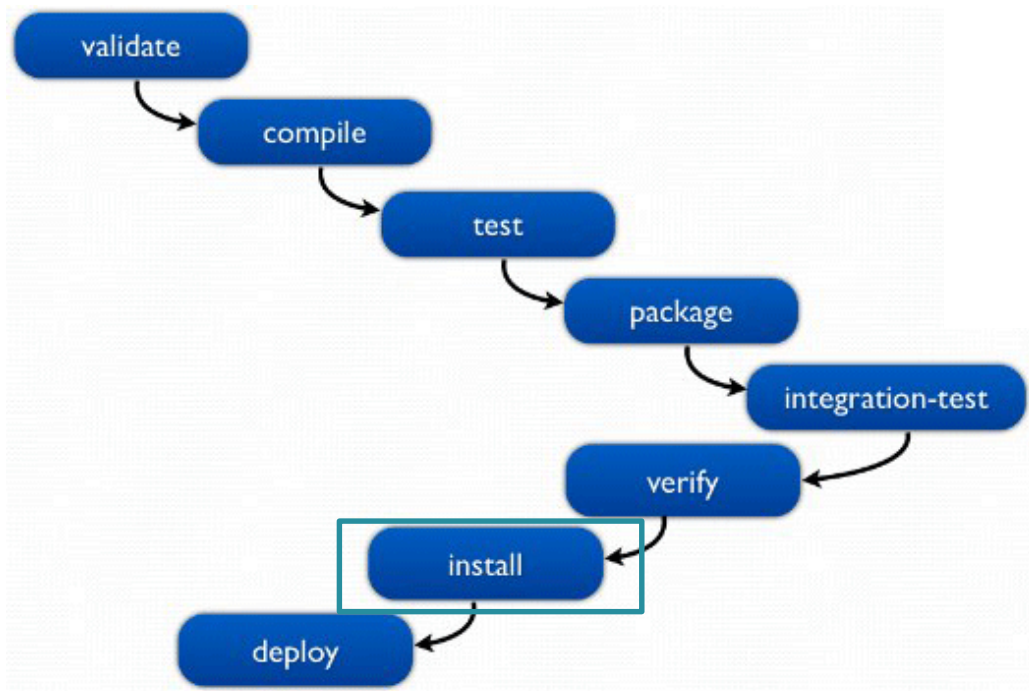
INSTALL

LYFECYCLE

Typical Project Object Model (POM) Build Life Cycle							
validate	compile	test	package	integration test	verify	install	deploy
validate all necessary information for the project	compile the source code	test the compiled source code	package up compiled source and other resources	deploy package for integration testing	verify package against criteria	install the package into the local repository	deploy the package to a remote repository

Figure 1

ЖИЗНЕННЫЙ ЦИКЛ ОЧИСТКИ

Подготовка к очистке (pre-clean)

Очистка (clean)

Фаза после очистки (post-clean)

„Default" lifecycle
For packaging: ejb, ejb3, jar, ...

Lifecycle phase	plugin:goal
process-resources	resources:resources
compile	compiler:compile
process-test-resources	resources:testResources
test-compile	compiler:testCompile
test	surefire:test
package	ejb:ejb *or* jar:jar *or* ejb3:ejb3...
install	install:install
deploy	deploy:deploy

„Clean" lifecycle:

phase	plugin:goal
pre-clean	
clean	clean:clean
post-clean	

„Site" lifecycle:

phase	plugin:goal
pre-site	
site	site:site
post-site	
site-deploy	site:deploy

LYFECYCLE

```
[INFO] Scanning for projects...
[INFO] ------------------------------------------------------------------------
[INFO] Building aproject
[INFO]    task-segment: [clean]
[INFO] ------------------------------------------------------------------------
[INFO] [antrun:run {execution: id.pre-clean}]
[INFO] Executing tasks
     [echo] in pre-clean phase
[INFO] Executed tasks
[INFO] [clean:clean]
[INFO] Deleting directory C:\dev\workspace\aproject\target
[INFO] [antrun:run {execution: id.clean}]
[INFO] Executing tasks
     [echo] in clean phase
[INFO] Executed tasks
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESSFUL
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 1 second
[INFO] Finished at: Wed Feb 13 14:44:26 PST 2008
[INFO] Final Memory: 3M/7M
[INFO] ------------------------------------------------------------------------
```

```
[INFO] Scanning for projects...
[INFO] ------------------------------------------------------------------------
[INFO] Building aproject
[INFO]    task-segment: [post-clean]
[INFO] ------------------------------------------------------------------------
[INFO] [antrun:run {execution: id.pre-clean}]
[INFO] Executing tasks
     [echo] in pre-clean phase
[INFO] Executed tasks
[INFO] [clean:clean]
[INFO] Deleting directory C:\dev\workspace\aproject\target
[INFO] [antrun:run {execution: id.clean}]
[INFO] Executing tasks
     [echo] in clean phase
[INFO] Executed tasks
[INFO] [antrun:run {execution: id.post-clean}]
[INFO] Executing tasks
     [echo] in post-clean phase
[INFO] Executed tasks
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESSFUL
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 1 second
[INFO] Finished at: Wed Feb 13 14:44:55 PST 2008
[INFO] Final Memory: 3M/6M
[INFO] ------------------------------------------------------------------------
```

```xml
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-antrun-plugin</artifactId>
            <version>1.1</version>
            <executions>
                <execution>
                    <id>id.pre-clean</id>
                    <phase>pre-clean</phase>
                    <goals>
                        <goal>run</goal>
                    </goals>
                    <configuration>
                        <tasks>
                            <echo>in pre-clean phase</echo>
                        </tasks>
                    </configuration>
                </execution>
                <execution>
                    <id>id.clean</id>
                    <phase>clean</phase>
                    <goals>
                        <goal>run</goal>
                    </goals>
                    <configuration>
                        <tasks>
                            <echo>in clean phase</echo>
                        </tasks>
                    </configuration>
                </execution>
                <execution>
                    <id>id.post-clean</id>
                    <phase>post-clean</phase>
                    <goals>
                        <goal>run</goal>
                    </goals>
                    <configuration>
                        <tasks>
                            <echo>in post-clean phase</echo>
                        </tasks>
                    </configuration>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>
```

PROJECT OBJECT MODEL
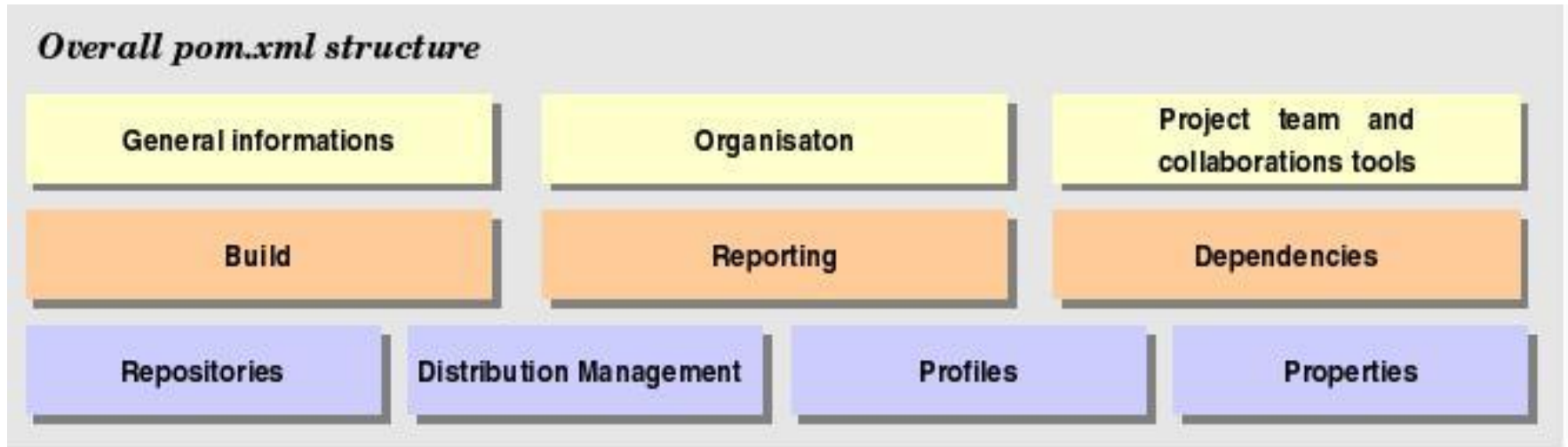
Overall pom.xml structure

Объектная модель проекта – это XML представление проекта, которая включает все метаданные, относящиеся к проекту.

OUR POM

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>by.academy</groupId>
  <artifactId>app</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>app</name>
  <url>http://maven.apache.org</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

PROJECT OBJECT MODEL

```xml
<project>
    <modelVersion>4.0.0</modelVersion>
    <!-- Базис -->
    <groupId>...</groupId>
    <artifactId>...</artifactId>
    <version>...</version>
    <packaging>...</packaging>
    <dependencies>...</dependencies>
    <parent>...</parent>
    <dependencyManagement>...</dependencyManagement>
    <modules>...</modules>
    <properties>...</properties>
```
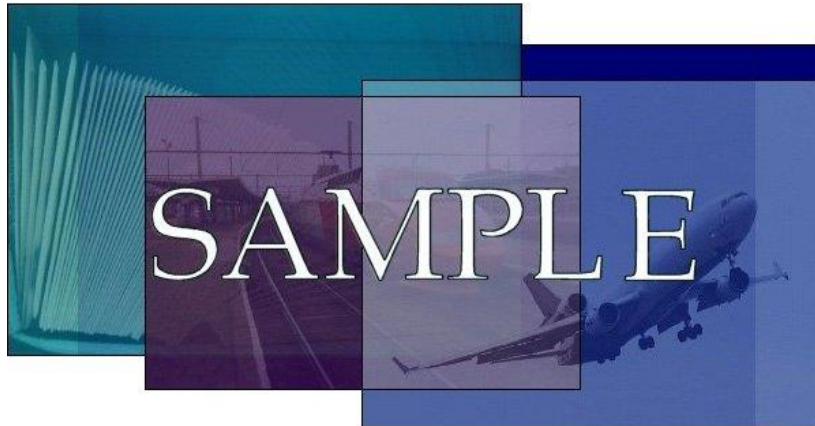
PROJECT OBJECT MODEL

```xml
<project>
    <modelVersion>4.0.0</modelVersion>
    <!-- Базис -->
    <groupId>...</groupId>
    <artifactId>...</artifactId>
    <version>...</version>
    <packaging>...</packaging>
    <dependencies>...</dependencies>
    <parent>...</parent>
    <dependencyManagement>...</dependencyManagement>
    <modules>...</modules>
    <properties>...</properties>
```

SAMPLE


```xml
<modelVersion>4.0.0</modelVersion>
<parent>
    <groupId>com.ivelon</groupId>
    <artifactId>pamm-investor</artifactId>
    <version>1.0</version>
    <relativePath>../pom.xml</relativePath>
</parent>
<artifactId>common</artifactId>
<packaging>jar</packaging>
<name>Common</name>
<version>1.0</version>
```

```xml
<project>
    <modelVersion>4.0.0</modelVersion>
    <!-- Базис -->
    <groupId>...</groupId>
    <artifactId>...</artifactId>
    <version>...</version>
    <packaging>...</packaging>
    <dependencies>...</dependencies>
    <parent>...</parent>
    <dependencyManagement>...</dependencyManagement>
    <modules>...</modules>
    <properties>...</properties>
```

PROJECT OBJECT MODEL

```xml
<!--Настройки сборки -->
<build>...</build>
<reporting>...</reporting>
```

```xml
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-jar-plugin</artifactId>
            <version>2.1</version>
            <configuration>
                <archive>
                    <addMavenDescriptor>false</addMavenDescriptor>
                </archive>
            </configuration>
        </plugin>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-resources-plugin</artifactId>
            <version>2.3</version>
            <executions>
                <execution>
                    <id>resources</id>
                    <phase>process-resources</phase>
                    <goals>
                        <goal>resources</goal>
                    </goals>
                    <configuration>
                        <outputDirectory>${project.build.directory}/classes</outputDirectory>
                        <includeEmptyDirectories>true</includeEmptyDirectories>
                        <encoding>UTF-8</encoding>
                    </configuration>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>
```

PROJECT OBJECT MODEL

```xml
<!--Метаданные проекта -->
<name>...</name>
<description>...</description>
<url>...</url>
<inceptionYear>...</inceptionYear>
<licenses>...</licenses>
<organization>...</organization>
<developers>...</developers>
<contributors>...</contributors>
```

PROJECT OBJECT MODEL

```xml
<!--Среда разработки -->
<issueManagement>...</issueManagement>
<ciManagement>...</ciManagement>
<mailingLists>...</mailingLists>
<scm>...</scm>
<prerequisites>...</prerequisites>
<repositories>...</repositories>
<pluginRepositories>...</pluginRepositories>
<distributionManagement>...</distributionManagement>
<profiles>...</profiles>
</project>
```

AGENDA

- What is it?
- Concepts
- Configuration
- Lifecycles and phases
- **Profiles**
- Dependencies
- Multi-Module projects
- How to …

PROFILE

Профайл сборки - это настройки, описанные в pom.xml, которые могут быть запущены когда необходимо.

Запуск профайла через командную строку

Запуск профайла в зависимости от настроек

Запуск профайла в зависимости от среды

START PROFILE THROUGH COMMAND LINE

```xml
<profile>
    <id>test</id>
    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-antrun-plugin</artifactId>
                <version>1.1</version>
                <executions>
                    <execution>
                        <phase>test</phase>
                        <goals>
                            <goal>run</goal>
                        </goals>
                        <configuration>
                            <tasks>
                                <echo>test profile</echo>
                            </tasks>
                        </configuration>
                    </execution>
                </executions>
            </plugin>
        </plugins>
    </build>
</profile>
```

START PROFILE THROUGH COMMAND LINE

mvn test -Ptest

START PROFILE ACCORDING TO MAVEN SETTINGS

```
<settings>
 ...
  <activeProfiles>
    <activeProfile>test</activeProfile>
 </activeProfiles>
  ...
</settings>
```

START PROFILE ACCORDING TO MAVEN SETTINGS

mvn test

START PROFILE DEPENDS ON THE ENVIRONMENT

```xml
<profile>
    <id>test</id>
    <activation>
        <property>
            <name>deployLocal</name>
        </property>
    </activation>
    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-antrun-plugin</artifactId>
                <version>1.1</version>
                <executions>
                    <execution>
                        <phase>test</phase>
                        <goals>
                            <goal>run</goal>
                        </goals>
                        <configuration>
                            <tasks>
                                <echo>test profile</echo>
                            </tasks>
                        </configuration>
                    </execution>
                </executions>
            </plugin>
        </plugins>
    </build>
</profile>
```

START ENVIRONMENT DEPENDS ON ENVIRONMENT


```
C:\WINDOWS\system32\cmd.exe

Microsoft Windows XP [Версия 5.1.2600]
(C) Корпорация Майкрософт, 1985-2001.

H:\Work\maven\app\app>mvn test -DdeployLocal

H:\Work\maven\app\app>set MAVEN_OPTS=-Xmx256M
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building app 1.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @ app ---
[debug] execute contextualize
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory H:\Work\maven\app\app\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @ app ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.5:testResources (default-testResources) @ app ---
[debug] execute contextualize
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory H:\Work\maven\app\app\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @ app ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.10:test (default-test) @ app ---
[INFO] Surefire report directory: H:\Work\maven\app\app\target\surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running by.academy.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.046 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-antrun-plugin:1.1:run (default) @ app ---
[INFO] Executing tasks
    [echo] test profile
[INFO] Executed tasks
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 7.734s
[INFO] Finished at: Thu Nov 15 02:04:05 GMT+03:00 2012
[INFO] Final Memory: 6M/15M
[INFO] ------------------------------------------------------------------------
H:\Work\maven\app\app>
```

mvn test -DdeployLocal

PROFILE

- Profile contains information which can extends main configuration or override it.

```xml
<project>
 ...
 <profiles>
  <profile>
   <id>test</id>
   <activation>...</activation>
   <build>...</build>
   <modules>...</modules>
   <repositories>...</repositories>
   <pluginRepositories>...</pluginRepositories>
   <dependencies>...</dependencies>
   <reporting>...</reporting>
   <dependencyManagement>...</dependencyManagement>
   <distributionManagement>...</distributionManagement>
  </profile>
 </profiles>
</project>
```

ADD CONFIGURATION

- Default pom

```xml
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>habr</groupId>
  <artifactId>aggr</artifactId>
  <version>1.0.0-SNAPSHOT</version>
  <packaging>pom</packaging>
  <name>Profiles Demo - Top</name>

  <modules>
    <module>module1</module>
    <module>integration-tests</module>
    <module>selenium-tests</module>
  </modules>
</project>
```

ADD CONFIGURATION

- Integration test
- mvn clean install

```
Profiles Demo - Module1 .........
Profiles Demo - Selenium Tests .
Profiles Demo - Top .............
```

```
Profiles Demo - Module1 ...........
Profiles Demo - Selenium Tests ....
Profiles Demo - Integration Tests .
Profiles Demo - Top ...............
```

```xml
<project>
 <modelVersion>4.0.0</modelVersion>
 <groupId>habr</groupId>
 <artifactId>aggr</artifactId>
 <version>1.0.0-SNAPSHOT</version>
 <packaging>pom</packaging>
 <name>Profiles Demo - Top</name>

 <modules>
  <module>module1</module>
  <module>selenium-tests</module>
 </modules>

 <profiles>
  <profile>
   <id>it</id>
   <modules>
    <module>integration-tests</module>
   </modules>
  </profile>
 </profiles>
</project>
```

OVERRIDE MODULES

```xml
<profile>
 <id>default</id>
 <activation>
  <activeByDefault>true</activeByDefault>
 </activation>
  <modules>
   <module>module1</module>
   <module>create-server</module>
  </modules>
</profile>
```

```
Profiles Demo - Module1 ......
Profiles Demo - Create Server
Profiles Demo - Top .........
```

```
Profiles Demo - Module1
Profiles Demo - Top ....
```

```xml
<profile>
 <id>noserver</id>
 <modules>
  <module>module1</module>
 </modules>
</profile>
```

PRACTICE

Настройте профиль development. Настройте следующий плагин:

```xml
<plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-jar-plugin</artifactId>
        <configuration>
          <archive>
            <manifest>
              <addClasspath>true</addClasspath>
              <classpathPrefix>lib/</classpathPrefix>
```

```xml
<mainClass>by.academy.it.App</mainClass>
            </manifest>
          </archive>
        </configuration>
      </plugin>
```
и покажите его работу.
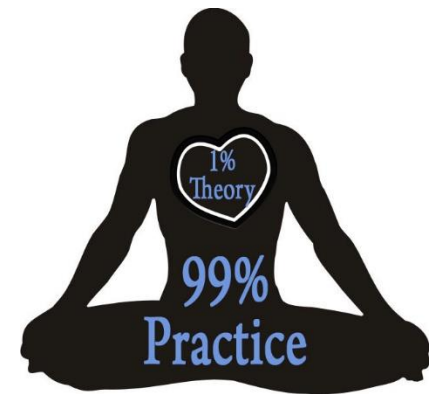
AGENDA

- What is it?
- Concepts
- Configuration
- Lifecycles and phases
- Profiles
- **Dependencies**
- **Multi-Module projects**
- How to ...

BULD AUTOMATION

mvn archetype:generate -DgroupId=by.academy.it -DartifactId=MySampleApp
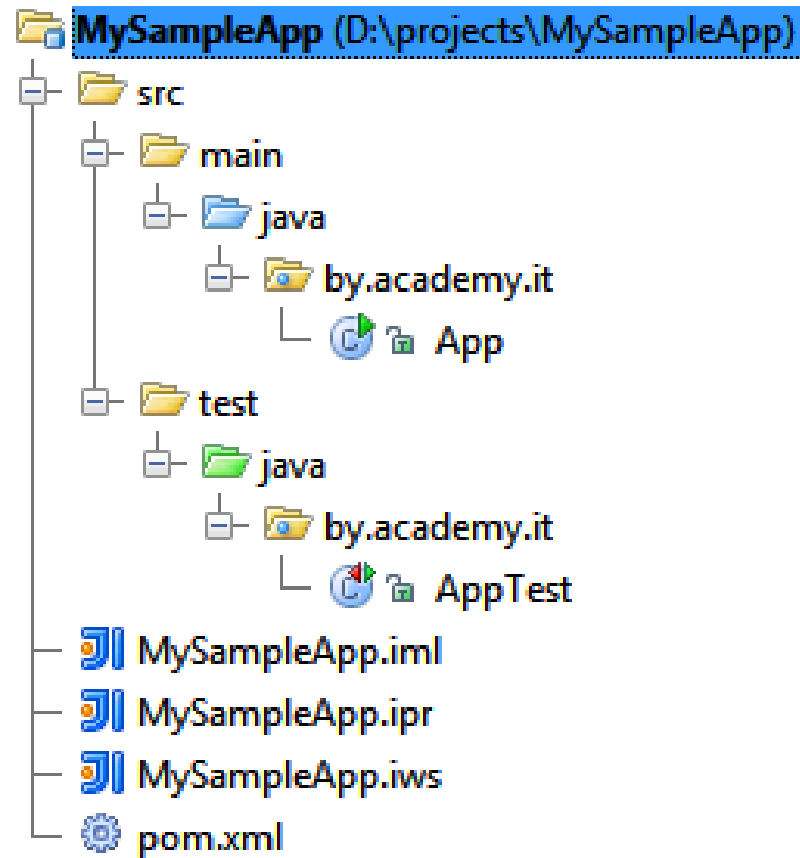
MULTI-MODULES
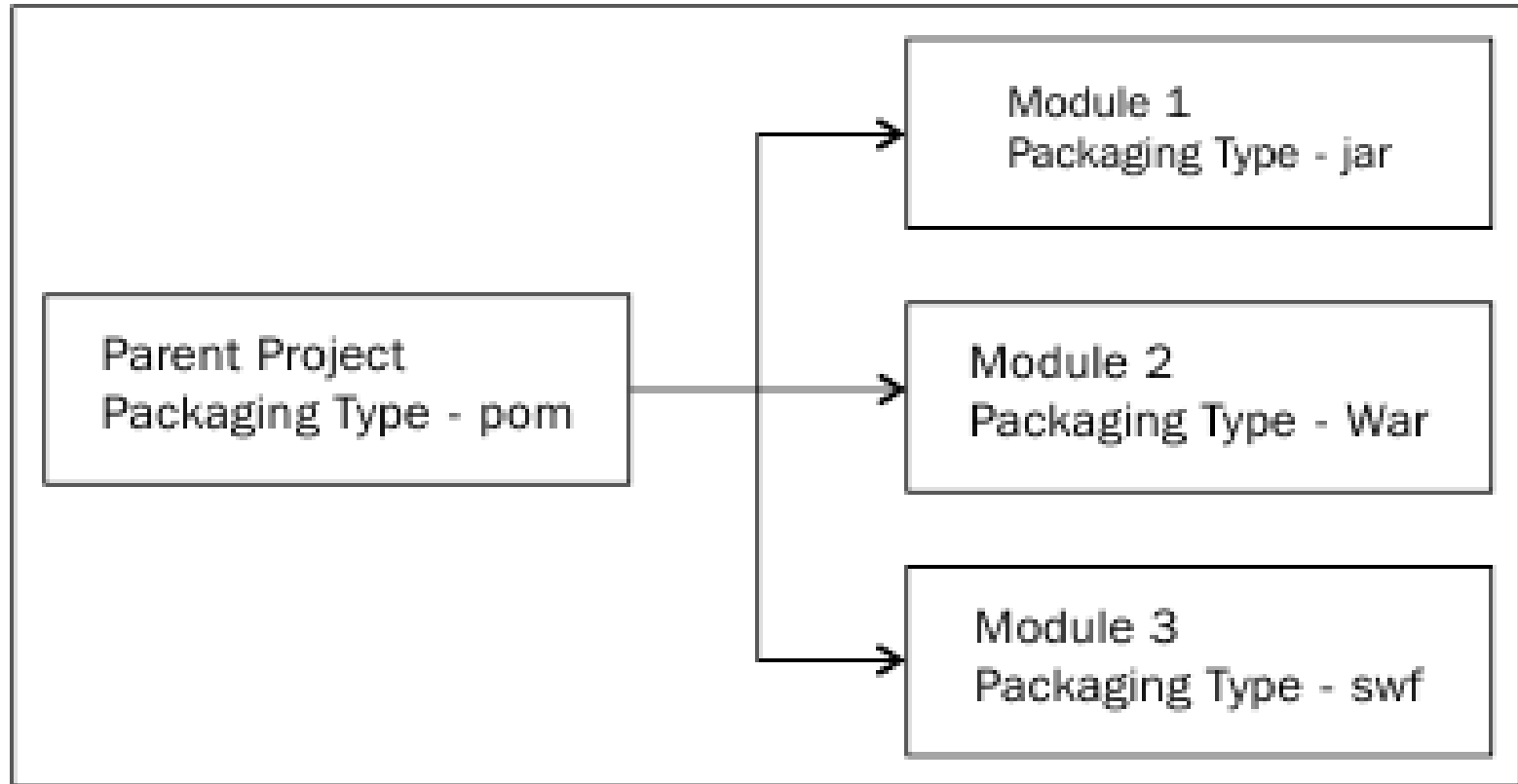
MULTI-MODULES

mvn archetype:generate

```
Choose a number or apply filter (format: [groupId:]artifactId, case sensitive contains):
225: 225
Define value for property 'groupId': : by.academy
Define value for property 'artifactId': : app
Define value for property 'version':  1.0-SNAPSHOT: :
Define value for property 'package':  by.academy: :
[INFO] ------------------------------------------------------------------------
[INFO] Parameter: groupId, Value: by.academy
[INFO] Parameter: packageName, Value: by.academy
[INFO] Parameter: package, Value: by.academy
[INFO] Parameter: artifactId, Value: app
[INFO] Parameter: basedir, Value: H:\Work\maven\app
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: H:\Work\maven\app\app
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 4:36.218s
[INFO] Finished at: Thu Jan 12 00:06:07 GMT+03:00 2012
[INFO] Final Memory: 9M/26M
```

MULTI-MODULES

базовый родительский pom-файл

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>by.academy.it</groupId>
    <artifactId>MySampleApp</artifactId>
    <packaging>pom</packaging>
    <version>1.0-SNAPSHOT</version>
    <name>MySampleApp</name>
    <url>http://maven.apache.org</url>
    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
</project>
```

MULTI-MODULES
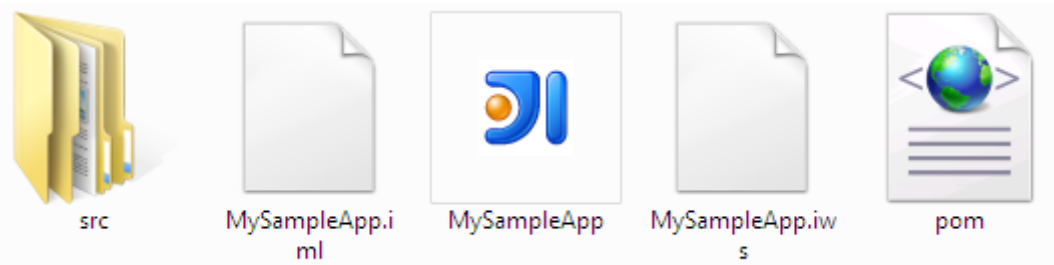
mvn archetype:generate -DgroupId=by.academy.it -DartifactId=jarModule

[INFO] Parameter: groupId, Value: by.academy.it

[INFO] Parameter: packageName, Value: by.academy.it

[INFO] Parameter: package, Value: by.academy.it

[INFO] Parameter: artifactId, Value: jarModule

[INFO] Parameter: basedir, Value: D:\projects\MySampleApp

[INFO] Parameter: version, Value: 1.0-SNAPSHOT

[INFO] project created from Old (1.x) Archetype in dir:
D:\projects\MySampleApp\jarModule

MULTI-MODULES

родительский pom

дочерний pom

```xml
<modules>
    <module>jarModule</module>
</modules>
```

```xml
<?xml version="1.0"?>
<project xsi:schemaLocation="http://maven
    xmlns:xsi="http://www.w3.org/2001/XML
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>by.academy.it</groupId>
    <artifactId>MySampleApp</artifactId>
    <version>1.0-SNAPSHOT</version>
  </parent>
  <groupId>by.academy.it</groupId>
  <artifactId>jarModule</artifactId>
  <version>1.0-SNAPSHOT</version>
  <name>jarModule</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```
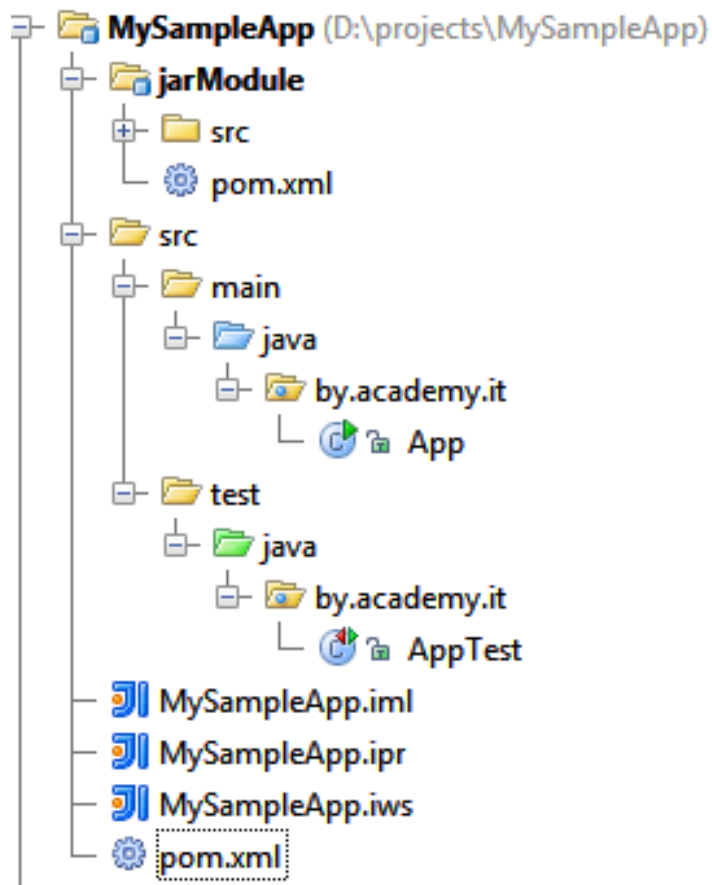
MULTI-MODULES
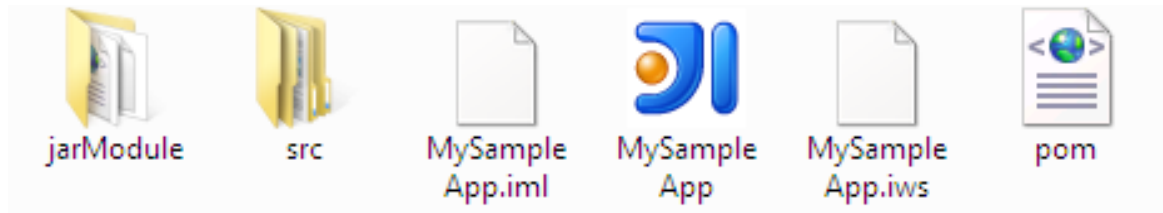
mvn archetype:generate -DarchetypeArtifactId=maven-archetype-webapp
 Parameter: groupId, Value: by.academy.it
 Parameter: packageName, Value: by.academy.it
 Parameter: package, Value: by.academy.it
 Parameter: artifactId, Value: webModule
 Parameter: basedir, Value: D:\projects\MySampleApp
 Parameter: version, Value: 1.0-SNAPSHOT
 project created from Old (1.x) Archetype in dir: D:\projects\MySampleApp\webModule

MULTI-MODULES

```
MySampleApp (D:\projects\My
  jarModule
    src
    jarModule.iml
    pom.xml
  src
    main
      java
        by.academy.it
          App
    test
      java
        by.academy.it
          AppTest
  webModule
    src
      main
        resources
        webapp
    pom.xml
```
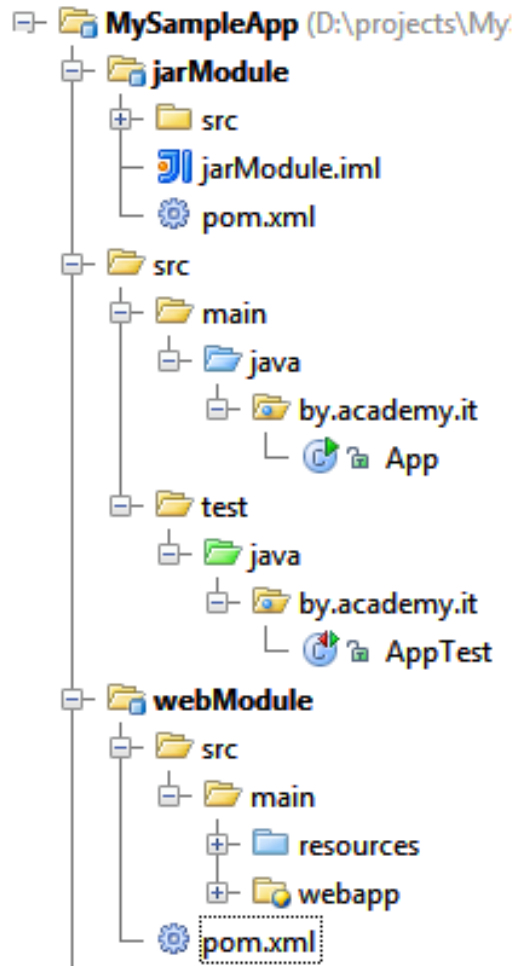
родительский pom

```xml
<modules>
    <module>jarModule</module>
    <module>webModule</module>
</modules>
```

MULTI-MODULES. BUILD

```
D:\projects\MySampleApp>mvn clean install

D:\projects\MySampleApp>set MAVEN_OPTS=-Xmx256M
[INFO] Scanning for projects...
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Build Order:
[INFO]
[INFO] MySampleApp
[INFO] jarModule
[INFO] webModule Maven Webapp
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building MySampleApp 1.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.4.1:clean (default-clean) @ MySampleApp ---
[INFO]
[INFO] --- maven-install-plugin:2.3.1:install (default-install) @ MySampleApp ---
[INFO] Installing D:\projects\MySampleApp\pom.xml to C:\Users\olga\.m2\repository\by\academy\it\MySa
mpleApp\1.0-SNAPSHOT\MySampleApp-1.0-SNAPSHOT.pom
[INFO]
```

MULTI-MODULES. BUILD


```
[INFO]
[INFO] --- maven-jar-plugin:2.3.2:jar (default-jar) @ jarModule ---
[INFO] Building jar: D:\projects\MySampleApp\jarModule\target\jarModule-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:2.3.1:install (default-install) @ jarModule ---
[INFO] Installing D:\projects\MySampleApp\jarModule\target\jarModule-1.0-SNAPSHOT.jar to C:\Users\ol
ga\.m2\repository\by\academy\it\jarModule\1.0-SNAPSHOT\jarModule-1.0-SNAPSHOT.jar
[INFO] Installing D:\projects\MySampleApp\jarModule\pom.xml to C:\Users\olga\.m2\repository\by\acade
my\it\jarModule\1.0-SNAPSHOT\jarModule-1.0-SNAPSHOT.pom
[INFO]
```



```
[INFO] --- maven-install-plugin:2.3.1:install (default-install) @ webModule ---
[INFO] Installing D:\projects\MySampleApp\webModule\target\webModule.war to C:\U
tory\by\academy\it\webModule\1.0-SNAPSHOT\webModule-1.0-SNAPSHOT.war
[INFO] Installing D:\projects\MySampleApp\webModule\pom.xml to C:\Users\olga\.m2
my\it\webModule\1.0-SNAPSHOT\webModule-1.0-SNAPSHOT.pom
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO]
[INFO] MySampleApp ........................................ SUCCESS [0.769s]
[INFO] jarModule .......................................... SUCCESS [4.698s]
[INFO] webModule Maven Webapp ............................. SUCCESS [1.743s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO]
```

MULTI-MODULES. BUILD
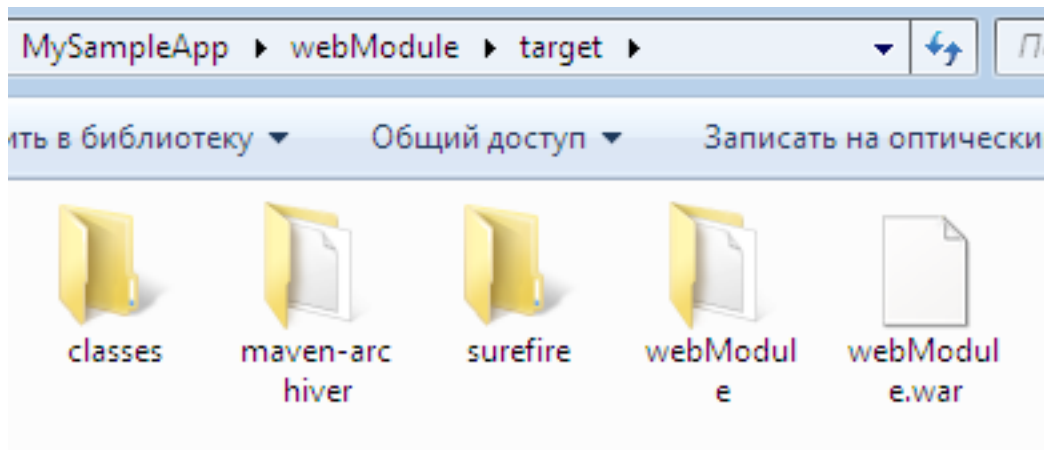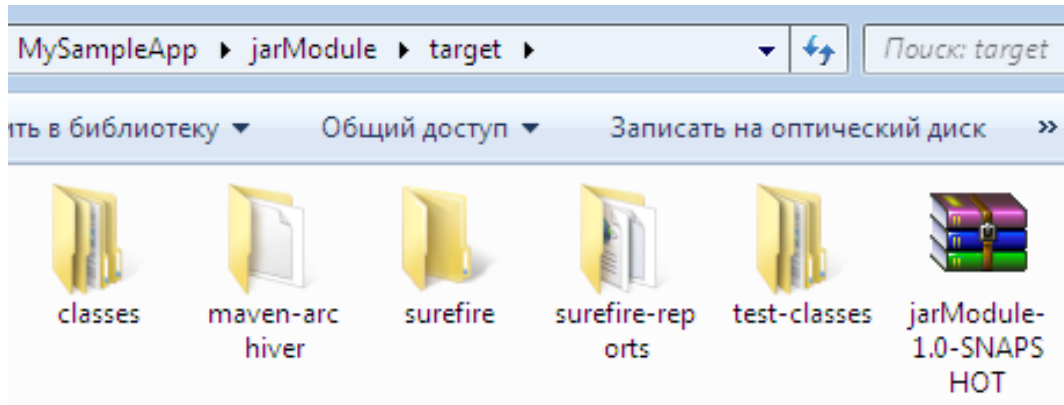
PRACTICE
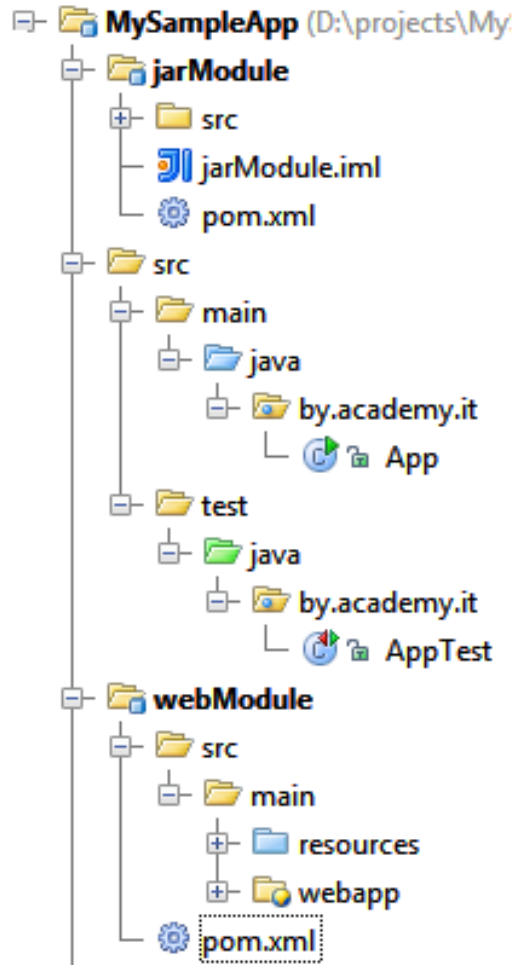
Создайте многомодульный проект. Одни модуль - jar, другой модуль – war. Создайте parent модуль. Проверьте процесс сборки и опишите ваши вопросы.

DEPENDENCY MANAGEMENT

```xml
<dependency>
   <groupId>junit</groupId>
   <artifactId>junit</artifactId>
   <version>3.8.1</version>
   <scope>test</scope>
</dependency>
```

```xml
<dependency>
    <groupId>javax</groupId>
    <artifactId>javaee-api</artifactId>
    <version>6.0</version>
    <type>jar</type>
    <scope></scope>
</dependency>
<dependency>
    <gro                    upId>
    <art          web-api</artifactId>
    <ver          on>
    <sco          ope>
</dependency>
```

☐	compile
☐	test
☐	runtime
☐	**provided**
☐	system

SAMPLE

Родительский pom.xml

```xml
<modules>
    <module>jarModule</module>
    <module>webModule</module>
</modules>
<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
</dependencyManagement>
```
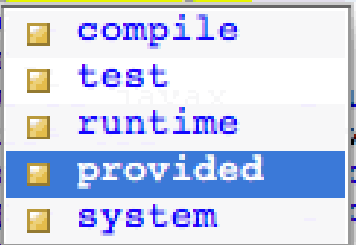
Дочерний pom.xml

```xml
<dependencies>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
    </dependency>
</dependencies>
```

SAMPLE

```
-------------------------------------------------------------------------------
Reactor Summary:

MySampleApp .............................................. SUCCESS [0.761s]
jarModule ................................................ SUCCESS [4.828s]
webModule Maven Webapp ................................... SUCCESS [1.891s]
-------------------------------------------------------------------------------
BUILD SUCCESS
-------------------------------------------------------------------------------
Total time: 7.859s
Finished at: Sat Nov 17 15:55:05 EET 2012
Final Memory: 5M/10M
-------------------------------------------------------------------------------
```

PRACTICE

Настройте зависимость многомодульного проекта на Oracle database driver. Выполните сборку проекта и проверьте, что она прошла успешно.

http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html

mvn install:install-file -Dfile=ojdbc5.jar -DgroupId=com.oracle -DartifactId=ojdbc5 -Dversion=11.2.0.3.0 -Dpackaging=jar -DgeneratePom=true

```
<dependency>
        <groupId>com.oracle</groupId>
        <artifactId>ojdbc5</artifactId>
        <version>11.2.0.3.0</version>
</dependency>
```

MULTI-MODULE PROJECTS

- The POM hierarchy, super POM, and parent POM
- Extending and overriding POM files
- Maven coordinates
- Managing dependencies
- Transitive dependencies
- Dependency scopes and optional dependencies

MAVEN POM

```xml
<project>
  <parent>...</parent>
  <modelVersion>4.0.0</modelVersion>
  <groupId>...</groupId>
  <artifactId>...</artifactId>
  <version>...</version>
  <packaging>...</packaging>

   <name>...</name>
  <description>...</description>
  <url>...</url>
  <inceptionYear>...</inceptionYear>
  <licenses>...</licenses>
  <organization>...</organization>
  <developers>...</developers>
  <contributors>...</contributors>

  <dependencies>...</dependencies>
  <dependencyManagement>...</dependencyManagement>
  <modules>...</modules>
  <properties>...</properties>

  <build>...</build>
  <reporting>...</reporting>

  <issueManagement>...</issueManagement>
  <ciManagement>...</ciManagement>
  <mailingLists>...</mailingLists>
  <scm>...</scm>
  <prerequisites>...</prerequisites>

  <repositories>...</repositories>
  <pluginRepositories>...</pluginRepositories>

  <distributionManagement>...</distributionManagement>

  <profiles>...</profiles>
</project>
```

SUPER POM

EFFECTIVE POM

- mvn help:effective-pom

```xml
<repositories>
  <repository>
    <releases>
      <enabled>true</enabled>
      <updatePolicy>daily</updatePolicy>
      <checksumPolicy>ignore</checksumPolicy>
    </releases>
    <id>wso2-nexus</id>
    <name>WSO2 internal Repository</name>
    <url>http://maven.wso2.org/nexus/content/groups/wso2-public/</url>
  </repository>
  <repository>
    <snapshots>
      <enabled>false</enabled>
    </snapshots>
    <id>central</id>
    <name>Central Repository</name>
    <url>http://repo.maven.apache.org/maven2</url>
  </repository>
</repositories>
```
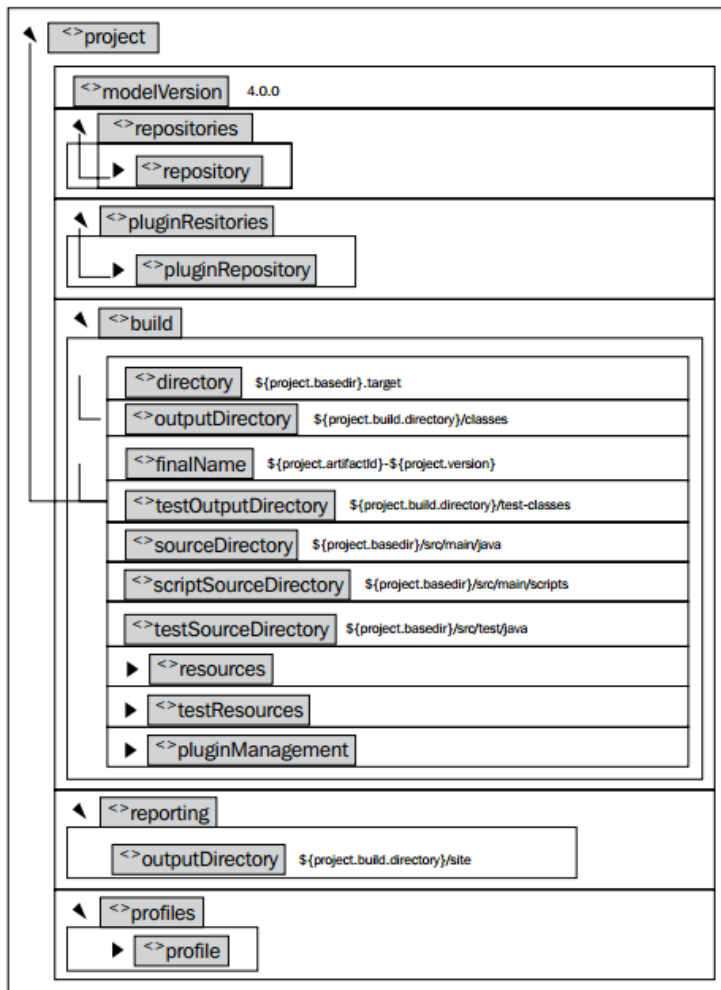
MAVEN COORDINATES

- A valid Maven POM file must have groupId, artifactId, and version. The groupId and the version elements can also be inherited from the parent POM file.

```
<plugin>
    <artifactId>maven-antrun-plugin</artifactId>
    <version>1.3</version>
</plugin>
```

MAVEN COORDINATES

settings.xml

```
<!-- pluginGroups
 | This is a list of additional group identifiers that
 | will be searched when resolving plugins by their prefix, i.e.
 | when invoking a command line like "mvn prefix:goal".
 | Maven will automatically add the group identifiers
 | "org.apache.maven.plugins" and "org.codehaus.mojo"
 | if these are not already contained in the list.
 |-->
<pluginGroups>
  <!-- pluginGroup
    | Specifies a further group identifier to use for plugin
    | lookup.
  <pluginGroup>com.your.plugins</pluginGroup>
  -->
</pluginGroups>
```

DEPENDENCY MANAGEMENT

- Mechanism to centralize dependency information

- Have all information about dependencies in the parent POM (version, exclusions, scope, type)

- Child POMs only have to have a simple reference to the dependencies, excluding version, exclusions, scope type.

CHILD POM

```xml
<dependencies>
    <dependency>
        <groupId>org.apache.axis2.wso2</groupId>
        <artifactId>axis2</artifactId>
    </dependency>
    <dependency>
        <groupId>org.apache.ws.commons.axiom.wso2</groupId>
        <artifactId>axiom</artifactId>
    </dependency>
</dependencies>
```

Here, you will see only groupId and artifactId of a given dependency but **not version**.

UMBRELLA PROJECTS (DEPENDENCY GROUPING)

```xml
<dependency>
    <groupId>org.apache.axis2</groupId>
    <artifactId>axis2-kernel</artifactId>
    <version>1.6.2</version>
</dependency>
<dependency>
    <groupId>org.apache.axis2</groupId>
    <artifactId>axis2-adb</artifactId>
    <version>1.6.2</version>
</dependency>
<dependency>
    <groupId>org.apache.axis2</groupId>
    <artifactId>axis2-transport-http</artifactId>
    <version>1.6.2</version>
</dependency>
<dependency>
    <groupId>org.apache.axis2</groupId>
    <artifactId>axis2-transport-local</artifactId>
    <version>1.6.2</version>
</dependency>
<dependency>
    <groupId>org.apache.axis2</groupId>
    <artifactId>axis2-xmlbeans</artifactId>
    <version>1.6.2</version>
</dependency>
```

If you have multiple Axis2 client modules, in each module, you need to duplicate all these dependencies:

UMBRELLA PROJECTS (DEPENDENCY GROUPING)

```xml
<project>
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.epam</groupId>
    <artifactId>axis2-client</artifactId>
    <version>1.0.0</version>
    <packaging>pom</packaging>
    <dependencies>
        <dependency>
            <groupId>org.apache.axis2</groupId>
            <artifactId>axis2-kernel</artifactId>
            <version>1.6.2</version>
        </dependency>
        <dependency>
            <groupId>org.apache.axis2</groupId>
            <artifactId>axis2-adb</artifactId>
            <version>1.6.2</version>
        </dependency>
        <dependency>
            <groupId>org.apache.axis2</groupId>
            <artifactId>axis2-transport-http</artifactId>
            <version>1.6.2</version>
        </dependency>
        <dependency>
            <groupId>org.apache.axis2</groupId>
            <artifactId>axis2-transport-local</artifactId>
            <version>1.6.2</version>
        </dependency>
        <dependency>
            <groupId>org.apache.axis2</groupId>
            <artifactId>axis2-xmlbeans</artifactId>
            <version>1.6.2</version>
        </dependency>
    </dependencies>
</project>
```

```xml
<project>
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.epam.maven</groupId>
    <artifactId>my-axis2-client</artifactId>
    <version>1.0.0</version>
    <dependencies>
        <dependency>
        <groupId>com.epam.maven</groupId>
        <artifactId>axis2-client</artifactId>
        <version>1.0.0</version>
        </dependency>
    </dependencies>
</project>
```

TRANSITIVE DEPENDENCIES
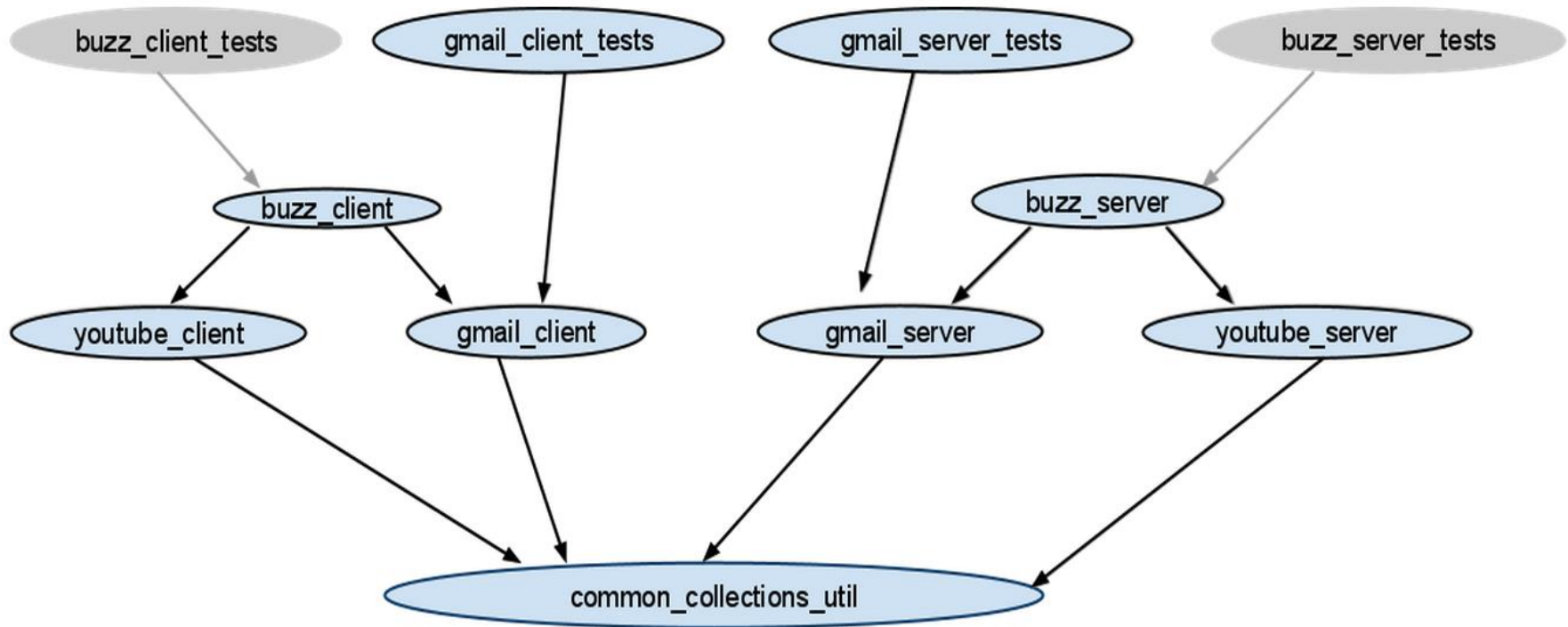
OPTIONAL DEPENDENCY

- The idea is that some of the dependencies are only used for certain features in the project, and will not be needed if that feature isn't used. Ideally

- **Project-A -> Project-B**

```xml
<project>
    ...
    <dependencies>
        <!-- declare the dependency to be set as optional -->
        <dependency>
            <groupId>sample.Projects</groupId>
            <artifactId>Project-B</artifactId>
            <version>1.0</version>
            <scope>compile</scope>
            <optional>true</optional>
            <!-- value will be true or false only -->
        </dependency>
    </dependencies>
</project>
```

- Project-X -> Project-A
  - You'll notice that Project-B is not included in the classpath of Project-X; you will need to declare it directly in your POM in order for B to be included in X's classpath

TRANSITIVE DEPENDENCIES

- Everything will work fine till one fine day, you update the version of lib, and the new version

- could have a reference to a new version of jar, which is not compatible with your code. This could easily break your build or might cause test cases to fail.

TRANSITIVE DEPENDENCIES

```
$ mvn dependency:analyze

[INFO] --- maven-dependency-plugin:2.8:analyze (default-cli) @
  jose ---

[WARNING] Used undeclared dependencies found:

[WARNING] net.minidev:json-smart:jar:1.1.1:compile
```

AVAILABLE SCOPES

- 6 Available Scopes
  - Compile
  - Provided
  - Runtime
  - Test
  - System
  - Import (Maven 2.0.9 onwards)

EXCLUDED DEPENDENCIES

```xml
<dependencies>
    <dependency>
        <groupId>GroupX</groupId>
        <artifactId>ArtifactX</artifactId>
        <exclusions>
            <exclusion>
                <groupId>GroupZ</groupId>
                <artifactId>ArtifactZ</artifactId>
            </exclusion>
        </exclusions>
    </dependency>
</dependencies>
```

PARALLEL BUILDS IN MAVEN 3

Maven 3.x has the capability to perform parallel builds. The command is as follows:

- mvn -T 4 clean install # Builds with 4 threads
  mvn -T 1C clean install # 1 thread per cpu core
  mvn -T 1.5C clean install # 1.5 thread per cpu core

- This build-mode analyzes your project's dependency graph and schedules modules that can be built in parallel according to the dependency graph of your project.

PARALLEL BUILDS IN MAVEN 3
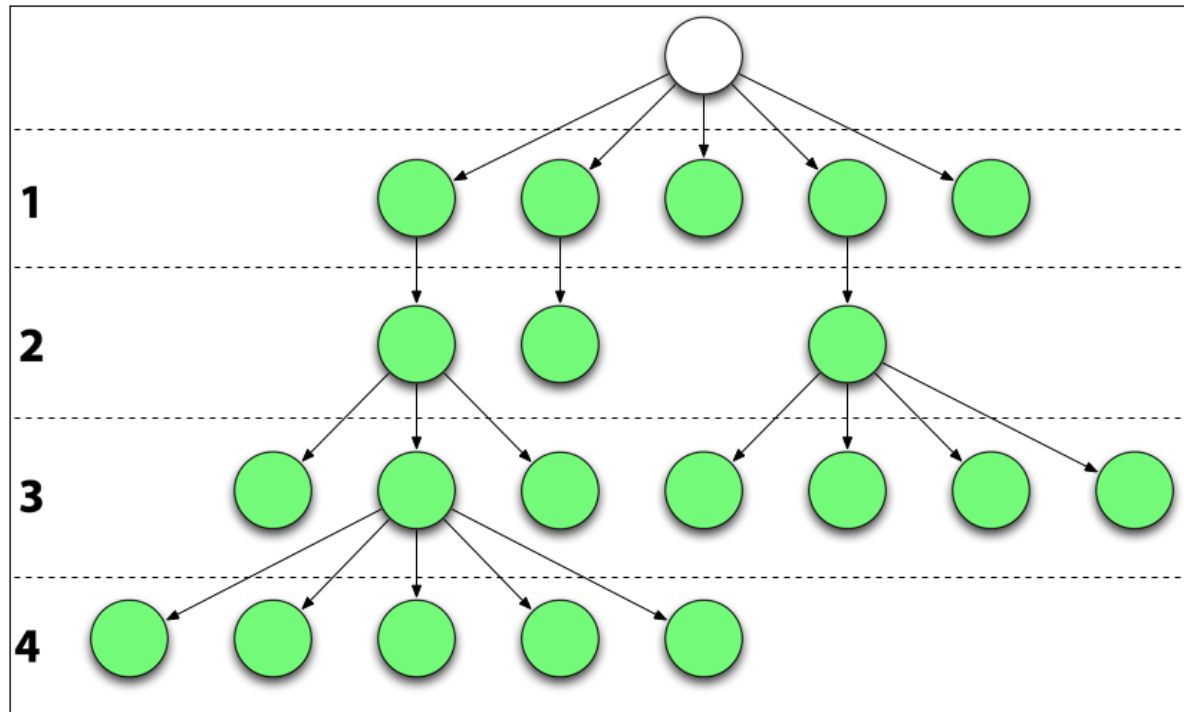
- On level 2 you'll be running 3 parallel modules, and 7 on 3, 5 on level 4.

- The parallel build feature rewards "correct" modularizations. If your project has degenerated inter-module dependencies (execessive dependencies inside reactor), you will probably see gains by cleaning up the dependencies.

MAVEN RELEASES HISTORY

Maven 3

Release Date	Version	Required Java Version	Links
2015-11-14	**3.3.9**	Java 7	**announce , release notes , reference documentation**
2015-04-28	3.3.3		announce , release notes , reference documentation
2015-03-18	3.3.1		announce , release notes , reference documentation
2014-12-20	**3.2.5**	Java 6	**announce , release notes , reference documentation**
2014-08-17	3.2.3		announce , release notes , reference documentation
2014-06-26	3.2.2		announce , release notes , reference documentation
2014-02-21	3.2.1		announce , release notes , reference documentation
2013-10-04	**3.1.1**	Java 5	**announce , release notes , reference documentation**
2013-07-15	3.1.0		announce , release notes , reference documentation

Q&A