

Object-Oriented Programming in

Guillaume Muller

based on work from:

Ch. Gravier, F. Laforest, J. Subercaze

Télécom Saint-Étienne

`{pénom.nom}@univ-st-etienne.fr`

14 September 2020

- Why? Eclipse fine, but does not transfer local config!

¹<https://dzone.com/refcardz/apache-maven-2>

- Why? Eclipse fine, but does not transfer local config!
- Goal: Manage compilation chain & dependencies

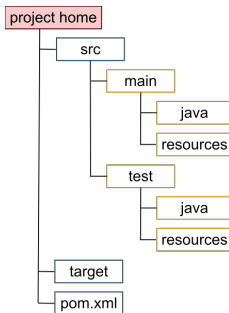
¹<https://dzone.com/refcardz/apache-maven-2>

Maven¹

- Why? Eclipse fine, but does not transfer local config!
- Goal: Manage compilation chain & dependencies
- Uses conventions

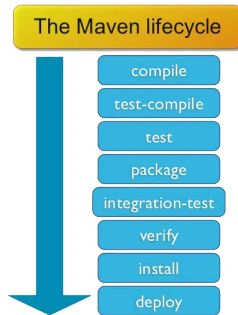
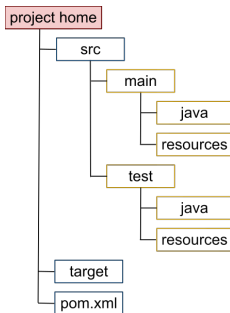
¹<https://dzone.com/refcardz/apache-maven-2>

- Why? Eclipse fine, but does not transfer local config!
- Goal: Manage compilation chain & dependencies
- Uses conventions
 - Project architecture



¹<https://dzone.com/refcardz/apache-maven-2>

- Why? Eclipse fine, but does not transfer local config!
- Goal: Manage compilation chain & dependencies
- Uses conventions
 - Project architecture
 - Pre-configured rules: clean, compile...



¹<https://dzone.com/refcardz/apache-maven-2>

- Why? Eclipse fine, but does not transfer local config!
- Goal: Manage compilation chain & dependencies
- Uses conventions
- Configuration via `pom.xml` file

¹<https://dzone.com/refcardz/apache-maven-2>

- Why? Eclipse fine, but does not transfer local config!
- Goal: Manage compilation chain & dependencies
- Uses conventions
- Configuration via `pom.xml` file
- Example usage:
`mvn clean compile`
- We will let Eclipse do (most of) that for us!!

¹<https://dzone.com/refcardz/apache-maven-2>

Maven – Example pom.xml file

```
1  <project>
2    <!-- model version is always 4.0.0 for Maven 2.x POMs -->
3    <modelVersion>4.0.0</modelVersion>
4    <!-- project coordinates, i.e. a group of values which uniquely identify this project -->
5    <groupId>com.mycompany.app</groupId>
6    <artifactId>my-app</artifactId>
7    <version>1.0</version>
8    <!-- library dependencies -->
9    <dependencies>
10      <dependency>
11        <!-- coordinates of the required library -->
12        <groupId>junit</groupId>
13        <artifactId>junit</artifactId>
14        <version>3.8.1</version>
15        <!-- this dependency is only used for running and compiling tests -->
16        <scope>test</scope>
17      </dependency>
18    </dependencies>
19  </project>
```

- archetype for simple JAvA project =
maven-archetype-quickstart
- `groupId` \approx package
- `artifactId` \approx main class