ID1020 Introduktion av Maven

Jim Dowling
jdowling@kth.se

Vad använder ni för byggverktyg?

- Inget
- En IDE
  - NetBeans, Eclipse, IntelliJ, osv.
- Ant
- Maven
- Gradle*

*Gradle har en högre inlärningströskel och svagare IDE stöd.

Maven

- Att bygga ett Java projekt innehåller några av följande steg:
  - Kompilera källkod
  - Kopiera resurser
  - Kompilera och köra tests
  - Paktera ihop projektet (t.ex. i en jar fil)
  - Deploy (t.ex. ladda upp jar filen på en server)
  - Clean

Apache Maven

- Maven är ett byggverktyg till Java
  - Med bra stöd i Netbeans och Intelli-J

- Maven beskriver ett projekt och konfigurera ett bygge
  - Du behöver inte skriva kod (bara xml)
  - Du konfigurerar plugins för att skrädda sy ditt bygge

Apache Maven

- De grundläggande idéerna med Maven är:
  - konvention hellre än konfiguration
    - T.ex., all källkod filer finns under ”/src”
    - T.ex., alla kompilerade klassfiler finns under ”/target”
  - metadatahantering via POM
  - livscykel för bygge
  - beroendehantering
  - kan byggas ut med Plugins

Maven Project Object Model (POM)

- En **pom.xml** fil beskriver ett projekt.
- En pom.xml måste innehålla:
  - GroupID
  - ArtifactID
  - Version
- En pom.xml kan innehålla:
  - Förälder projekt (Parent POM)
  - Artifakt-Typ
    - T.ex., jarfil
  - Beroenden
  - Plugins
  - Profiler

Minimal Projekt Beskrivning i en pom.xml

- Varje maven projekt behöver:

groupId	≈ packagenamn i Java
artfiactId	≈ klassnamn i Java
version	{Major}.{Minor}.{Maintanence}

```
<?xml version="1.0" encoding="UTF-8"?>
<project>
    <groupId>se.kth.id1020</groupId>
    <artifactId>lab1</artifactId>
    <version>1.0</version>
</project>
```

Beroenden (Dependencies)

- Ett beroende är ett annat maven projekt som ditt maven projekt behöver för att kompileras/exekveras
- Beroenden är transitiva
- När du bygger ett maven projekt, jar filer från beroenden laddas hem och som sparas lokalt i
  ${user.home}/.m2/repository

Inga classpath helvete längre!

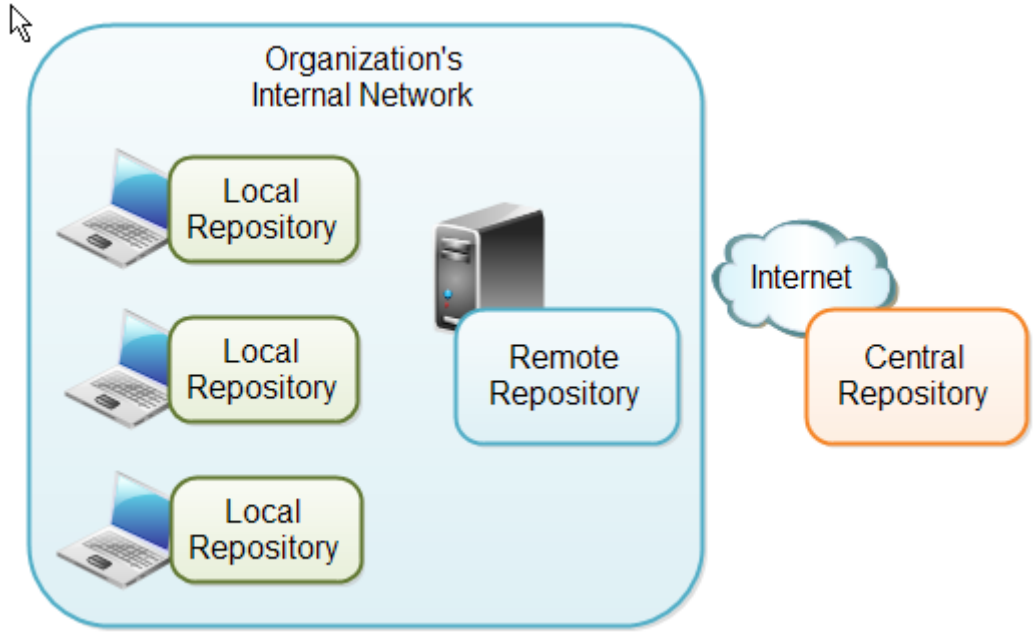
Beroenden som behövs för ID1020

```
<dependencies>
        <dependency>
                <groupId>edu.princeton.cs.introcs</groupId>
                <artifactId>algs4-package</artifactId>
                <version>1.0</version>
        </dependency>
        <dependency>
                <groupId>edu.princeton.cs.introcs</groupId>
                <artifactId>stdlib-package</artifactId>
                <version>1.0</version>
        </dependency>
</ dependencies >
```

Beroenden kommer från Repositories

- Lokal repository
  ${user.home}/.m2/repository

- Maven Central alltid används som en repository
  http://repo1.maven.org/maven2



- Vissa beroenden finns inte på Maven Central
  - Som biblioteken (jar filer) som används i boken

- Man kan också använda tredje part repositories
  - Lägg till en <repository> tag i pom.xml
    http://kompics.sics.se/maven/repository

Trejde part Repository

```
<repositories>
        <repository>
                <id>sics-release</id>
                <name>SICS Release Repository</name>
                <url>http://kompics.sics.se/maven/repository</url>
        </repository>
</repositories>
```

Maven plugins

- Plugins används för allt annat* i maven, t.ex.
  - Specificera vilken kompilator att använda (jdk 1.6/1.7/1.8)
  - Exekvera ett maven projekt
    - Man kan också exekvera ett maven projekt direkt från IDE:n
  - Generera en ueber-jar fil

- Mer info
  - http://maven.apache.org/plugins/

*Mavens kärn kommando (clean, compile, package, osv) är inbyggda plugins

T.ex., bygga en ”uber jar” med Maven

- Vi ska använda Apache Maven Shade plugin
  - http://maven.apache.org/plugins/maven-shade-plugin/
- Ändringar i pom filen:

```
<build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-shade-plugin</artifactId>
                <version>1.5</version>
                .........
            </plugin>
        </plugins>
    </build>
```

- Du måste anropa en plugin för att använda den:
  >mvn shade:shade

Maven Shade plugin inlägget i pom.xml

```
<build>
  <plugins>
    <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-shade-plugin</artifactId>
        <version>1.5</version>
        <executions>
          <execution>
            <phase>package</phase>
            <goals>
                <goal>shade</goal>
            </goals>
            <configuration>
                <transformers>
                  <transformer
implementation="org.apache.maven.plugins.shade.resource.ManifestResourceTransfo
rmer">
                        <mainClass>se.kth.id1020.Recursion</mainClass>
                  </transformer>
                </transformers>
            </configuration>
          </execution>
        </executions>
    </plugin>
  </plugins>
</build>
```

Vad gör man med en uber jar?

- Exekvera jar filen:

```
>java –jar target/myProject-shade.jar
```

- Exekvera jar filen med argument

```
>java –jar target/myProject-shade.jar arg1
arg2 arg3
```

- Exekvera jar filen med en annan Main klass (om ingen Main Class är definierad i Shade plugin)

```
>java –jar target/myProject-shade.jar MainKlass
```

Minimal pom.xml for Lab1 in ID1020 (1/4)

```
<?xml version="1.0" encoding="UTF-8"?>
<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>com.mycompany</groupId>
<artifactId>algs4-test</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>jar</packaging>
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <maven.compiler.source>1.7</maven.compiler.source>
  <maven.compiler.target>1.7</maven.compiler.target>
</properties>
<dependencies>
        <dependency>
            <groupId>edu.princeton.cs.introcs</groupId>
            <artifactId>algs4-package</artifactId>
            <version>1.0</version>
        </dependency>
```

Minimal pom.xml for Lab1 in ID1020 (2/4)

```
        <dependency>
            <groupId>edu.princeton.cs.introcs</groupId>
            <artifactId>stdlib-package</artifactId>
            <version>1.0</version>
        </dependency>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.10</version>
            <scope>test</scope>
        </dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-shade-plugin</artifactId>
        <version>1.5</version>
        <executions>
          <execution>
            <phase>package</phase>
```

Minimal pom.xml for Lab1 in ID1020 (3/4)

```
            <goals>
                <goal>shade</goal>
            </goals>
            <configuration>
                <transformers>
                  <transformer
implementation="org.apache.maven.plugins.shade.resource.ManifestResourc
eTransformer">
                        <mainClass>se.kth.id1020.Recursion</mainClass>
                  </transformer>
                </transformers>
            </configuration>
          </execution>
        </executions>
    </plugin>
  </plugins>
</build>
```

Minimal pom.xml for Lab1 in ID1020 (4/4)

```
<repositories>
        <repository>
            <id>sics-release</id>
            <name>SICS Release Repository</name>
            <url>http://kompics.sics.se/maven/repository</url>
        </repository>
</repositories>
</project>
```

Viktigaste Maven commando för ID1020

- mvn clean
- mvn install
- Man kan länka ihop commando
  - mvn clean install
- Andra commando
  - mvn compile
  - mvn package
  - mvn test
  - mvn site

Demo med Netbeans

Bibliotek med Kursboken

- I boken, använder vi `StdIn` och `StdOut` klassen för att läsa och skriva siffror till och från stdin och stdout (dvs., skärmen, filer, osv.)

```
int x = StdIn.readInt();
StdOut.println("value read was: " + x);
```

- `Stopwatch` kan användas för att mäta hur mycket tid har gått.

```
Stopwatch sw = new Stopwatch();
// göra ngt
System.out.println(sw.elapsedTime());
```

Tiden sedan "sw" skapades

Referenser

- http://maven.apache.org
- http://tutorials.jenkov.com/maven/maven-tutorial.html
- Bokens standard library bibliotek: http://introcs.cs.princeton.edu/java/stdlib/
- Bokens java algoritmer bibliotek: http://algs4.cs.princeton.edu/code/