

深圳大学实验报告

课程名称： 软件工程

实验项目名称： 实验 4 模块过程设计

学院： 计算机与软件学院

专业： 软件工程（腾班）

指导教师： 卢亚辉

报告人： 黄亮铭 学号： 2022155028 班级： 腾班

实验时间： 2022 年 10 月 17 日（周四）~2021 年 10 月 29 日（周日）

实验报告提交时间： 2021 年 10 月 29 日

教务部制

一、实验目的：

- (1) 了解项目管理和构建工具方法
- (2) 掌握 Maven 的使用方法

二、实验要求：

- (1) 掌握 Maven 的使用方法
- (2) 使用 Maven 进行项目依赖管理
- (3) 使用 Maven 发布库

三、实验过程

根据下面的网址，给出详细的实验过程，要有图和文字说明。

- (1) 安装 maven

<https://liaoxuefeng.com/books/java/maven/basic/index.html>

- (2) 设置 Maven 镜像

<https://liaoxuefeng.com/books/java/maven/dependency/index.html>

- (3) 使用 Idea 或者 Vscode 或者 Eclipse 导入 maven-hello 工程，编译并运行

<https://liaoxuefeng.com/books/java/maven/dependency/index.html>

- (4) 使用 maven-shade-plugin 创建可执行 jar。

<https://liaoxuefeng.com/books/java/maven/plugin/index.html>

- (5) (选做) 使用 Maven 发布一个 Artifact 发布到本地，然后引用这个 Artifact

<https://liaoxuefeng.com/books/java/maven/dependency/index.html>

3.1 安装 maven

1. 从 [Maven 官网](#) 下载最新的 Maven 3.9.9，然后在本地解压到期望的目录中。

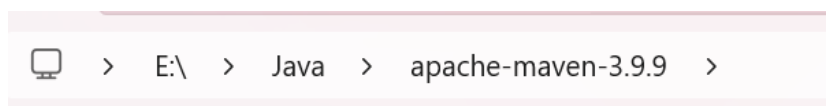


图 1：解压路径

2. 设置环境变量：在系统环境变量中添加 E:\Java\apache-maven-3.9.9\bin。

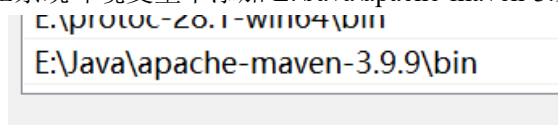


图 2：环境变量

3. 验证 maven 是否安装成功：在任意目录下打开 cmd，输入命令 `mvn -version`。结果如图 3 所示。

```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [版本 10.0.22631.4169]
(c) Microsoft Corporation。保留所有权利。

C:\Users\黄亮铭>mvn -version
Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfcdc97d260186937)
Maven home: E:\Java\apache-maven-3.9.9
Java version: 20.0.2, vendor: Oracle Corporation, runtime: E:\Java\JDK20
Default locale: zh_CN, platform encoding: UTF-8
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"
```

图 3：验证安装是否成功

3.2 设置 Maven 镜像

在 Maven/conf 目录下找到 settings.xml 文件，在文件中找到 mirror，修改相应配置为如下图所示。

```
<mirrors>
  <!-- mirror
  | Specifies a repository mirror site to use instead of a given repository. The repository that
  | this mirror serves has an ID that matches the mirrorOf element of this mirror. IDs are used
  | for inheritance and direct lookup purposes, and must be unique across the set of mirrors.
  -->
  <mirror>
    <id>mirrorId</id>
    <mirrorOf>repositoryId</mirrorOf>
    <name>Human Readable Name for this Mirror.</name>
    <url>http://my.repository.com/repo/path</url>
  </mirror>
  -->
  <!-- mirror
  <mirror>
    <id>maven-default-http-blocker</id>
    <mirrorOf>external:http:*/</mirrorOf>
    <name>Pseudo repository to mirror external repositories initially using HTTP.</name>
    <url>http://0.0.0.0/</url>
    <blocked>true</blocked>
  </mirror>
  -->
  <mirror>
    <id>aliyun</id>
    <name>aliyun</name>
    <mirrorOf>central</mirrorOf>
    <!-- 国内推荐阿里云的Maven镜像 -->
    <url>https://maven.aliyun.com/repository/central</url>
  </mirror>
</mirrors>
```

图 4：修改后的配置

3.3 导入 maven-hello 工程，编译并运行

导入 maven-hello 工程：这里我选择使用 Eclipse 导入 maven-hello 工程。在 Eclipse 中依次选择 File-》New-》New Maven Project，然后根据实验文档给出的链接中的文档填写相关信息。成功导入后项目的目录如图 5 所示。

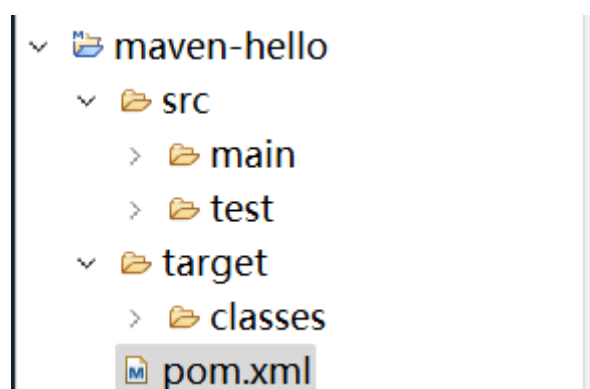


图 5：项目目录

编译并运行：在 pom.xml 所在的目录下输入命令：`mvn clean package`。

```
E:\Java\Eclipse\eclipseWorkPlace\maven-hello>mvn clean package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< software:maven-hello >-----
[INFO] Building maven-hello 0.0.1-SNAPSHOT
[INFO]    from pom.xml
[INFO] -----[ jar ]-----
.
.jar (6.8 MB at 6.4 MB/s)
[INFO] Building jar: E:\Java\Eclipse\eclipseWorkPlace\maven-hello\target\maven-he
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 47.440 s
[INFO] Finished at: 2024-10-13T21:48:01+08:00
[INFO]
```

图 6：运行成功

如果成功，则会在 target 目录下获取编译后自动打包的 jar。如图 7 所示，说明编译成功。

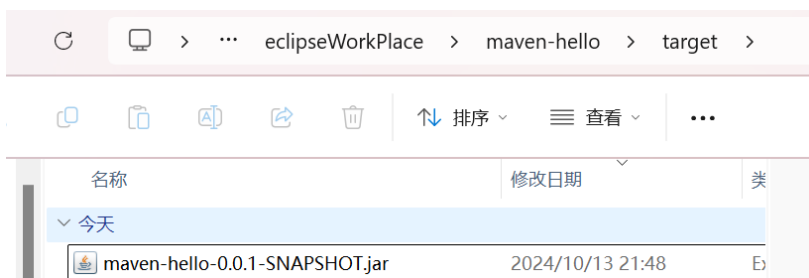


图 7：编译结果

3.4 使用 maven-shade-plugin 创建可执行 jar

```
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-shade-plugin</artifactId>
<executions>
<execution>
<phase>package</phase>
<goals>
<goal>shade</goal>
</goals>
<configuration>
<transformers>
<transformer
implementation="org.apache.maven.plugins.shade.resource.ManifestRe
<mainClass>com.itranswarp.learnjava.Main</mainClass>
</transformer>
</transformers>
</configuration>
</execution>
</executions>
</plugin>
```

图 8

在 maven-plugin 对应目录下输入命令：`mvn clean package`。结果如图 9 所示。

```
E:\Java\Eclipse\eclipseWorkPlace\maven-plugin>mvn clean package
[INFO] Scanning for projects...
[INFO]
[INFO] Replacing original artifact with shaded artifact.
[INFO] Replacing E:\Java\Eclipse\eclipseWorkPlace\maven-plugin\target\hello.jar
ven-plugin\target\hello-plugin-1.0-SNAPSHOT-shaded.jar
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 11.014 s
[INFO] Finished at: 2024-10-13T21:55:52+08:00
[INFO]
```

图 9：打包结果

在命令行中输入命令：`java -jar /path/to/your/jar`。运行可执行 jar 文件结果如下图所示。

```
C:\Users\黄亮铭>java -jar E:\Java\Eclipse\workspace\maven-plugin\target\hello.jar
22:01:40.095 [main] INFO com.itranswarp.learnjava.Main - start application...
22:01:40.117 [main] WARN com.itranswarp.learnjava.Main - begin task 1...
22:01:40.226 [main] WARN com.itranswarp.learnjava.Main - begin task 2...
22:01:40.336 [main] WARN com.itranswarp.learnjava.Main - begin task 3...
22:01:40.444 [main] WARN com.itranswarp.learnjava.Main - begin task 4...
22:01:40.555 [main] WARN com.itranswarp.learnjava.Main - begin task 5...
22:01:40.662 [main] WARN com.itranswarp.learnjava.Main - begin task 6...
22:01:40.772 [main] WARN com.itranswarp.learnjava.Main - begin task 7...
22:01:40.882 [main] WARN com.itranswarp.learnjava.Main - begin task 8...
22:01:40.991 [main] WARN com.itranswarp.learnjava.Main - begin task 9...
22:01:41.099 [main] WARN com.itranswarp.learnjava.Main - begin task 10...
22:01:41.193 [main] INFO com.itranswarp.learnjava.Main - done.
```

图 10：运行结果

通过阅读源代码，发现 jar 文件运行结果符合预期。

3.5 使用 Maven 发布一个 Artifact 发布到本地并引用

根据给定链接中的文档修改 pom.xml 文件。修改内容为增加如下图所示的内容。

```
<distributionManagement>
  <repository>
    <id>local-repo-release</id>
    <name>GitHub Release</name>
    <url>file://E:/Java/Eclipse/eclipseWorkPlace/maven-plugin/maven-repo</url>
  </repository>
</distributionManagement>

<plugin>
  <artifactId>maven-source-plugin</artifactId>
  <executions>
    <execution>
      <id>attach-sources</id>
      <phase>package</phase>
      <goals>
        <goal>jar-no-fork</goal>
      </goals>
    </execution>
  </executions>
</plugin>
<plugin>
  <artifactId>maven-javadoc-plugin</artifactId>
  <executions>
    <execution>
      <id>attach-javadocs</id>
      <phase>package</phase>
      <goals>
        <goal>jar</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

图 11：修改部分

在 Maven 项目的根目录添加文件夹 maven-repo 作为 Maven 本地文件仓库。

然后在项目根目录运行命令：`mvn clean deploy`。运行结果如下图所示。

```
E:\Java\Eclipse\eclipseWorkPlace\maven-plugin>mvn clean deploy
[INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model for
r:1.0-SNAPSHOT
[WARNING] 'build.plugins.plugin.version' for org.apache.maven.plugins:maven-shad
2
[WARNING] 'build.plugins.plugin.version' for org.apache.maven.plugins:maven-sour
21
[WARNING] 'build.plugins.plugin.version' for org.apache.maven.plugins:maven-java
21
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threaten t
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support buildin
[WARNING]
[INFO] -----< com.itranswarp.learnjava:hello-plugin >-----
[INFO] Building hello 1.0-SNAPSHOT
[INFO] from pom.xml
[INFO]
[INFO] -----< com.itranswarp.learnjava:hello-plugin >-----
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 3.572 s
[INFO] Finished at: 2024-10-13T22:16:52+08:00
[INFO]
E:\Java\Eclipse\eclipseWorkPlace\maven-plugin>
```

图 12: 运行结果

如果需要引用这个包，我们只需要在某个 Maven 目录的 pom.xml 文件添加如下依赖：

```
<dependency>
  <groupId>com.itranswarp.learnjava</groupId>
  <artifactId>hello-plugin</artifactId>
  <version>1.0-SNAPSHOT</version>
</dependency>
```

图 13: 添加的依赖

四、实验结论

1. 通过本次实验，我深入理解了项目管理和构建工具的重要性，并且掌握了 Maven 这一强大的项目管理工具的使用方法。
2. 在使用 maven-shade-plugin 创建可执行 jar 时，我遇到了打包后运行出错的问题，通过仔细检查代码和依赖，发现并修复了导致错误的依赖冲突。
3. 实验结果符合预期，我能够成功地使用 Maven 进行项目的构建、依赖管理和发布。
4. 通过本次实验，我学会了如何使用 Maven 进行项目构建和依赖管理。
5. 通过本次实验，我学会了如何将项目发布到本地仓库，并在其他项目中引用这些项目，这对于模块化开发和代码重用重要。

深圳大学学生实验报告用纸

指导教师批阅意见：

成绩评定：

指导教师签字：

2021 年 月 日

备注：

- 注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。