实验01 Maven项目构建基础

**【实验目的及要求】**

1. 熟悉Maven项目构建工具的基础知识；
2. 掌握Maven项目的基本结构；
3. 熟悉Maven项目的常用构建命令及生命周期管理；
4. 了解仓库及依赖管理；
5. 要求所有回答的文本格式：**五号，宋体、1.5倍行距，保留段单元格背景。**

**【实验步骤】**

**1、安装并配置Maven**

（1）从https://maven.apache.org/download.cgi站点下载最新版的Maven压缩包（如apache-maven-3.8.4-bin.zip）。

（2）将其解压至D盘根目录中，如D:\apache-maven-3.8.4。注意，存储路径尽量用英文且不应出现空格字符。

（3）新建系统环境变量M2\_HOME，值为D:\apache-maven-3.8.4。将路径D:\apache-maven-3.8.4\bin添加到path中。

（4）在D盘根目录中，新建文件夹m2\repository。

（5）将Maven安装目录（如D:\apache-maven-3.8.4\conf）中的settings.xml复制到D:\m2目录下，打开配置文件settings.xml，设置本地仓库至D:\m2\repository。

|  |
| --- |
| <localRepository>D:/m2/repository</localRepository> |

（6）启动IntelliJ Idea开发工具，在其Settings对话框中，设置Maven home directory、User settings file和localRepository的值，如图1.1所示。

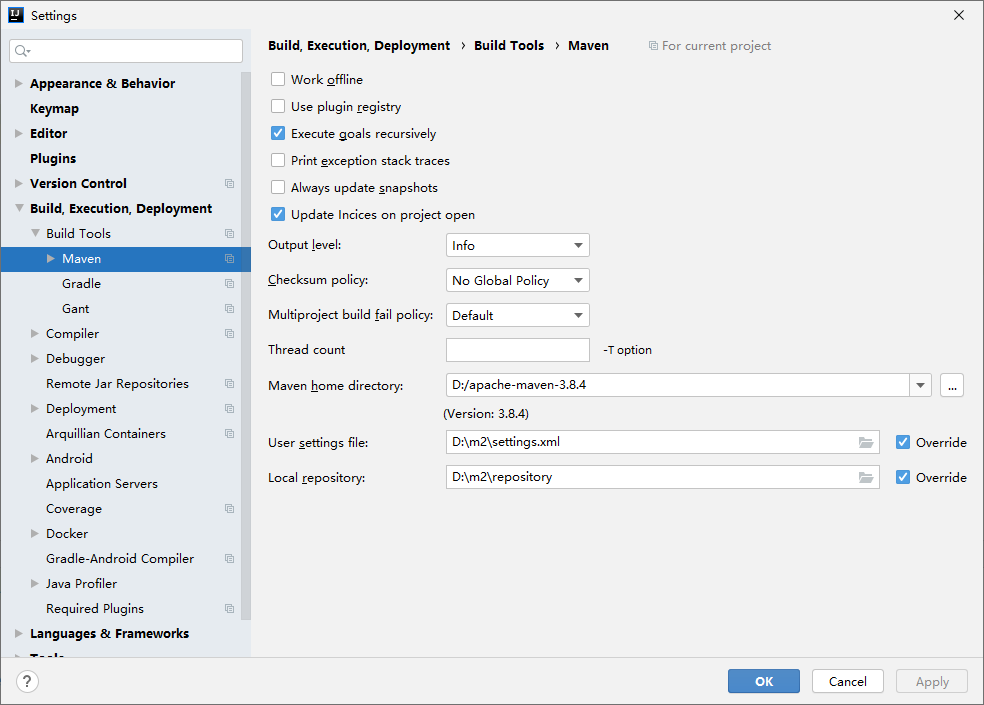


图1.1

**2、Maven项目的生命周期管理**

（1）启动IntelliJ Idea创建一个shape Maven项目，项目的坐标为：

|  |
| --- |
| <groupId>com.ex01</groupId>  <artifactId>shape</artifactId>  <version>1.0-SNAPSHOT</version> |

（2）在src\main\java目录中创建两个类，代码如下：

|  |
| --- |
| // Circle.java  package com.ex01;  public class Circle {  public double area(double radius) {  return 3.14 \* radius \* radius;  }  } |
| // Rectangle.java  package com.ex01;  public class Rectangle {  public double perimeter(double length, double width){  return 2 \* (width + length);  }  } |

（3）依次编译、打包及安装shape项目到本地仓库中。

|  |
| --- |
| // 问题1：编译情况（给出控制台的运行文本） |
| // 问题2：打包情况（给出控制台的运行文本） |
| // 问题3：安装情况（给出控制台的运行文本） |

**3、项目依赖应用**

（1）创建一个testshape Maven项目，在pom.xml文件中添加如下依赖项：

|  |
| --- |
| <?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.ex01</groupId>  <artifactId>testshape</artifactId>  <version>1.0-SNAPSHOT</version>  **<dependencies>**  **<dependency>**  **<groupId>com.ex01</groupId>**  **<artifactId>shape</artifactId>**  **<version>1.0-SNAPSHOT</version>**  **</dependency>**  **</dependencies>**  </project> |

（2）在src\main\java目录中创建主类TestApp，代码如下：

|  |
| --- |
| package com.ex01;  public class MainApp {  public static void main(String[] args) {  Circle circle = new Circle();  System.out.println("圆的面积是：" + circle.area(3.5f));  Rectangle rectangle = new Rectangle();  System.out.println("矩形的周长是：" + rectangle.perimeter(3.5f, 5.6f));  }  } |

**注意**：如果无法使用依赖项目shape中的Circle或Rectangle类，可右击项目名称，在快捷菜单中选择Run Maven→Reimport菜单项更新外部依赖的构件。

（3）执行testshape项目，查看运行结果，回答下列问题。

|  |
| --- |
| // 问题4：testshape的运行结果（给出控制台的运行文本） |
| // 问题5：shape和testshape项目之间是什么关系？ |
| // 问题6：本地仓库中shape项目的路径是什么？ |
| // 问题7：请归纳本地仓库在Maven项目构建过程中的作用？ |

4、创建一个项目[tax](javascript:;)，要求实现个人所得税计算，项目要安装到本地仓库中。个人所得税计算规则，请查询税务部门的相关网站。

|  |
| --- |
| // 问题8：项目源码（仅需包含个人所得税计算类的代码） |
| // 问题9：测试源码（给出测试类，要求提供不少于5个典型数据的测试用例） |
| // 问题10：项目的目录结构截图 |
| // 问题11：编译情况（给出控制台的运行文本） |
| // 问题12：测试情况（给出控制台的运行文本） |
| // 问题13：安装情况（给出控制台的运行文本） |

5、创建一个项目[individualtax](javascript:;)，要求调用tax中的类，完成个税的计算和输出。

|  |
| --- |
| // 问题14：列出项目的pom文件内容 |
| // 问题15：项目源码（仅需提供个税计算的类代码） |