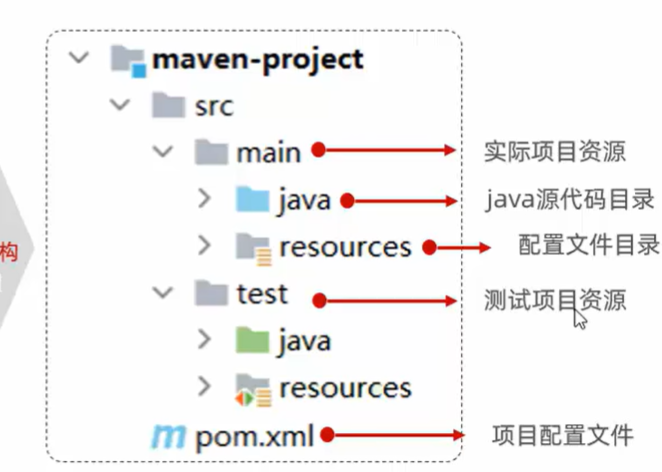
Maven

1. 作用
2. 依赖管理：对于jar包进行管理，不需要自己进行导包，只需要在Maven中输入相应的代码，Maven便可以自动进行联网下载
3. 统一项目结构：提供标准的、统一的项目结构、创建的Maven工程可以在多平台进行使用
4. 标准化项目构建流程：标准跨平台（Windows、Linux、macOS）的自动化项目构建方式
5. 标准项目流程：图示

   描述已自动生成
6. 仓库
7. 本地仓库：在自己计算机上存储的jar包
8. 中央仓库：Maven团队维护的，仓库地址：<https://repo1.maven.org/maven2/>
9. 远程仓库：由一些公司团队进行搭建
10. 查找顺序：先在本地仓库进行查找，如果本地仓库没有，加入存在远程仓库，会在远程仓库进行寻找，如果远程仓库没有，则会在中央仓库进行寻找
11. Maven的安装和配置
12. 在浏览器中找到Maven的官网下载压缩包
13. 在本地中找到一个没有中文和空格的文件夹，解压到文件夹中
14. 在bin文件夹中找到settings打开，在53行找到默认本地部署的代码，将其从注释里面复制出来，然后进行，在电脑中创建一个新的文件夹，建议在Maven文件夹中进行创建，然后将本地残酷的代码更改，本地仓库就配置好了
15. 配置阿里云私服：在浏览器中找到相关代码

<mirror>

<id>alimaven</id>

<name>aliyun maven</name>

<url>http://maven.aliyun.com/nexus/content/groups/public/</url>

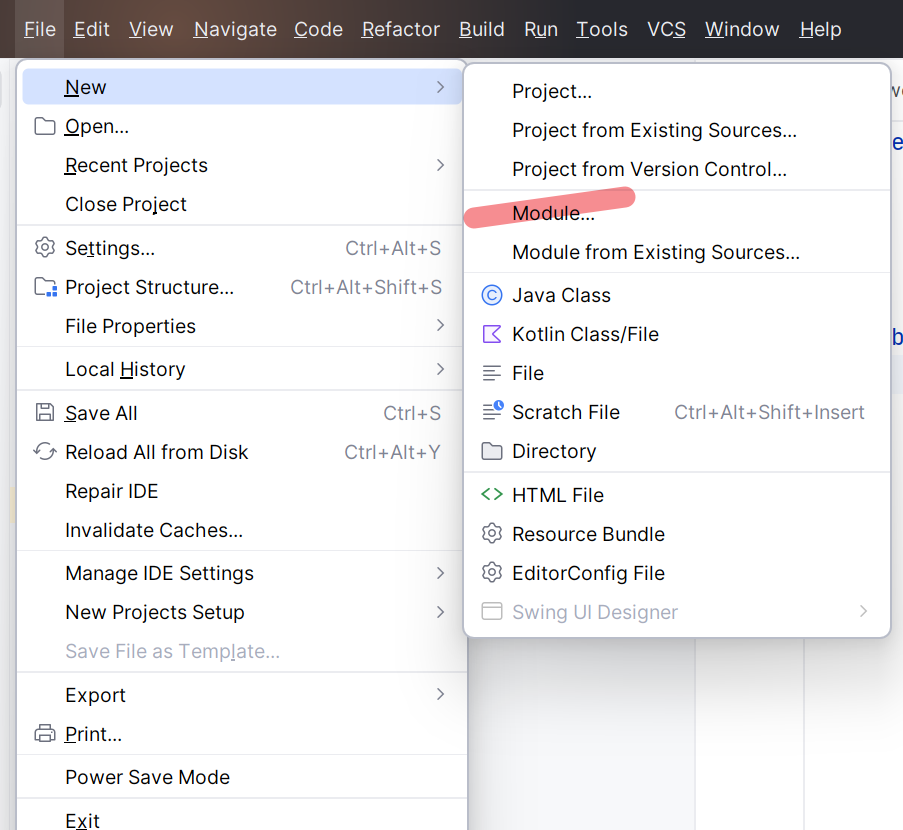
<mirrorOf>central</mirrorOf>

</mirror>

将其放在刚才settings文件夹中的mirrors中

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

1. 配置环境变量：将Maven所在的文件地址复制，打开电脑的高级设置，创建新的系统环境，在第二行中将地址放在这里，第一行的名字用英文进行起名，同时在Path环境中，添加新的路径，即%刚才创建的环境名%\bin
2. IDEA关于Maven的配置
3. 创建Maven项目：

图形用户界面, 应用程序

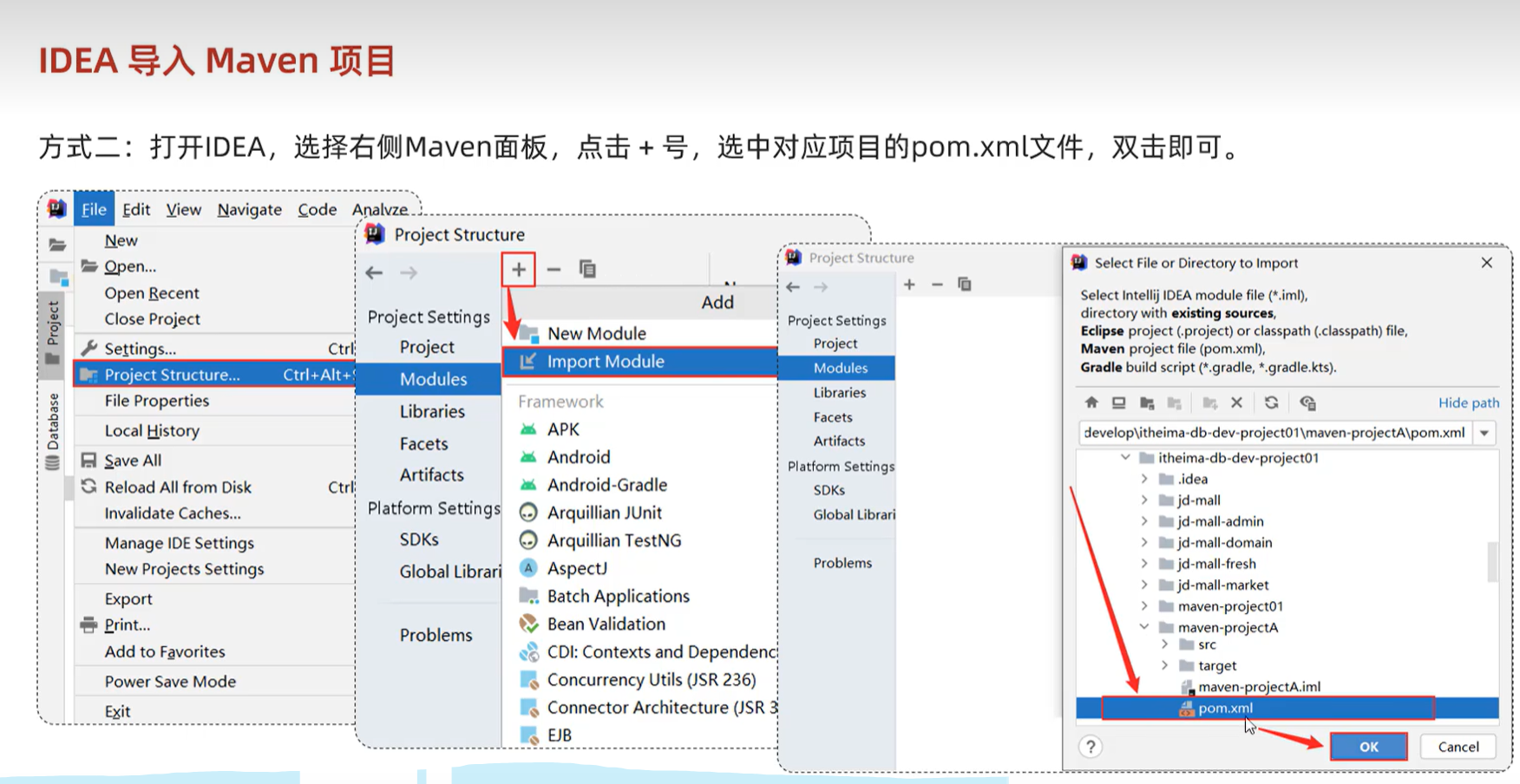
描述已自动生成

1. Maven坐标：将第三方包放在项目的pom中进行统一使用

图形用户界面, 文本

描述已自动生成

1. 导入Maven项目：



1. 在main中创建一个文件夹，名字叫resources
2. Maven坐标：Maven中的坐标是资源的唯一表示，通过该坐标可以唯一确定资源位置

使用坐标来定义项目或引入项目所需要的资料

Maven组成：groupld：通常定义Maven项目隶属的组织名称（一般是域名的反写）

Artfactld：定义项目的名称（通常是模版名称）

Version：版本号

二、依赖管理

1、添加依赖：在dependencies中的dependency，会自动生成artifactId（模块名）、groupId（组织名）

2、在artifactId中可以使用其它Maven项目

三、依赖传递

1、图示

描述已自动生成

2、间接依赖：在A项目中，可以看到直接依赖的只有一个jar包，而其它的jar包都是间接的依赖

3、排除依赖：当项目A中只是想要项目B，而不需要使用B中所以来的jar包的时候，可以排除jar包，

语法：<exclusions><exclusion>

< groupId >组织名< groupId >

<artifactId>模块名<artifactId>

<exclusion><exclusions>

4、依赖范围：

（1）定义：在默认情况下，依赖的jar包默认作用范围为全局，可以通过<scope>…<scope>进行约束

（2）作用范围：主程序范围有效（main文件）、测试程序范围内有效（test文件）、是否参与打包（package指令范围）

表格

描述已自动生成

5、生命周期

（1）种类：clean：清理工作

Default：核心工作，编译、测试、打包

Site：生成报告，发布站点

文本

中度可信度描述已自动生成

注：阶段有先后关系，按顺序运行

（2）功能介绍：clean：清理上一次构建的生成文件

Compile：编译项目源代码

Test：选择合适的单元测试框架运行测试

Package：将测试好的程序进行打包

Install：安装项目到本地仓库

1. 执行指定生命周期：单击右侧的Maven工具栏或用cmd执行命令行
2. 单元测试的依赖：junit