

* 学习目标

* 能够理解Maven相关的概念

- * 仓库管理：本地仓库，远程仓库(中心远程仓库，私服)

- * 生命周期：clean,default(compile,test,install),site

- * Maven插件

* 能够掌握Maven和idea结合开发方式

- * java项目，javaweb项目

* 能够掌握Maven多模块的开发

- * 按开发方式：MVC和Web三层架构:shopping--domain,dao,service,web,common,....

- * 按功能模块: smartcar: common,customer,employee,pay,...

* 能够掌握Maven私服的部署和使用

- * nexus

* 回顾

- * 文件上传：FileUpload，Servlet3.0，...

- * Maven

- * Apache 项目

- * 项目管理工具：项目构建，依赖的管理，项目信息的管理

- * 项目构建：clean--compile--test--package--install--deploy

- * 依赖管理：denpency-->modelVersion---groupid--artifactid--version--package--scope(compile,test,runtime,provided)

- * Maven约定目录：hellomaven--src----- main --- java

resources

target test ---

pom.xml

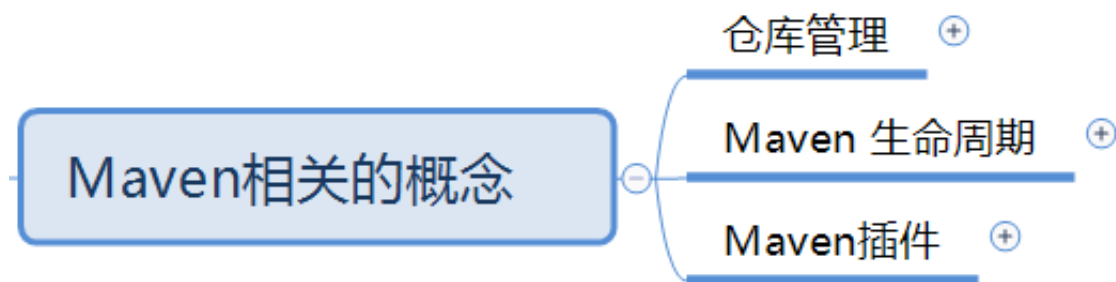
- * 如何更换本地仓库，远程中心的仓库，....

* 自动构建java项目，javaweb项目，idea构建java项目，

* 学习目标

- * 能够理解Maven相关的概念
- * 能够掌握Maven和idea结合开发方式
- * 能够掌握Maven多模块的开发
- * 能够掌握Maven私服的部署和使用

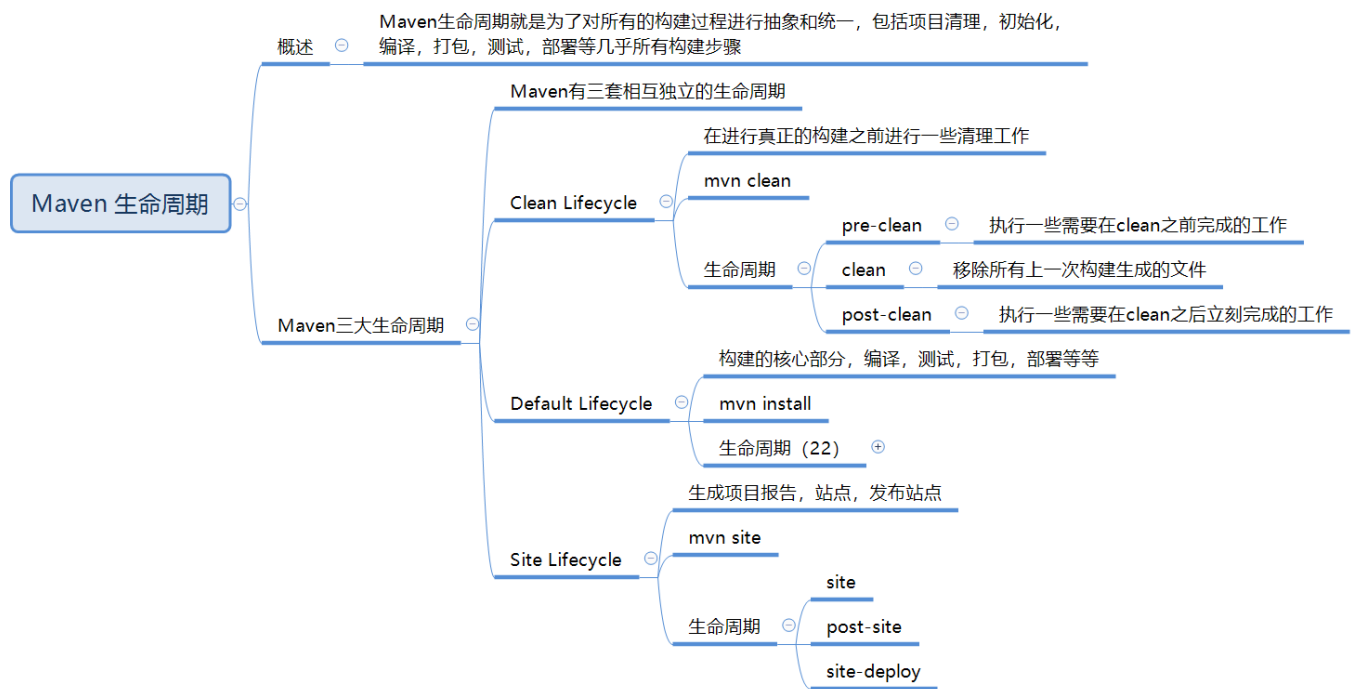
* 能够理解Maven相关的概念



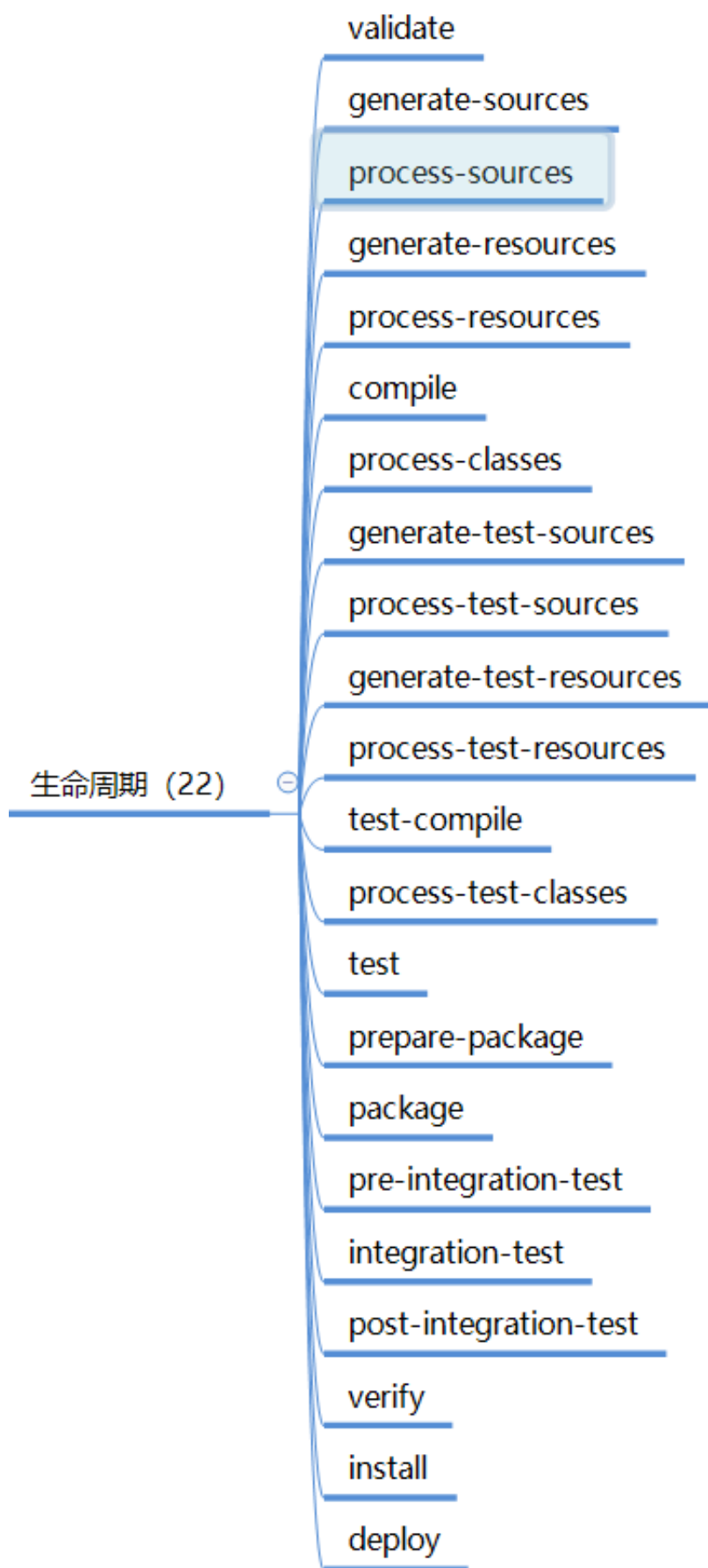
* Maven的仓库



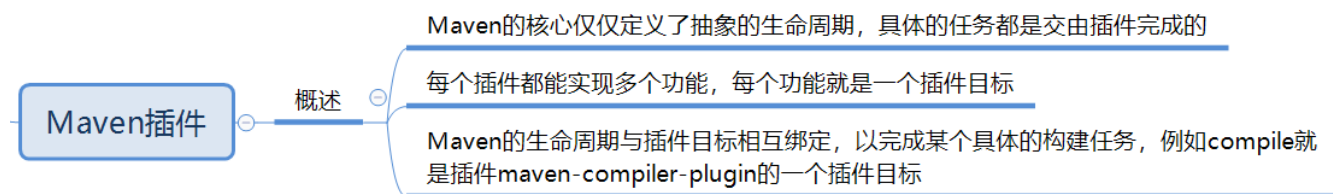
* Maven的生命周期



* Default Lifecycle



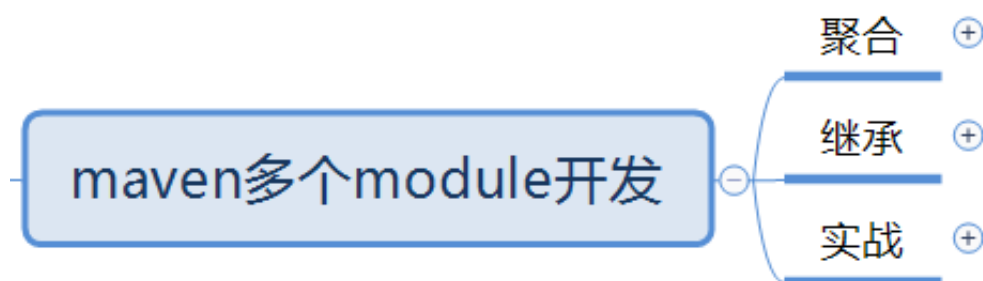
* Maven 插件



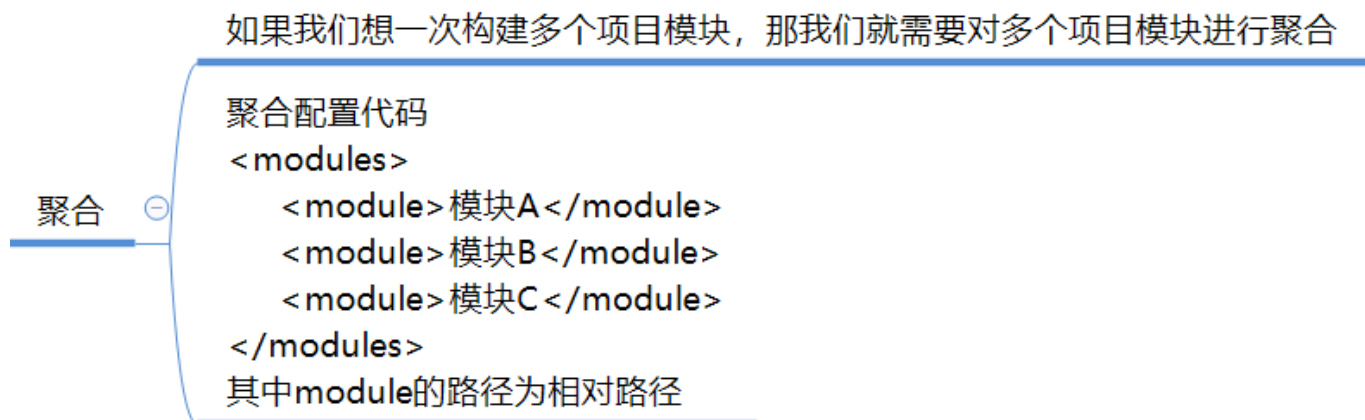
* 能够掌握Maven和idea结合开发方式

* 参考：[01-Maven和idea结合开发方式](#)

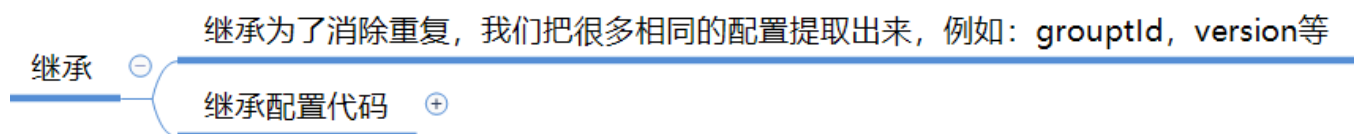
* 能够掌握Maven多模块的开发



* 聚合



* 继承



1 继承配置代码

```
<parent>
    <groupId>com.lg.maven</groupId>
    <artifactId>ParentProject</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <relativePath>../ParentProject/pom.xml</relativePath>
</parent>
```

2 继承代码中定义属性

```
<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <junit.version>4.9</junit.version>
    <maven.version>0.0.1-SNAPSHOT</maven.version>
</properties>
```

访问属性的方式为`${junit.version}`

```
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>${junit.version}</version>
    <scope>test</scope>
</dependency>
```

3 父模块用dependencyManagement进行管理

```
<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>${junit.version}</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>com.lg.maven</groupId>
            <artifactId>helloMaven</artifactId>
            <version>${maven.version}</version>
            <type>jar</type>
            <scope>compile</scope>
        </dependency>
    </dependencies>
</dependencyManagement>
```

这样的好处是子模块可以有选择的继承，而不需要全部继承

* 实战



* 按MVC模式和三层架构分Module

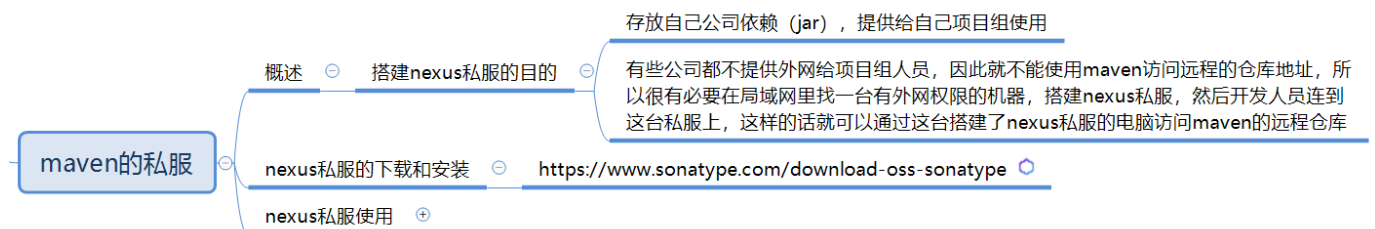
* 参考：[01-Maven结合MVC模式和三层架构分Module](#)

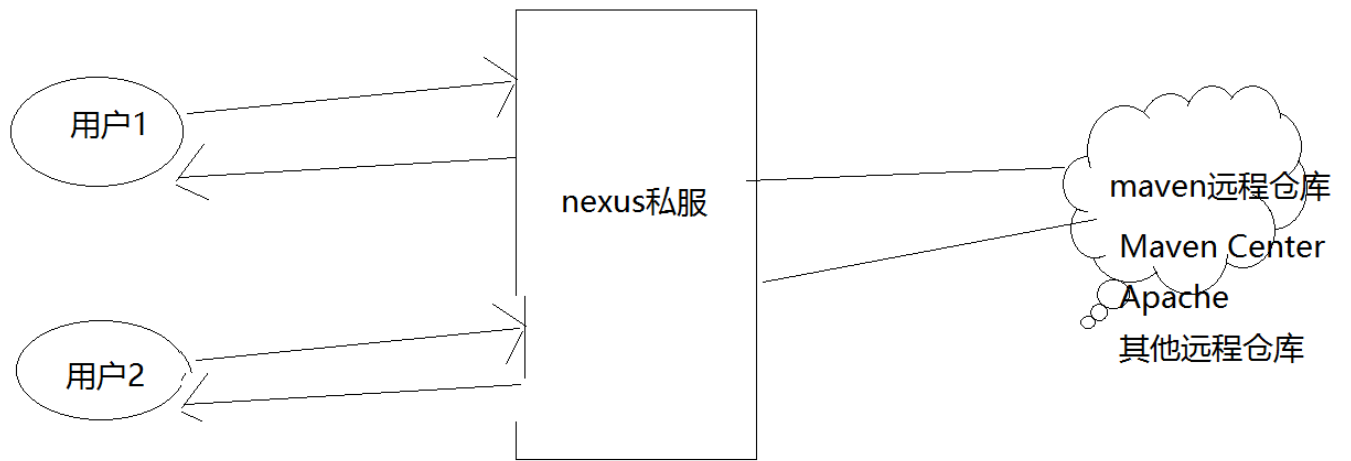
* 按功能模块来分Module

* 参考：[01-Maven结合功能模块来分Module](#)

* 能够掌握Maven私服的部署和使用

* Maven私服的概述





* Maven私服具体操作

* 参考：[01-Maven私服具体操作](#)