

# a01.模块化系统

讲师：李刚

## 本章要点

---

- 问题
- 目标
- 结构
- 语句

## Java 9之前的问题

---

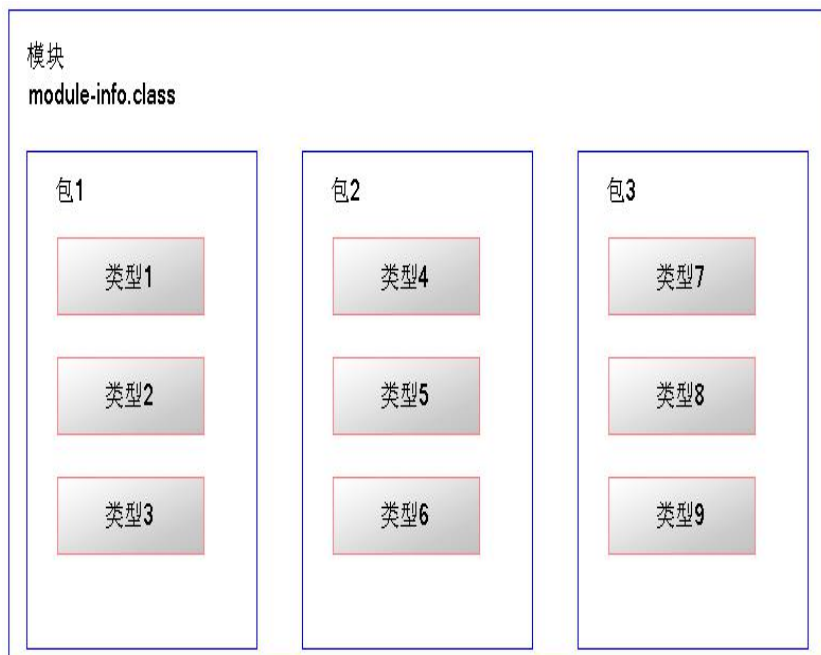
- 在Java 9之前，一个Java程序通常会以N个包的形式进行组织，每个包下可包含个Java类型（类、接口、枚举和注解），这种程序组织结构本身就存在以下问题。
  - 包只是充当命名空间的角色，包中的公共类型可以在所有其他包中访问；包并没有真正充当访问权限的界定边界。
  - Java程序运行时只能看到程序加载系列JAR包，无法真正确定不同JAR包中是否包含多个相同类型的不同副本，而Java程序运行默认加载类路径中遇到第一个JAR包所包含的Java类型。
  - Java程序运行时经常由于缺失某个JAR包所导致的ClassNotFoundException。有时候也会因为包含错误的JAR版本而导致运行时错误。

## 模块的目标

---

- 模块系统从两方面进行规范：
  - 模块化系统将整个JDK、JRE本身分解成多个相对独立的模块，这样应用程序可根据需要只加载必须的模块。
  - 应用程序、框架、库本身可以被分解成相对独立的模块，模块与模块相对独立，而且模块可作为访问权限的界定边界。

## 模块的结构



- 模块是一个比“包”更大的程序单元，一个模块可以包含多个包，而每个包下又可包含N个Java类型（类、接口、枚举和注解）。

## 模块结构

---

- 模块包括N个包，不同包的.java文件放在包对应的文件结构下。
- 模块的根目录下包含一个 **module-info.java** 文件，该文件是一个 **模块描述文件**（类似于XML配置文件，但比XML配置文件好）
  - [open] module <modulename>
  - {
  - <module-statement>;
  - <module-statement>;
  - ...
  - }



## 模块语句

---

- Java 9共支持以下5种模块语句。
- 导出语句 (exports statement) : 用于指定暴露该模块中哪些包。
- 开放语句 (opens statement) : 用于指定开发该模块中哪些包。
- 需要语句 (requires statement) : 用于声明该模块需要依赖的其他模块。
- 使用语句 (uses statement) : 用于声明该模块所使用服务的接口。
- 提供语句 (provides statement) : 用于声明为服务接口提供实现。

## 编译、运行模块

---

- javac增加了如下选项。
- `--module-source-path`: 指定一个或多个模块的源路径。
- `--module-version`: 指定模块的版本。
- java增加了如下用法（就像-jar用法）
  - `java [options] -m/module <模块>[/<主类>] [args...]`
- 还增加如下选项：
  - `--module-path`或`-p <模块路径>`: 用于指定模块的加载路径。
  - `--list-modules`: 列出模块。
  - `--d`或`--describe-module <模块名称>`: 用于描述指定模块。



## 使用jar命令打包模块JAR包

---

- Java 9同样增强了jar命令，开发者可通过jar命令打包模块。Java 9为jar命令增加了如下与模块相关的选项。
  - `--module-version=VERSION`: 设置模块的版本。
  - `-p`或`--module-path`: 设置模块的加载路径。

## 模块导出或开放

---

- 模块导出使用exports语句，exports语句的完整语法如下：
  - exports <package>;
  - exports <package> to <module1>, <module2>...;
- 模块还支持opens语句，opens语句的完整语法与exports相似，同样支持如下两种用法：
  - opens <package>;
  - opens <package> to <module1>, <module2>...;

## 模块依赖

- 模块依赖使用requires语句，requires语句的完整语法如下：
  - requires [transitive] [static] <module>;
- requires语句中的static修饰符表示该依赖模块在编译时是必需的，但在运行时则是可选的。比如在模块P中声明如下requires 语句：
  - requires static Q;
  - 它表明程序编译模块P时必须依赖模块Q；但模块P在运行时，模块Q是可选的。
- requires语句中的transitive修饰符表明该依赖具有传递性。假如现在有三个模块：P、Q和R，模块P依赖模块Q，而在模块Q中声明了如下语句：
  - requires transitive R;
  - 这意味着模块Q对模块R的依赖具有传递性，既然模块P依赖模块Q，那么模块P也依赖模块R。

## 声明的服务接口

---

- 使用uses语句来声明服务接口，**服务接口可供实现，可供调用**。
  - `uses org.crazyit.oa.UserService;`
- ServiceLoader的load传入服务UserService名，即可返回该服务接口的所有实现类。接下来程序既可根据名字获取指定的服务实现，也可获取第一个加载的服务实现。

## 提供服务实现

---

- 提供服务实现用provides语句。
  - provides UserService with UserServiceBest;
- 而ServiceLoader正是根据所有模块中module-info.java文件中的provides语句确定服务的实现类。