



Java 核心技术(高阶)

第九章 Java 模块化

第一节 Java 模块化概述

华东师范大学 陈良育



现状(1)

- JDK 8及以前开发模式

- Java源代码以Java文件（类/接口/注解等）的形式编写
- 每个Java文件被**明确地放入到一个包中**
- Java文件编译后的class文件，可以压缩为jar包，供别的程序调用
- 一个程序可以使用类库，类库通常以jar包呈现
- 将所有程序jar包，类库jar包，都放在classpath上，来运行程序
- **以Jar为中心**

现状(2)

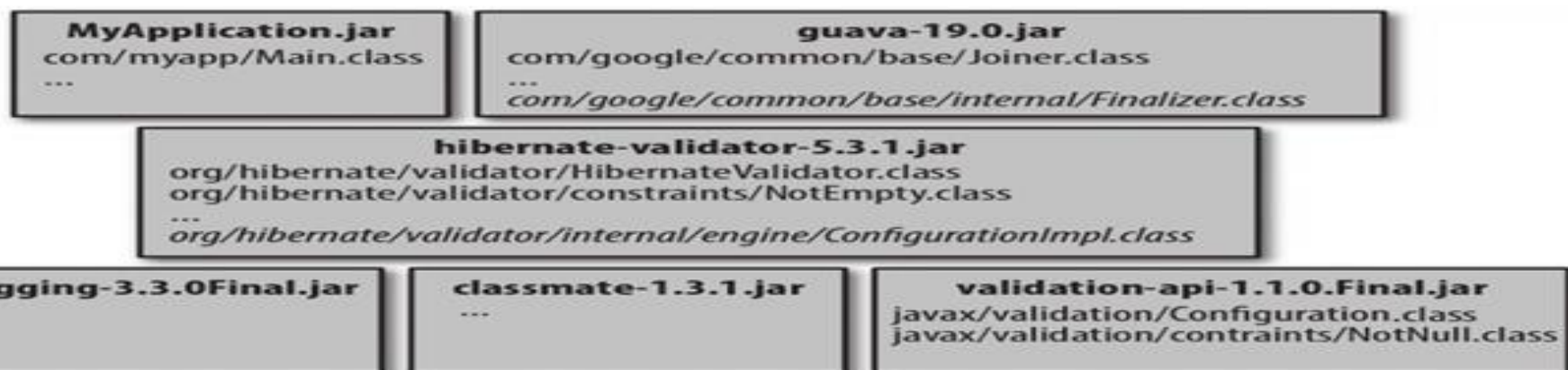


- Jar Hell
 - jar文件无法控制别人访问其内部的public的类
 - 无法控制不同jar包中，相同的类名(包名+类名)
 - Java运行时，无法判定classpath路径上的jar中有多少个不同版本的文件。Java加载第一个符合名字的类
 - Java运行时，无法预判classpath路径上是否缺失了一些关键类

现状(3)



应用程序



Java 运行时

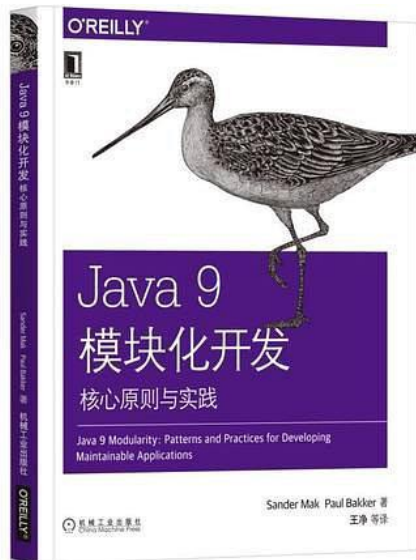


```
java -classpath lib/guava-19.0.jar:\
lib/hibernate-validator-5.3.1.jar:\
lib/jboss-logging-3.3.0Final.jar:\
lib/classmate-1.3.1.jar:\
lib/validation-api-1.1.0.Final.jar \
-jar MyApplication.jar
```


Java 模块化系统(1)



- 模块化必须遵循的三个原则
 - 强封装性：一个模块必须能够对其他模块隐藏其部分代码。
 - 定义良好的接口：模块必须向其他模块公开定义良好且稳定的接口。
 - 显式依赖：明确一个模块需要哪些模块的支持才能完成工作。





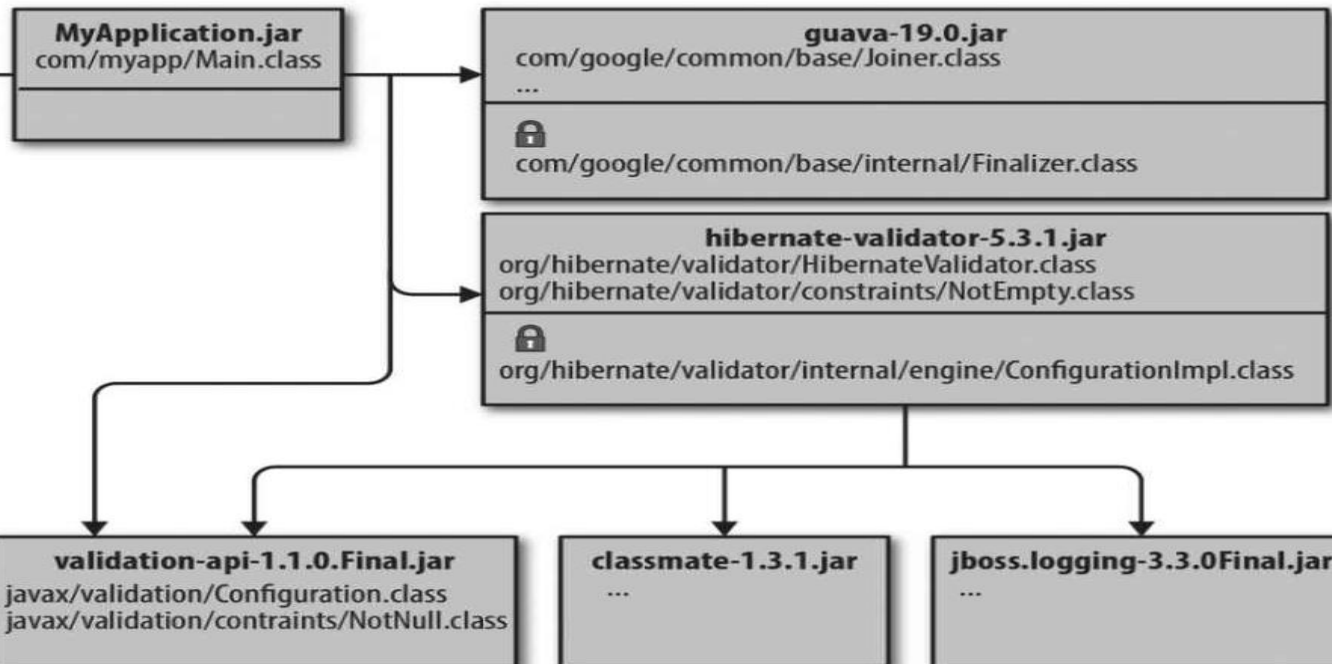
Java 模块化系统(2)

- Java 9开始引入新的模块化系统：Jigsaw 拼图
 - 以模块(module)为中心
 - 对JDK 本身进行模块化
 - 提供一个应用程序可以使用的模块系统
 - 优点
 - 可靠的配置
 - 强封装性
 - 可扩展开发
 - 安全性
 - 性能优化

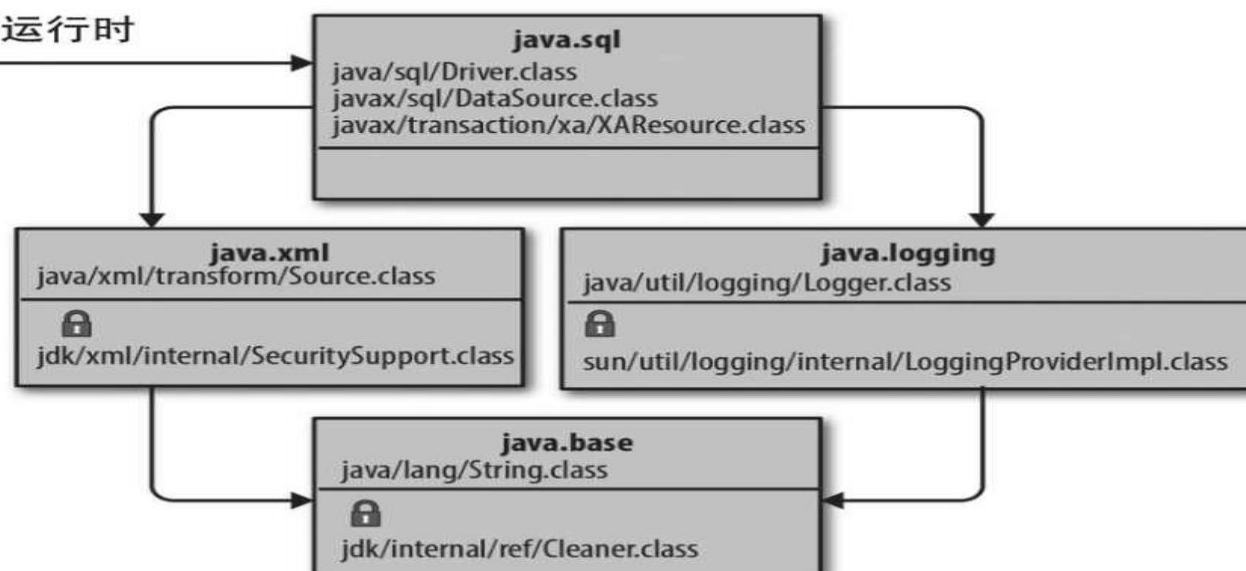
Java 模



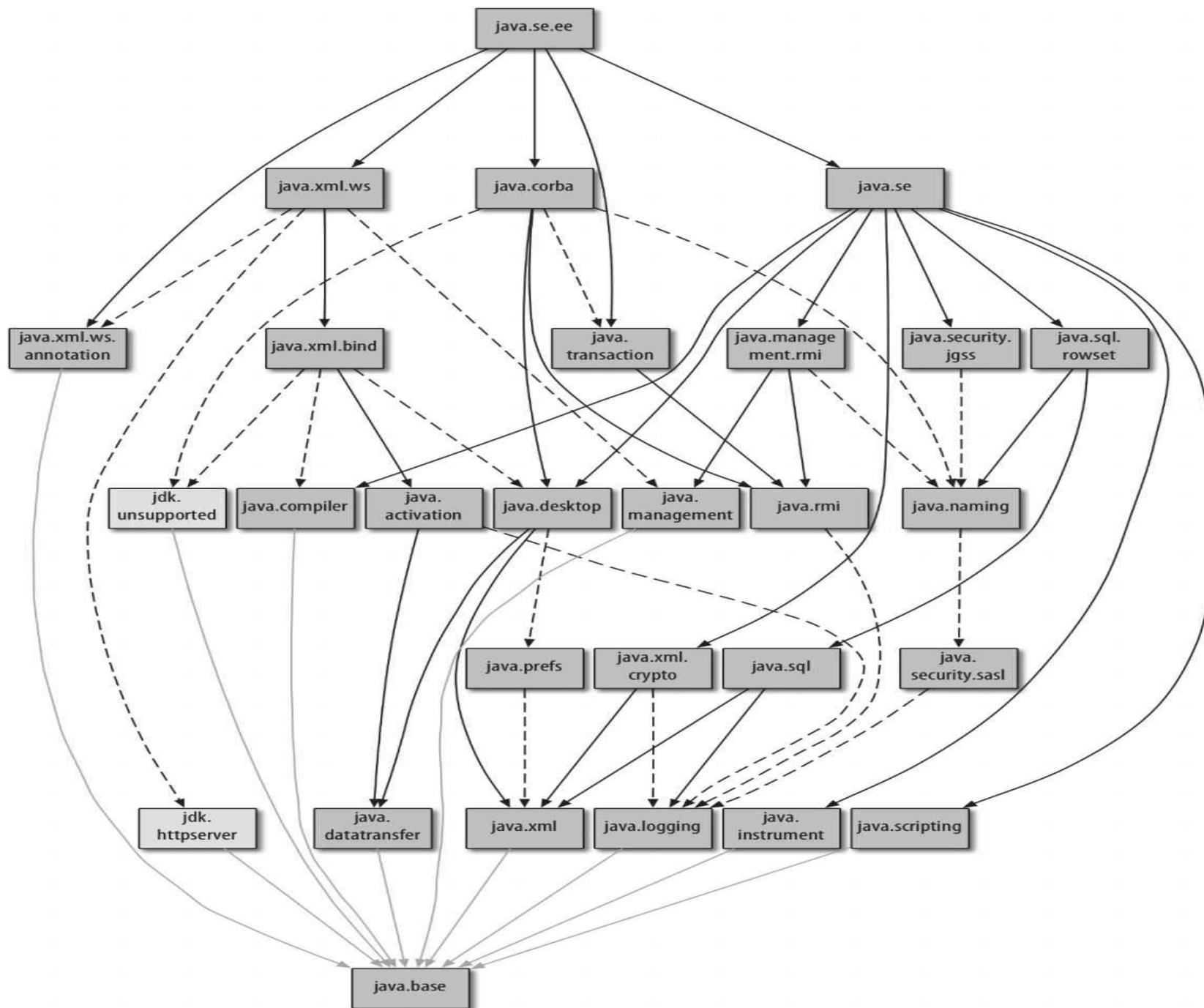
应用程序



Java 运行时



Ja





Java 模块化系统(5)

- 以Java 11.0.1为例
 - 一共有71个模块
 - 最底层的是java.base
 - 每个模块都有明确的依赖模块，不存在循环依赖
 - 使用java --list-modules 可以查看JDK的模块列表(需JDK9及以上)
 - 每个类都自动引用java.base模块
 - 使用java --describe-module 查看平台模块声明

总结



- 了解jar的优点和缺点
- 了解Java模块化的三个原则
- 了解Java新的模块系统



谢谢!