3、JVM的执行子系统

T A H N K Y O U F O R W A T C H I N G

主讲老师Mark：446106311

课程咨询安生老师：669100976

Class类文件结构

- Java跨平台的基础
- Class类的本质
  - ✓ Class文件是一组以8位字节为基础单位的二进制流
  - ✓ 类似于结构体的伪结构来存储数据
  - ✓ 只有两种数据类型：无符号数和表
  - ✓ 无符号数属于基本的数据类型，以u1、u2、u4、u8
  - ✓ 表是由多个无符号数或者其他表作为数据项构成的复合数据类型

Class文件格式

类　型	名　称	数　量
u4	magic	1
u2	minor_version	1
u2	major_version	1
u2	constant_pool_count	1
cp_info	constant_pool	constant_pool_count-1
u2	access_flags	1
u2	this_class	1
u2	super_class	1
u2	interfaces_count	1
u2	interfaces	interfaces_count
u2	fields_count	1
field_info	fields	fields_count
u2	methods_count	1
method_info	methods	methods_count
u2	attributes_count	1
attribute_info	attributes	attributes_count

Class文件格式详解

- 魔数与Class文件的版本
- 常量池
- 访问标志
- 类索引、父类索引与接口索引集合
- 字段表集合
- 方法表集合
- 属性表集合

字节码指令

- 简介和重要性
- 指令和数据类型
- 指令分类
  - ✓ 加载和存储指令
  - ✓ 运算指令
  - ✓ 类型转换指令
  - ✓ 对象创建与访问指令
  - ✓ 操作数栈管理指令
  - ✓ 控制转移指令

类加载机制

- 类加载过程详解

类加载器

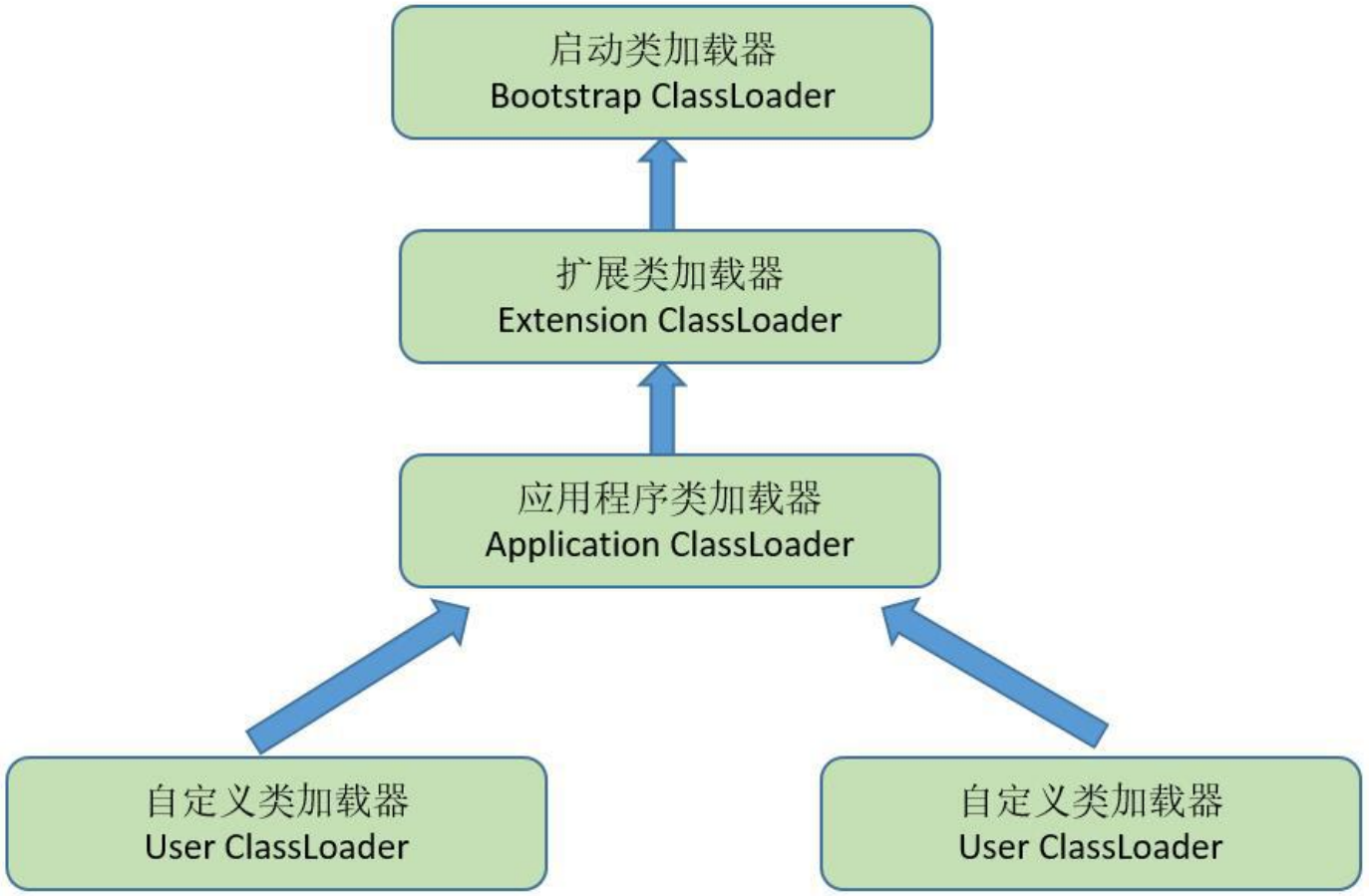
什么是类加载器？

- 用途：热加载、代码保护和加解密、类层次划分、OSGi等
- 自定义类加载对类进行加密和解密

系统的类加载器

类加载器名称	加载的范围
启动类加载器 Bootstrap ClassLoader	存放在<JAVA_HOME>\lib目录中的，并且是虚拟机识别的类库加载到虚拟机内存中
扩展类加载器 Extension ClassLoader	存放在<JAVA_HOME>\lib\ext目录中的所有类库，开发者可以直接使用；
应用程序加载器 Application ClassLoader	加载用户类路径上指定的类库，开发者可以直接使用，一般情况下这个就是程序中默认的类加载器；

双亲委派模型



- 双亲委派模型过程

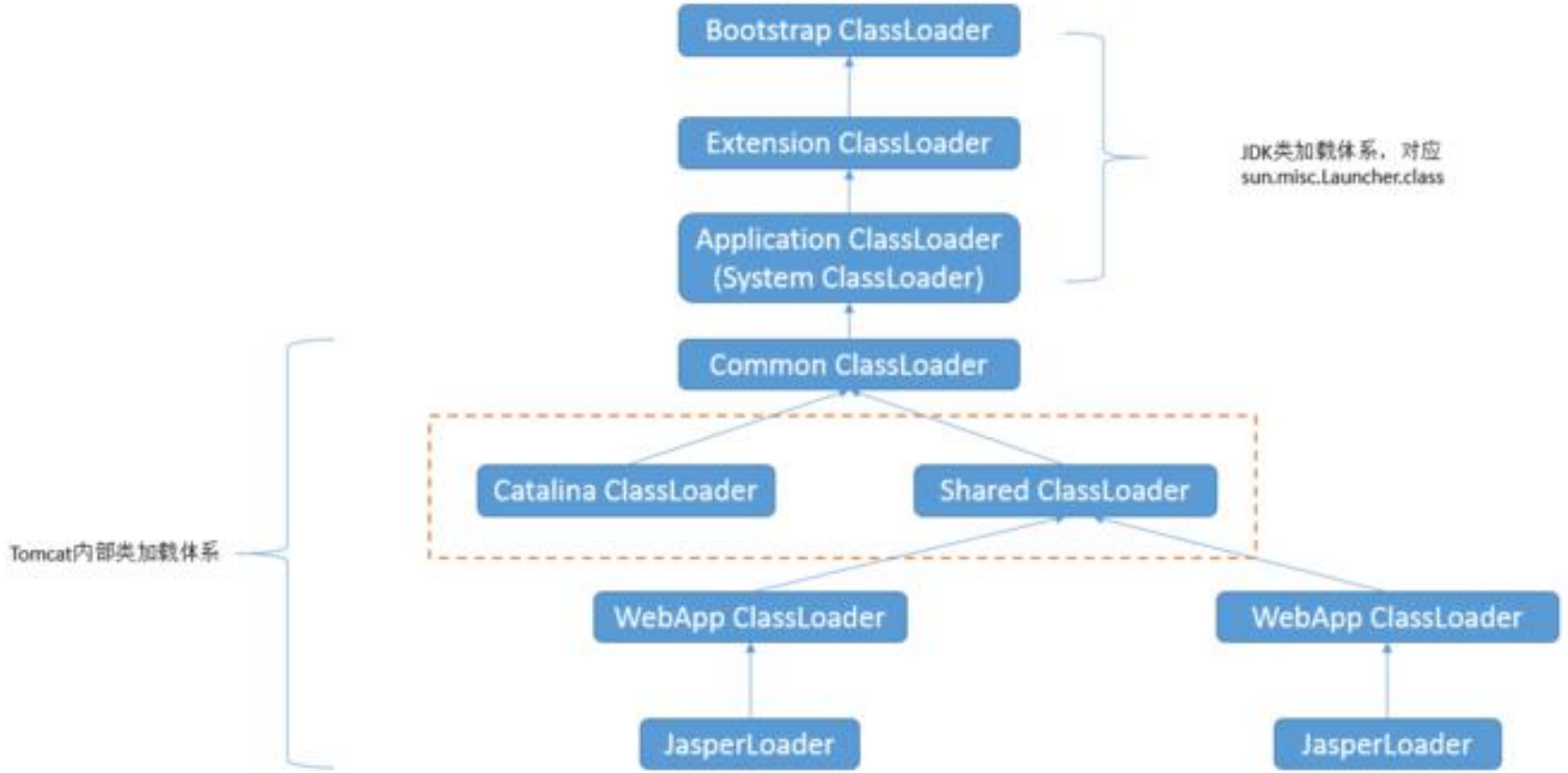
某个特定的类加载器在接到加载类的请求时，首先将加载任务委托给父类加载器，依次递归，如果父类加载器可以完成类加载任务，就成功返回；只有父类加载器无法完成此加载任务时，才自己去加载。

- 双亲委派模型好处

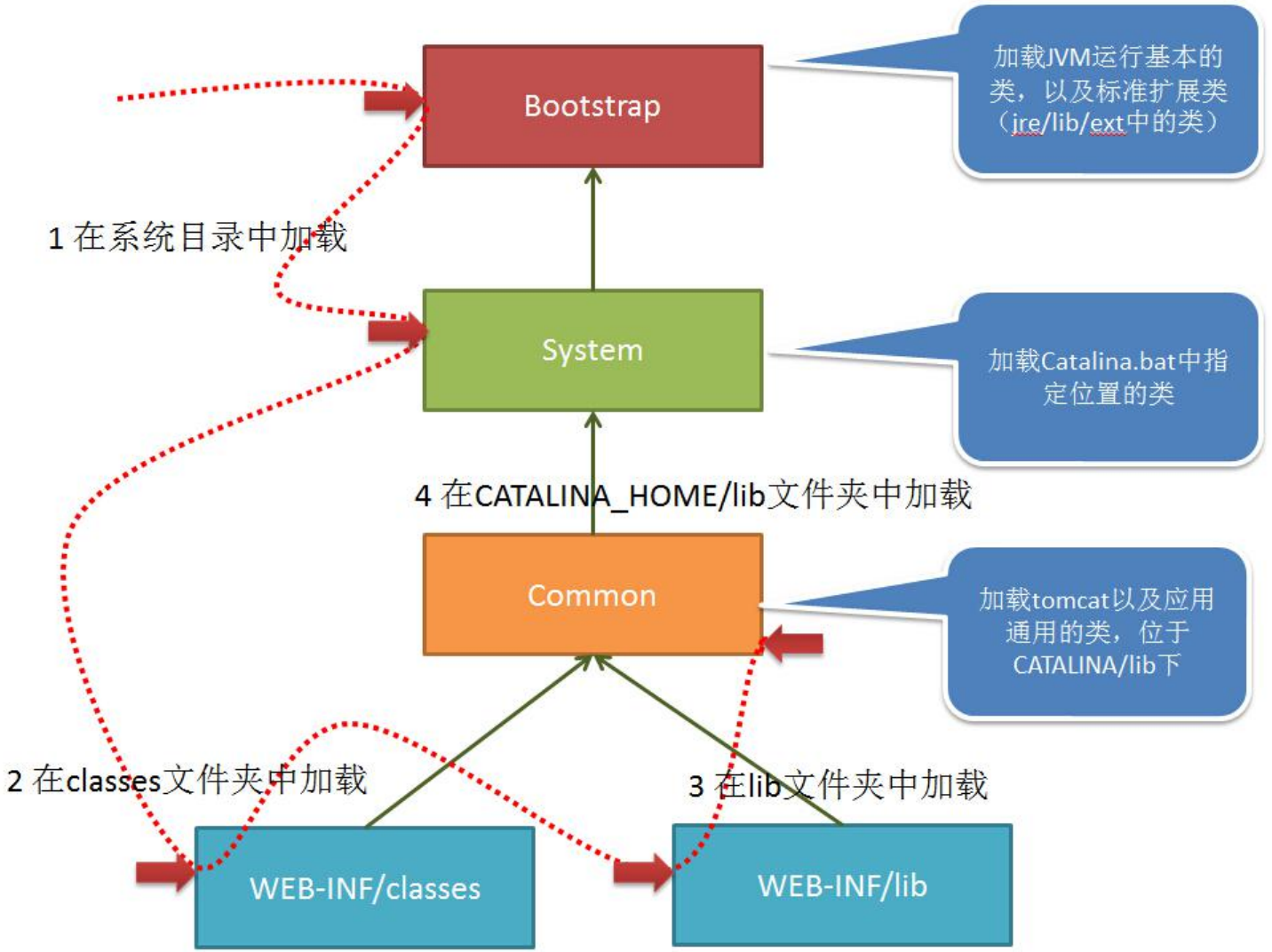
Java类随着它的类加载器一起具备了带有优先级的层次关系，保证java程序稳定运行

Tomcat类加载机制

- 同一个tomcat容器下的两个应用以及tomcat的lib目录中都有UserServiceImpl类，tomcat怎么样保证类的隔离性？



Tomcat类加载机制



栈桢

- 运行时的栈桢结构

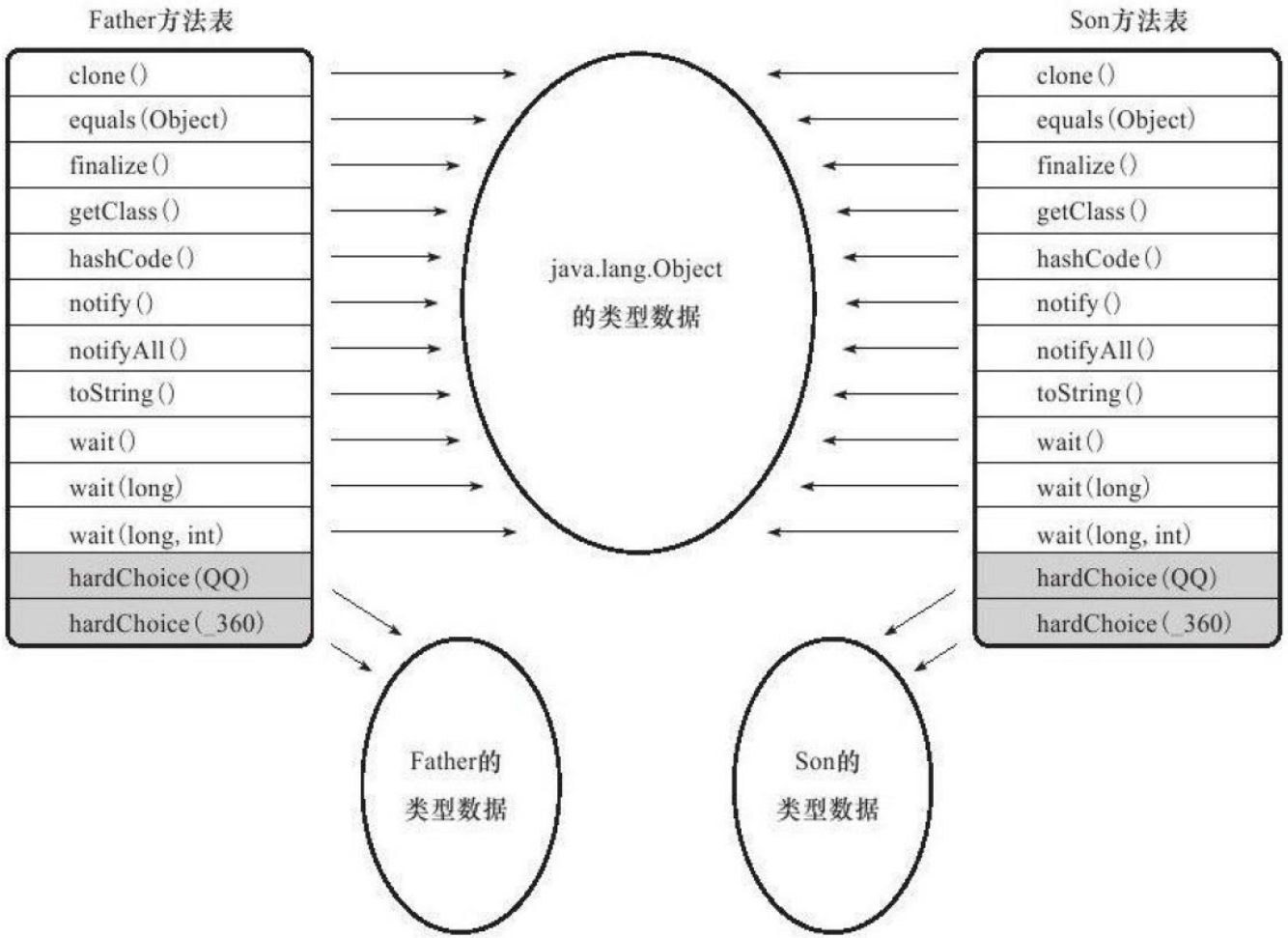

当前线程
线程2
线程n
当前栈帧
Current Stack Frame
局部变量表
Local Variable Table
操作栈
Operand Stack
动态连接
Dynamic Linking
返回地址
Return Address
…… ……
栈帧n
Stack Frame n
栈帧2
Stack Frame 2
栈帧1
Stack Frame 1
……

栈桢详解

- 局部变量表
- 操作数栈
- 动态连接
- 方法返回地址

方法调用详解

- 解析
- 分派
  - ✓ 静态分派
  - ✓ 动态分派
  - ✓ 动态分派的实现



基于栈的字节码解释执行引擎

- 基于栈的指令集与基于寄存器的指令集
- 基于栈的解释器执行过程，分析下面这段代码在虚拟机中的执行情况

```
public  int calc（）{
            int a=100；
            int b=200；
            int c=300；
            return（a+b）*c；
}
```