

*Write once, Run everywhere.*

# 初识Java开发

---

北京理工大学计算机学院  
金旭亮



# Java是什么?

Java是一个统称，它包括丰富的内容，并且其应用范围还在不断地拓展当中，以下是一个粗略的描述：



Java编程语言（及可运行于JVM之上的其他编程语言Kotlin, Scala）规范



Java虚拟机（JVM）规范及其各种实现



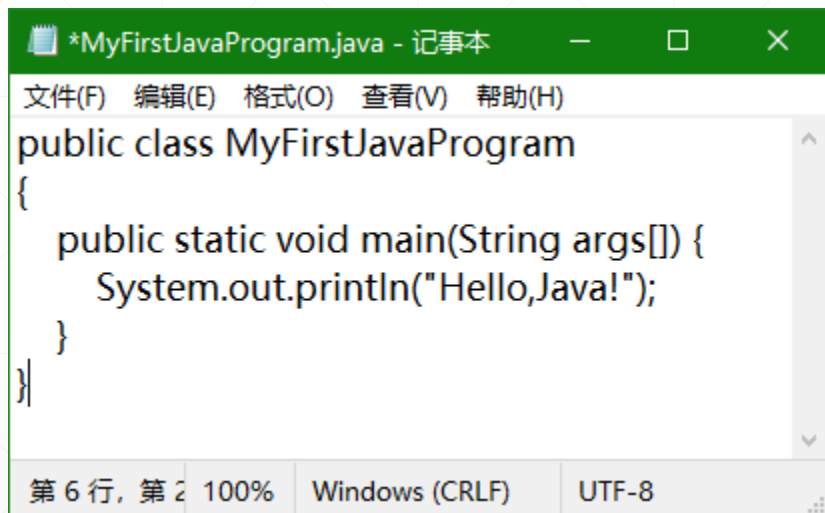
JDK，包容Java基础类库和各种开发工具



相关应用开发技术框架与各种开源项目：Java EE, Android, Spring, Netty等等

# 实例展示

在命令行下编译第一个Java程序：Hello,Java!

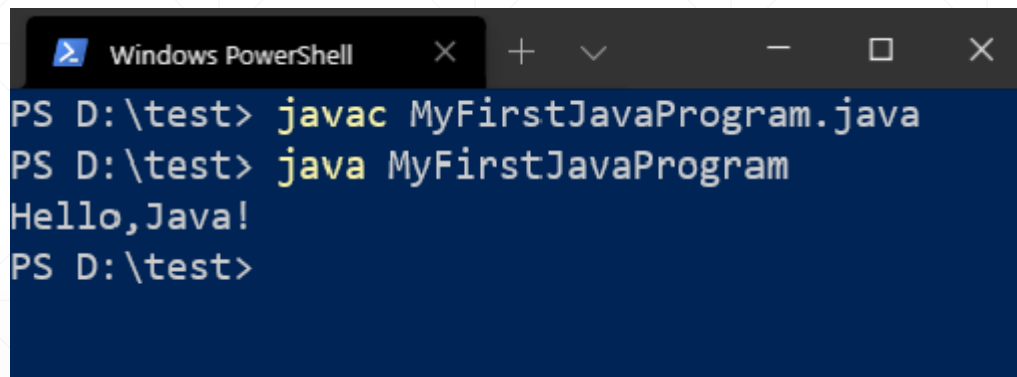


```
*MyFirstJavaProgram.java - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
public class MyFirstJavaProgram
{
    public static void main(String args[]) {
        System.out.println("Hello,Java!");
    }
}
```

第 6 行, 第 2 100% Windows (CRLF) UTF-8

1

编写Java源代码文件



```
Windows PowerShell
PS D:\test> javac MyFirstJavaProgram.java
PS D:\test> java MyFirstJavaProgram
Hello,Java!
PS D:\test>
```

2

用javac编译，生成字节码  
(.class) 文件

3

用java运行生成的字节码文件

# javac干了些什么事?

```
Windows PowerShell
PS D:\test> javac MyFirstJavaProgram.java -verbose
[语法分析开始时间 SimpleFileObject[D:\test\MyFirstJavaProgram.java]]
[语法分析已完成, 用时 22 毫秒]
[正在加载/modules/jdk.internal.vm.ci/module-info.class]
[正在加载/modules/jdk.naming.rmi/module-info.class]
[正在加载/modules/jdk.management.jfr/module-info.class]
[正在加载/modules/jdk.naming.dns/module-info.class]
[正在加载/modules/jdk.jdeps/module-info.class]
[正在加载/modules/java.desktop/module-info.class]
[正在加载/modules/jdk.incubator.jpackage/module-info.class]
[正在加载/modules/jdk.crypto.ec/module-info.class]
[正在加载/modules/java.security.sasl/module-info.class]
[正在加载/modules/jdk.editpad/module-info.class]
[正在加载/modules/jdk.httpserver/module-info.class]
[正在加载/modules/jdk.hotspot.agent/module-info.class]
```

`javac MyFirstJavaProgram.java -verbose`

1. 语法分析
2. 加载JDK中相应类型的字节码
3. 将编译结果写入.class文件。

# 注意事项



javac 编译，java 运行



类名和源文件名必须完全一致，标识符区分大小写



语句以分号结束（如果使用JShell，则分号可省）



一定要正确地设置环境变量，否则，javac 命令会找不到。

# 初步了解用Java编程的特点

- 使用Java编程，是一种面向对象的编程方式
- 与C的对比：不存在独立于类的函数，编程的最基本单元为类
- Class（类）
  - 包含完成某项任务的方法（methods）当任务完成时返回信息
  - 类是Java应用程序的基本构造块
- 类放在“包（package）”中，“包”对应于C++中的“命名空间（name space）”，主要用于解决同名问题。
- Java代码可以放到一个独立的压缩文件（.jar）中以方便部署。

# 跨平台与跨语言

一次编译处处运行 vs. 一次编码处处编译

Java

C/C++

一次编写处处重用 vs. 一次编写处处运行

.NET下的开发语言族

JavaScript, Python等

# Java语言与运行环境

- 编译型和解释型编程语言
- Java是一个“混合类型”的编程语言，跨平台，运行在JVM中，源代码是标准的文本文件，被编译为字节码，运行时再被转换为机器指令

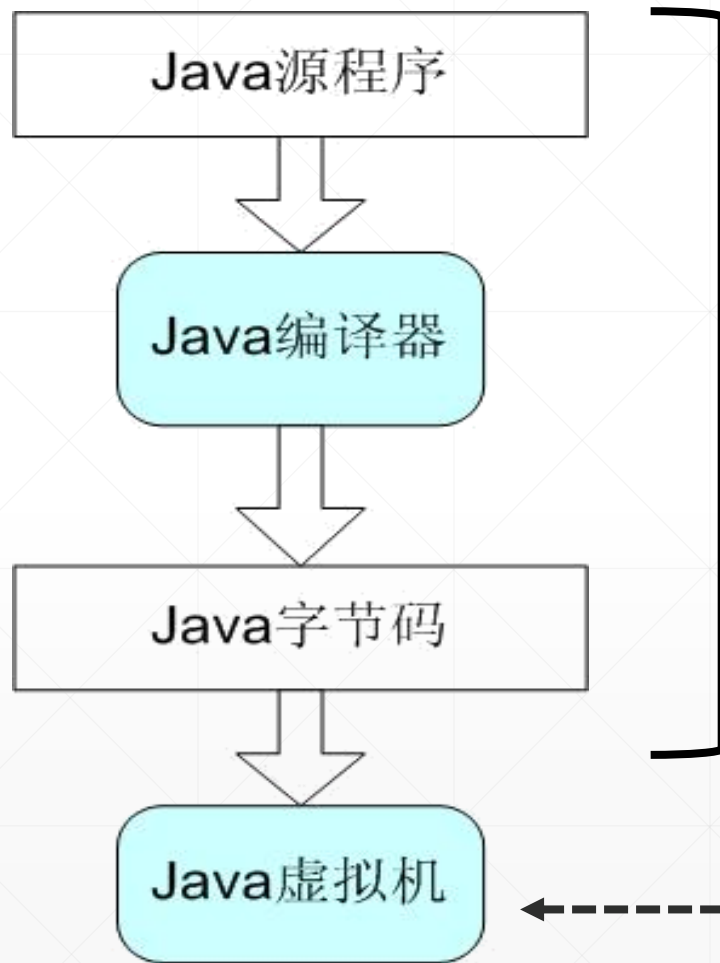
**Java应用程序**

**Java运行环境(JRE)**

**计算机操作系统 (windows,Mac,Linux,...)**



# Java虚拟机与Java应用程序



Java源代码被Java编译器转换为字节码写入.class文件。

Java虚拟机从.class文件中读入字节码指令并执行它们。

# Java运行时环境—JRE

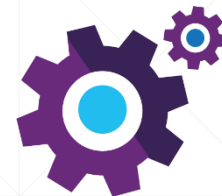


## JRE(Java Runtime Environment):

包容JVM以及运行Java程序的其他基础设施（比如基类库），没有它就不可能在Windows等操作系统上运行Java程序。

早期的Java程序(称为Java Applet), 可以在浏览器中直接运行, 直到现在, Windows的“控制面板”中仍然有一个相应的“Java控制面板”程序, 但现在已经不再使用这个技术特性了。

# 区分JDK与JRE



JDK的全称是：Java SE Development Kit。

**JDK = JRE + 相关的开发工具（如javac）**



如需要开发Java程序，必须安装JDK



只需要运行编译好的Java程序，安装JRE即可。

# 小结



## Java开发工具箱（JDK）

### Java运行时环境（JRE）

#### Java虚拟机（JVM）

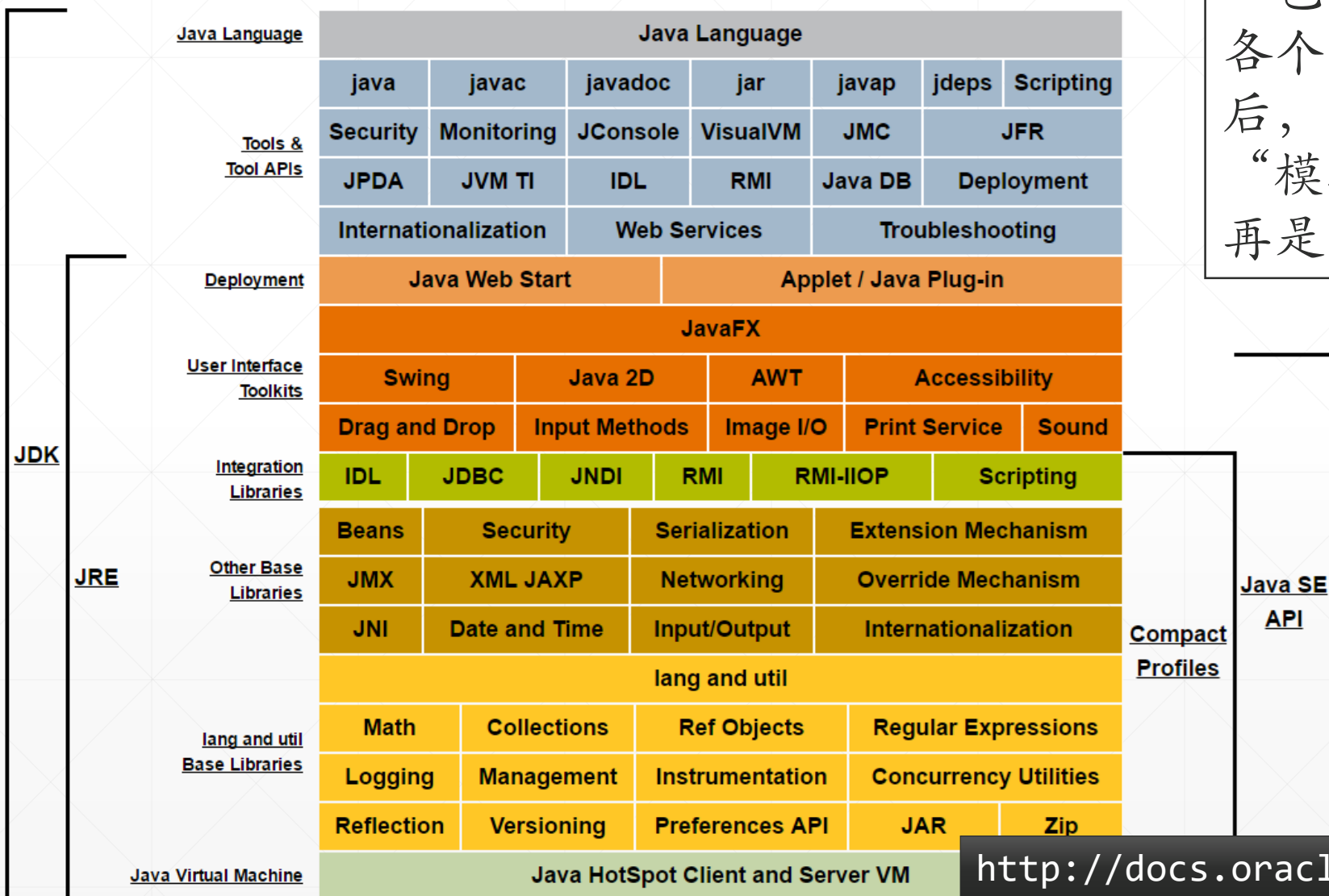
用于执行Java字节码  
的虚拟计算机

JRE可以运行Java应用程序，  
它包容着Java基础类库及  
Java虚拟机

JDK包容有用于开发Java应用的所有工具，比如Java编译器，JShell，JConsole等等，特别地，它包容JRE。

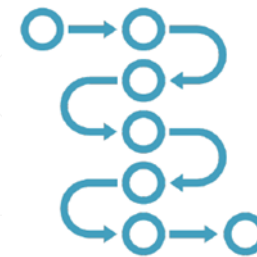
# Java SE 8 全家福

在JDK9之前，JDK是按照“包（package）”来管理各个组件的，到了JDK9之后，整个JDK被打散为多个“模块（module）”，不再是一个整体。



<http://docs.oracle.com/javase/8/docs/>

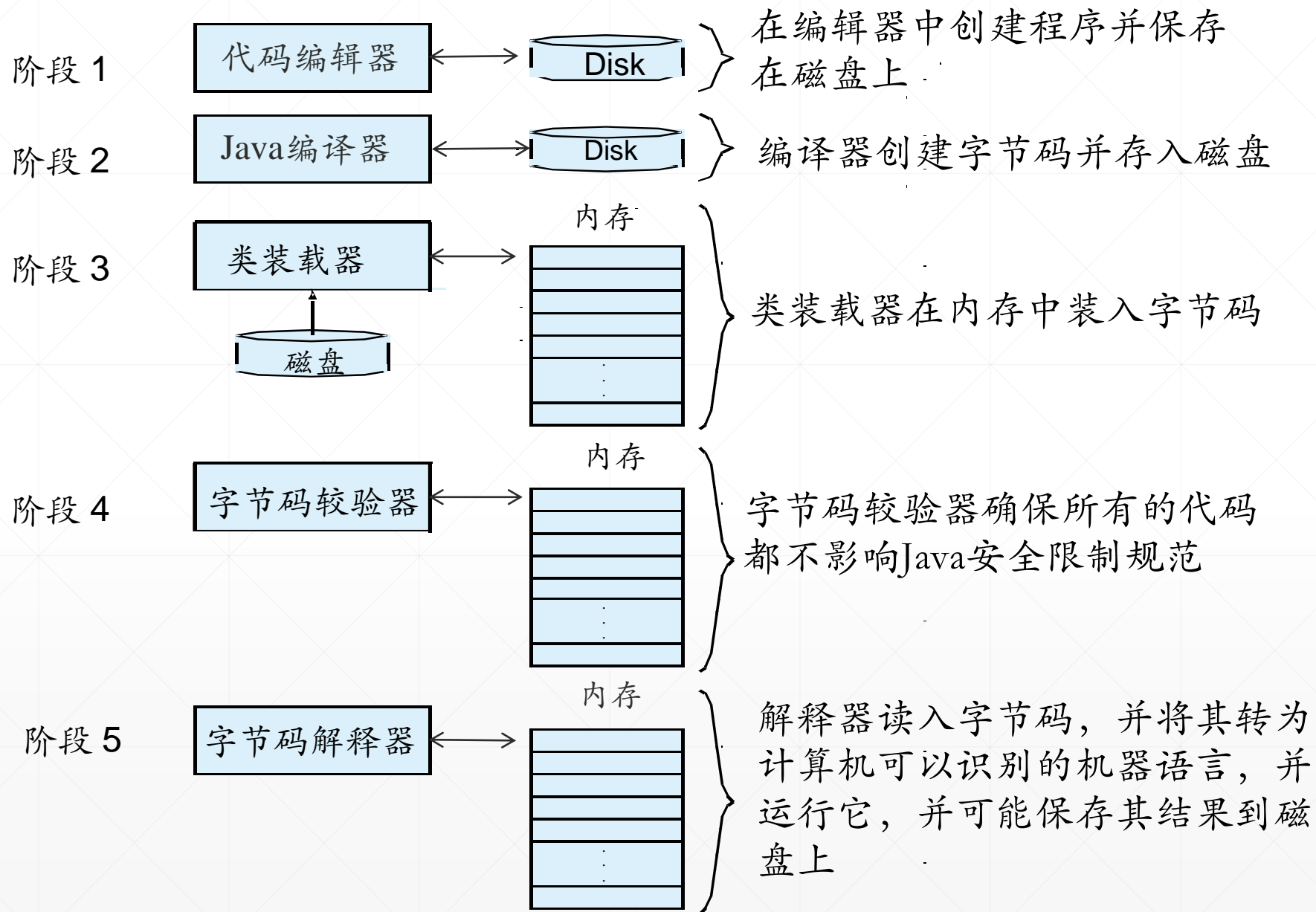
# 一个典型的Java程序开发过程



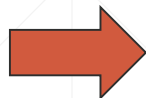
编写Java程序通常要经历五个阶段：

1. **Edit（编辑）**：程序员书写代码并保存到磁盘上
2. **Compile（编译）**：编译器生成字节码（byte code）
3. **Load（装入）**：类装载器（Class loader）在内存中保存字节码
4. **Verify（校验）**：校验器Verifier保证字节码不违反安全规则
5. **Execute（执行）**：解释器将字节码翻译成机器码

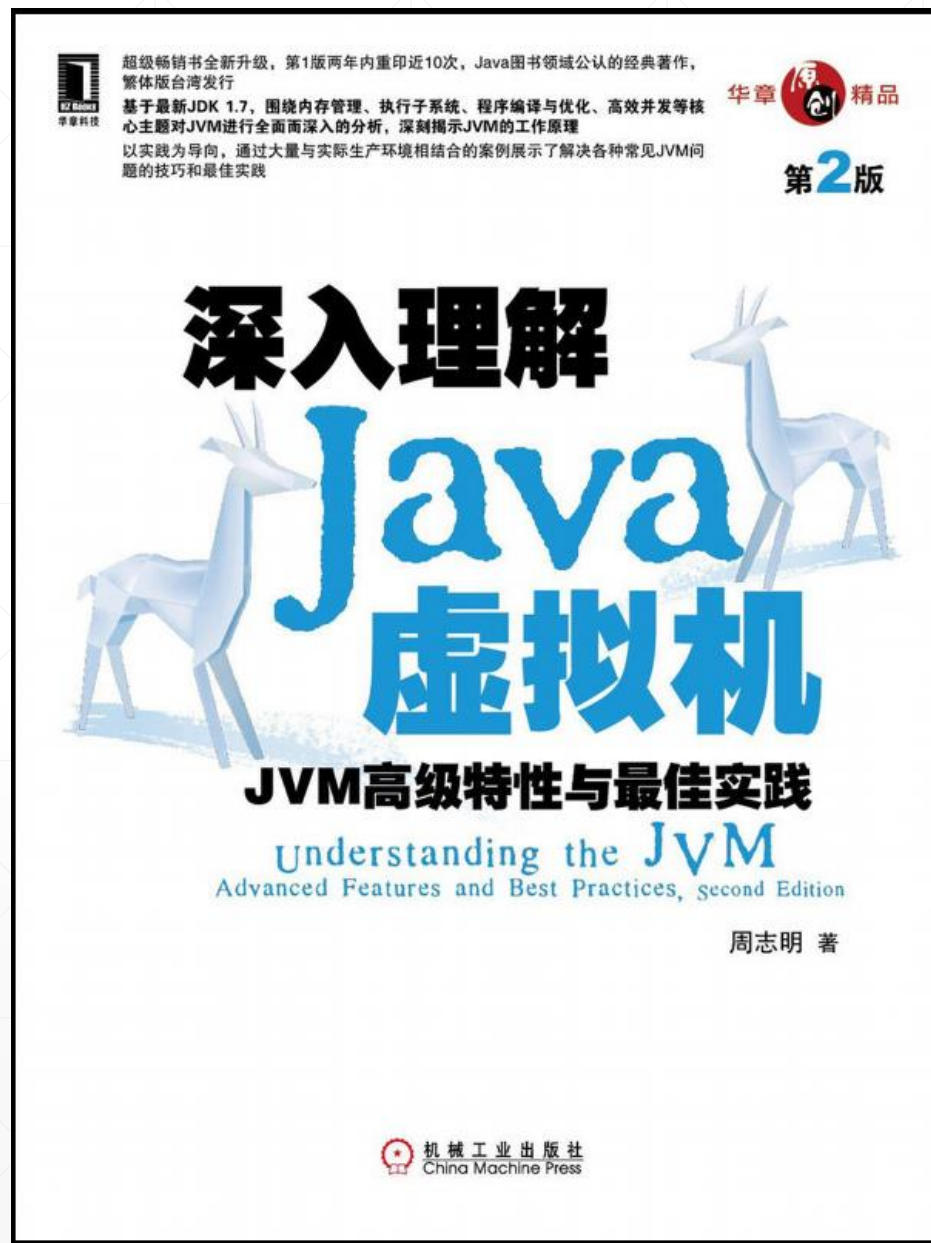
# 图解



如果想深入地了解Java虚拟机的内部运行机理，推荐看周志明先生写的这本书。



Java虚拟机调优是一门学问，对于开发高并发高性能的Java应用有很重要的意义和作用。





# 课后作业



1

手工用javac编译HelloWorld.java成功

2

在互联网上通过搜索引擎寻找java相关资源，为学习与开发实践积累资源。

3

着手开始建立起自己的编程资料库，平时应该将一些可以重用的代码整理好，放入自己的代码仓库以便随时取用。  
在学习过程中可以将所思所得随时以博客的方式记录下来。  
推荐在GitHub上申请帐号，从现在开始创建个人的开源项目。