JAVA优点：

“一次编写，到处运行”；

提供了相对安全的内存管理和访问机制，避免了绝大部分的内存泄露和指针越界问题；

实现了热点代码检测和运行时编译及优化，使得Java应用能随着运行时间的增加而获得更高的性能；

有一套完善的应用程序接口，还有无数来自商业机构和开源社区的第三方类库

**1.2　Java技术体系**

1. Java技术体系：

Java程序设计语言

Java虚拟机

Class文件格式

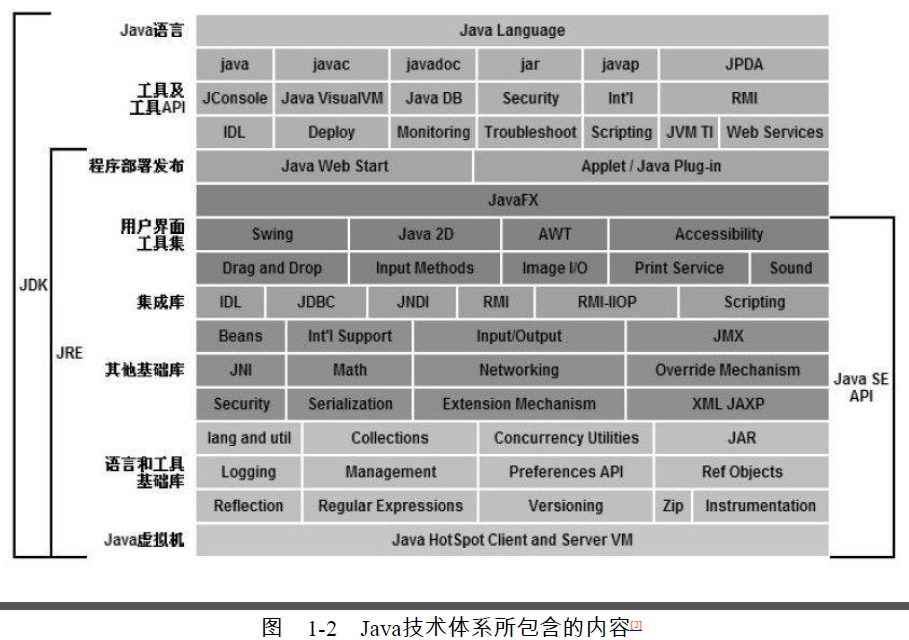
Java API类库

来自商业机构和开源社区的第三方Java类库

2. JDK（Java Development Kit）:Java程序设计语言、Java虚拟机、Java API类库.是用于支持Java程序开发的最小环境

3. 可把Java虚拟机和Java API类库中的JavaSE API子集[1]统称为JRE（Java Runtime Environment），支持Java程序运行的标准环境。

4. 根据组成部分的功能划分:



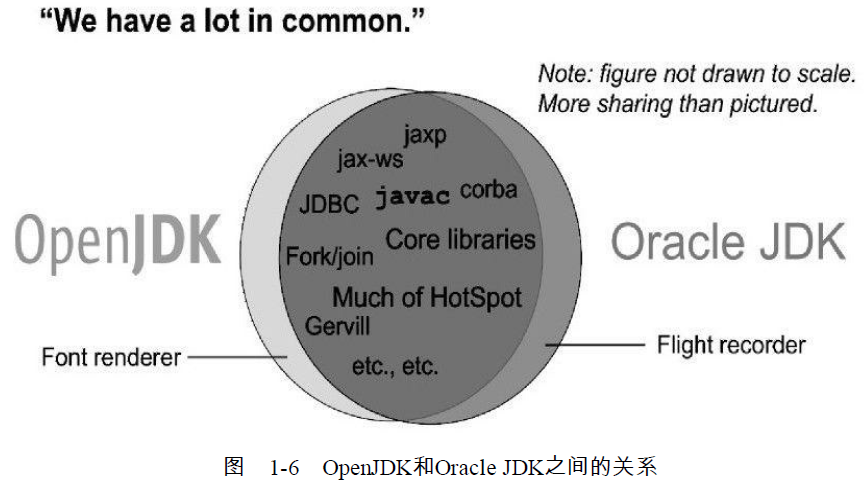
5.按技术所服务的领域划分，或者说按照Java技术关注的重点业务领域来划分:

* Java Card：支持一些Java小程序（Applets）运行在小内存设备（如智能卡）上的平台。
* Java ME（Micro Edition）：支持Java程序运行在移动终端（手机、PDA）上的平台，对Java API有所精简，并加入了针对移动终端的支持，这个版本以前称为J2ME。
* Java SE（Standard Edition）：支持面向桌面级应用（如Windows下的应用程序）的Java平台，提供了完整的Java核心API，这个版本以前称为J2SE。
* Java EE（Enterprise Edition）：支持使用多层架构的企业应用（如ERP、CRM应用）的Java平台，除了提供Java SE API外，还对其做了大量的扩充[3]并提供了相关的部署支持，以前称为J2EE。

**1.3　Java发展史**

自动装箱、泛型、动态注解、枚举、可变长参数、遍历循环（foreach循环）等语法特性都是在JDK1.5中加入的。在虚拟机和API层面上，这个版本改进了Java的内存模型（Java MemoryModel,JMM）、提供了java.util.concurrent并发包等。

**1.4　Java虚拟机发展史**



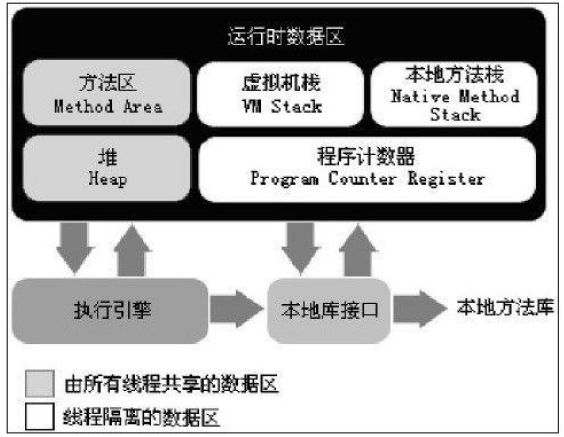
获取OpenJDK源码: 从Source Bundle Releases页面（地址：http://jdk7.java.net/source.html）取得打包好的源码

**第二部分　自动内存管理机制**

**第2章　Java内存区域与内存溢出异常**

1. 对Java程序员来说，在虚拟机自动内存管理机制帮助下，不再需要为每一new操作写配对的delete/free代码，不易出现内存泄漏和内存溢出问题.不过，也正是因此，一旦出现内存泄漏和溢出方面的问题，排查错误艰难。

**2.2　运行时数据区域**



**2.2.1　程序计数器**

是一块较小的内存空间，可看作是当前线程所执行的字节码的行号指示器。在虚拟机的概念模型里，字节码解释器工作时就是通过改变这个计数器的值来选取下一条需要执行的字节码指令，分支、循环、跳转、异常处理、线程恢复等基础功能都需要依赖这个计数器来完成。

Java虚拟机的多线程是通过线程轮流切换并分配处理器执行时间的方式来实现，在任何一个确定的时刻，一个处理器（对于多核处理器来说是一个内核）都只会执行一条线程中的指令。因此，为了线程切换后能恢复到正确的执行位置，每条线程都需要有一个独立的程序计数器，各条线程之间计数器互不影响，独立存储，称这类内存区域为“线程私有”的内存。

如果线程正在执行的是一个Java方法，这个计数器记录的是正在执行的虚拟机字节码指令的地址；如果正在执行的是Native方法，这个计数器值则为空（Undefined）。此内存区域是唯一一个在Java虚拟机规范中没有规定任何OutOfMemoryError情况的区域。

**第3章　垃圾收集器与内存分配策略**

**第4章　虚拟机性能监控与故障处理工具**

**第5章　调优案例分析与实战**