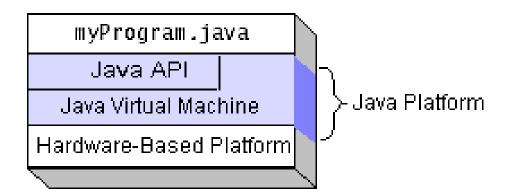
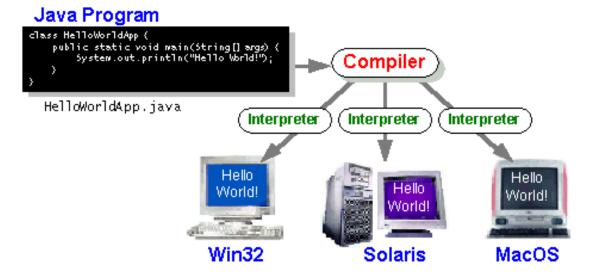
## Java程序设计

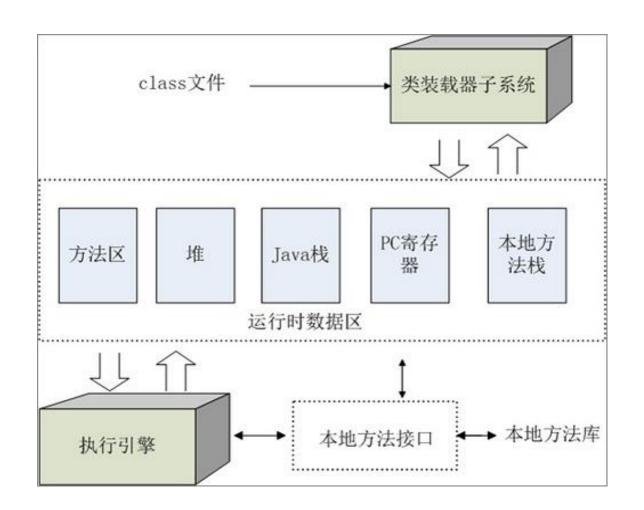
2019春季 彭启民 pengqm@aliyun.com

# Write once, run anywhere.

# Java Virtual Machine (JVM): "write once, run anywhere"







- .
- (1)简单的语言。
- (2)面向对象的语言
- (3)平台无关性 (虚拟机 结构中立)
- (4)多线程
- (5)健壮性
- (6) 分布性
- (7) 安全性
- (8)高性能
- (9)动态性
- (10)解释
- (11)可移植

### ■ NASA的科学家们通过Java程序Maestro来 控制勇气探索火星,像玩游戏一样简单。

### New Mars Data for Maestro: Opportunity #1

Press Release Source: Jet Propulsion Laboratory Posted Friday, February 13, 2004

The first package of Mars data from the rover Opportunity is now available for download. This package includes data acquired between Jan. 24 and Jan. 29, 2004, including the first images from Meridiani Planum.

To view the data, the Maestro application must first be downloaded and installed. The Maestro application allows the user to view Mars data in a variety of ways, including as a 3D terrain model. Plans for rover activities can also be created and simulated. This data package is the fourth of a series that will include data from both Spirit and Opportunity throughout the mission.

The Maestro application and both data packages are available through the Maestro Headquarters website at http://mars.telascience.org.

This email is addressed to people and organizations who were involved in some way with the first Maestro announcement. We appreciate your help in publicizing Maestro and the availability of data packages. If you do not want to receive further emails, please reply.

Please follow SpaceRef on Twitter and Like us on Facebook.

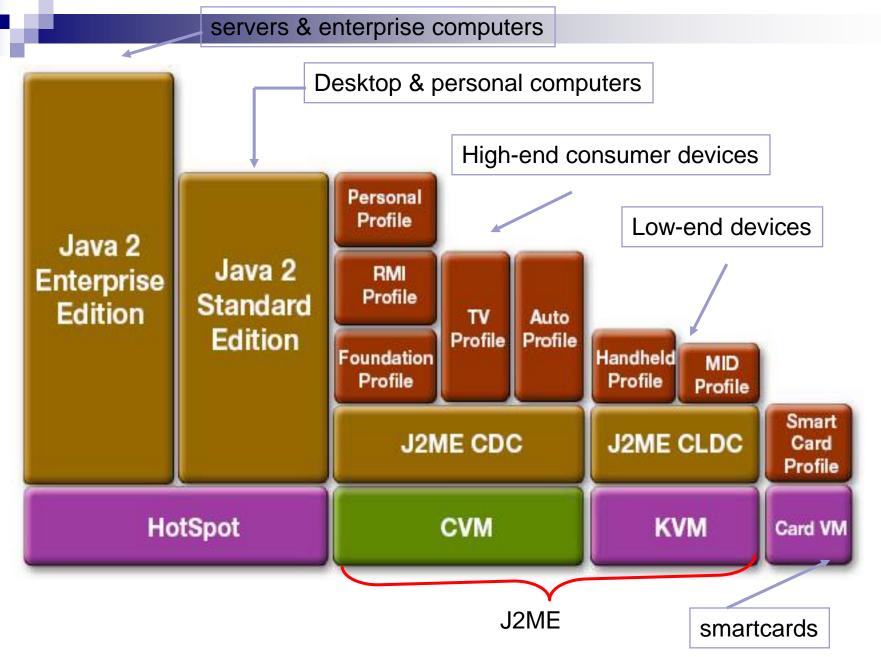
- JDK(Java Development Kit)
  - □开发工具,Javac及基础核心类
  - □运行环境, java及基础核心类
- Java平台有三种版本
  - □ JAVASE (Java Platform Stand Edition)
  - □ JAVAEE(Java Platform Enterprise Edition )
  - □ JAVAME(Java Platform Micro Edition )
    - Embedded

- Java平台有三种版本
  - □ JAVASE (Java Platform Stand Edition)
    - (J2SE)
    - ■基础包,允许开发和部署在桌面、服务器、嵌入式环境和实时环境中使用的 Java 应用程序。
    - 并且包含了支持 Java Web 服务开发的类,并为 Java Platform, Enterprise Edition (Java EE) 提供基础。

学习java应该从 JavaSE开始。

- Java平台有三种版本
  - □ JAVAEE(Java Platform Enterprise Edition )
    - (J2EE)
    - 在 Java SE 的基础上构建,提供 Web 服务、组件模型、管理和通信 API,用于实现企业级的面向服务体系结构(service-oriented architecture,SOA)和 Web 2.0 应用程序。
    - 开发和部署可移植、健壮、可伸缩且安全的服务器端 Java 应用程序。

- Java平台有三种版本
  - □ JAVAME(Java Platform Micro Edition )
    - (J2ME)
    - ■包括灵活的用户界面、健壮的安全模型、许多内置的网络协议以及对可以动态下载的连网和离线应用程序的丰富支持。
    - ■为在移动设备和嵌入式设备(比如手机、PDA、电视机 顶盒和打印机)上运行的应用程序提供一个健壮且灵活 的环境。
    - 基于 Java ME 规范的应用程序只需编写一次,就可以用于许多设备,而且可以利用每个设备的本机功能。



Java 2 Platform editions and their target markets

# Java应用的运行环境

■ Java的桌面应用 需要JRE的支持。

■ Java Web应用 至少分为三层:

Browser层:浏览器显示用户页面

Web层:运行Servlet/JSP

DB层: 后端数据库,向Java程序提供数据访问

服务

■ Java企业级应用 最简单分为4层:

Browser层或Client层

■ Web层:运行Servlet/JSP

EJB层:运行EJB,完成业务逻辑运算

DB层: 后端数据库,向Java程序提供数据访问服务

■ Java嵌入式应用 需要J2ME开发包,包含了嵌入式设备专用虚拟机KVM, 和普通的JDK中包含的JVM有所不同。另外还需要到特定 的嵌入式厂商那里下载模拟器。 ۲

- 1995.5.23 Sun发布了Java
- 1996年1月,第一个JDK-JDK1.0诞生(Java Development Kit JDK1.0)
  - □ Write Once, Run Anywhere.
- 1997年2月18日发布了JDK1.1,为JVM增加了JIT(即时编译)编译器(1997年4月2日,JavaOne会议召开,参与者逾万人,创当时全球同类会议纪录)
- 从1998.12开始,发布JDK1.2,并使用"Java2"
- 1999年6月分成了J2EE、J2SE和J2ME三大块

٧

- • 2000年5月8日, JDK1.3发布;
- • 2000年5月29日, JDK1.4发布;
- • 2001年6月5日, Nokia宣布到2003年将出售1亿部支持Java的手机;
- • 2001年9月24日, J2EE1.3发布;
- • 2002年2月26日,J2SE1.4发布,此后Java的计算能力有了大幅提升;
- • 2004年9月30日,J2SE1.5发布,成为Java语言发展 史上的又一里程碑。为了表示该版本的重要性,J2SE1.5 更名为Java SE 5.0;

# Java发展的历史

### **Platform Evolution**

### MSJVM

### Java 1.1

- Reflection
- Inner classes
- Basic Printing
- New event model
- RMI & Serialization
- AWT enhancements

Feb '97

### **J2SE 1.2**

- New Security Framework
- Swing
- Java 2D
- Accessibility
- Collections Framework
- Corba
- Input method framework
- Iava Sound
- Solaris native thread support
- Plug-in
- JPDA, JVMPI

Dec '98

### **J2SE 1.3**

- Performance
- Stability
- Java Hotspot VM
- Synchronized platform release trains
- JNDI
- Swing Enhancements

May '00

### **J2SE 1.4**

- New I/O
- 64 bit Solaris
- Security & Encryption
  - JSSE
  - JAAS
  - JCE
- WebStart
- XML
- Swing Improvements
- Performance
- Preferences
- Assertions
- Image I/O
- Printing

Feb '02

### **J2SE 5.0**

- Language enhancements
  - Generics
  - Metadata
- Enums
- Profiling
- JVM Monitoring & Management
- Improved Startup and Footprint
- Accessibility enhancements
- Scalability improvements
  - GC
  - Heap
- Improved L&F features
- Smart Tuning
- Swing improvements

Sept '04

М

- 2005年6月,JAVA10周年JavaOne大会召开,SUN公司公开Java SE 6。正式为J2SE、J2ME、J2EE重新命名:
  - □ J2SE—>JAVASE
  - □ J2ME—>JAVAME
  - □ J2EE—>JAVAEE
- 2006年12月,SUN公司发布JRE6.0;
- 2009年12月,SUN公司发布Java EE 6;
- 2011年7月28日,甲骨文发布Java SE 7;
- 2014年3月18日,甲骨文发布Java SE 8。
- 2017年9月21日,甲骨文发布Java SE 9。
- 2018 年 3 月 20 日, JDK 10
- 2018年9月25日,JAVA11
- 2019年?

# 1

### Java SE 8 (2014/3/27)

- Project Lambda (JSR 335)
- Nashorn JavaScript引擎
- 新的日期与时间API(JSR 310)
- 简洁的配置文件
- 从HotSpot Jave虚拟机(JVM)中去除了"永久代(permanent generation)"。
- JDK 8的性能提高了12%至14%
- 最新版JavaFX括: 一套特定嵌入图形、最新UI控制功能、Modena主题、使开发人员能够将Swing内容嵌入JavaFX应用的功能、新的3D图形功能以及对HTML 5的增强支持。

目前国内近 70% 的用户仍在使用 Java 8

### Oracle Java SE 8 发行版更新

Oracle Java SE 8 的公开更新仍面向单独的个人使用提供,至少持续至 2020 年底。

2019 年 1 月以后发布的 Oracle Java SE 8 公开更新将不向没有商用许可证的业务、商用或生产用途提供。

如果您是使用者,将 Java 用于单独的个人用途,则至少在 2020 年底之前,您对 Oracle Java SE 8 更新仍具有与现在相同的访问权限。在大多数情况下,您运行的基于 Java 的应用程序由 Oracle 之外的公司单独授予许可(例如,您在 PC 上玩的游戏很可能由游戏公司开发)。这些应用程序可以在 Java 平台上运行,并依赖于 Oracle Java SE 8 更新,直至 2020 年。因此,Oracle 建议您与应用程序开发商联系,以了解对方继续为您提供应用程序支持的计划的详细信息。

如果您是开发人员,Oracle 建议您查看 Java SE 8 及更高版本的路线图信息,并根据您开发的应用程序类型以及分发模式采取相应的操作。

如果您代表企业工作,Oracle 建议您查看 Java SE 8 及更高版本的路线图信息,并评估您的持续 Java 支持需求,以便根据实际情况,及时迁移到更高的版本或者获取商用许可证。对于在使用其他 Oracle 产品的过程中会用到 Java SE 的 Oracle 客户,可以继续享有这些 Oracle 产品的 Oracle Java SE 8 更新访问权限,直至 2019 年。有关详细信息,请参见此 My Oracle Support (MOS) 简讯。

https://java.com/zh CN/download/release notice.jsp

# м

### Java SE 9 (2017/9/23)

- Java 平台级模块系统:通过 "requires" 表示模块依赖
- Linking: 的jlink 工具可以创建针对应用程序进行优化的最小运行时映像而不需要使用完全加载 JDK 安装版本。
- Shell: 交互式 Java REPL
- 改进的 Javadoc: 支持在 API 文档中的进行搜索。输出符合兼容 HTML5 标准。每个 Javadoc 页面都包含有关 JDK 模块类或接口来源的信息。
- 集合工厂方法、改进的 Stream API、私有接口方法、HTTP/2、多版本兼容 JAR \_\_\_\_\_\_

```
▶ jshell
| Welcome to JShell -- Version 9-ea
| For an introduction type: /help intro

jshell> "abc".matches("a[bc]{2}")
$1 ==> true

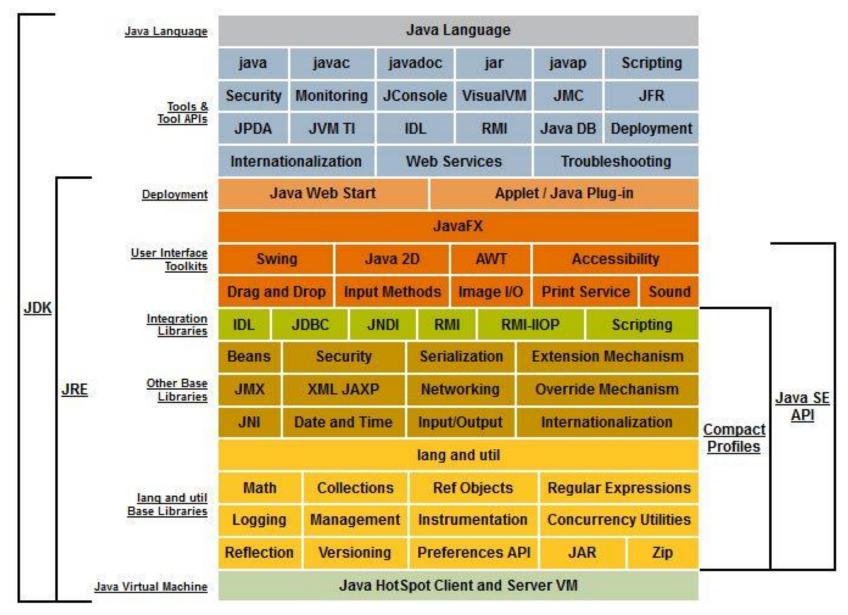
jshell> ■
```

### Java SE 10、11.....

- Jigsaw计划成为Java的一部分。
- .....

■Java已不再仅是语言+API, 而是成为一种标准与框架工 具,乃至为操作系统。

## Java Technology



- 方向
  - □ 想要做 application, 就可以专研 awt, swing 等等
  - □ 想要做 webapps, 就可以专研 jsp/servlet 等等
  - □ 想要做 pda/ phone, 就可以专研 midp, j2me 等等
- 应用服务器(App Server)
  - □ Tomcat
    - Tomcat严格意义上并不是一个真正的App Server,它只是一个可以 支持运行SerIvet/JSP的Web容器
  - □ BEA Weblogic
  - □ IBM Webshpere
  - □ Jboss 开源
  - □ 数据库系统
  - ... ...

- Java IDE
  - □ Eclipse(IBM)
  - Netbeans
  - □ Jbuilder(Borland)
  - □ Jcreator
  - ...
- Java web框架
  - □ Struts
  - □ Spring MVC
  - WebWork
  - □ Tapestry
  - □ JSF
  - □ ...

М

- JDK
- Java IDE
  - □ Eclipse(IBM) 已经成为开发java程序的首选IDE
  - □ Netbeans(SUN) 集成了web开发以及最近流行的ajax
    - ,而Eclipse要达到这点必须安装第三方插件
  - □ Jbuilder(Borland)
  - □ Jcreator 轻量级javalDE,功能非常单一

# Java SDK Installation Instructions (for Windows)

- 100
  - Download java SDK standard edition(SE)
     http://www.oracle.com/technetwork/java/index.html
  - 2. Run the Java SDK executable (\*.exe).
    - □ determine where to install java se. (ex: c:\java\jdk8)
    - □ > set JAVA\_HOME= c:\java\jdk8
  - 3. Update the PATH variable
    - so that you can type 'java' instead of 'c:\java\jdk8\bin\java' to invoke java tools.
    - > path=%JAVA\_HOME%\bin;%PATH%
  - 4. Check the CLASSPATH variable
    - Used by java tools to determine where to find your personal (nonsystem) java class files
    - Types of java byte codes (class files):
      - System: java tools know where to find them.
      - Extensions: put in %JAVA\_HOME%\jre\lib\ext
      - Personal: via CLASSPATH or –cp/–classpath options
  - 5. Start using the Java SDK tools!
    - □ java, javac, javadoc, jdb, javap,...

М

- 在资源页面点击下载Java 按钮 https://www.oracle.com/technetwork/java/javase/downloads/index.html
- 显示"文件下载"对话框,提示您运行或保存下载文件
- 单击运行或保存文件以便稍后安装,请单击保存。
- 选择文件夹位置,将文件保存到本地系统(将文件保存到 计算机上的某个已知位置,例如保存到桌面上)。
- 安装过程启动(安全机制可能会阻止,选"仍要运行"),接受 许可条款并继续安装。

(默认的防火墙设置可能会在某些条件下阻止 Java 的下载/ 安装操作。) М

■ 检测旧版本(8u20 和更高版本)。从 Java 8 Update 20 (8u20) 开始在 Windows 系统上,Java 卸载工具将集成在安装程序中,用于提供从系统中删除较早 Java 版本的选项。此更改适用于 32 位和 64 位 Windows 平台。

■ (注意: windows10不要使用edge浏览器)



适用于您的计算机的 Java 软件,即 Java 运行时环境,也称为 Java 运行 时、运行时环境、运行时、JRE、Java 虚拟机、虚拟机、Java VM、JVM 、VM、Java 插件、Java 附加程序或 Java 下载。

















#### 可用的操作系统

- Windows
- Mac
- Linux
- Solaris

#### 帮助资源

» Java 疑难解答

#### Java 7

» 从何外获取 Java 7?

#### JDK

» 正在寻找 JDK?

#### 适用于所有操作系统的 Java 下载

#### 推荐 Version 8 Update 161

发行日期: 2018年1月16日

根据您的操作系统,从下面的列表中选择文件以获取适用于您计算机的最新 Java 版本。

> <u>删除旧版本</u> > <u>什么是 Java?</u>

#### 下载 Java 即表明您确认阅读过并接受最终用户许可协议的条款

| <b>#</b> . | Windows | 我应该选择哪一项? | _ |
|------------|---------|-----------|---|
|            |         |           |   |

Windows 联机
 文件大小: 1.78 MB

在安装 Java 后,可能需要

 Windows 脱机

 文件大小: 61.35 MB

<u>说明</u>

说明

重新启动浏览器以在浏览

器中启用 Java。

○ Windows 脱机 (64 位) 文件大小: 68.02 MB

<u>说明</u>

如果您交替使用 32 位和 64 位浏览器,则需要同时安装 32 位和 64 位 Java,从而获得这两种浏览器的 Java 插件。 » 适用于 Windows 的 64 位 Java 的常见问题解答

#### Mac OS X Mac 常见问题解答

Mac OS X (版本 10.7.3 和更高版本) 文件大小: 74.17 MB

说明

在安装 Java 后,可能需要 重新启动浏览器以在浏览 器中启用 Java。

\* Oracle Java (版本 7 和更高版本) 需要基于 Intel 的 Mac, 该操作系统运行的是 Mac OS X 10.7.3 (Lion) 或更高版本,并且需要管理员权限才能安装。 » 详细信息

Linux

Linux RPM 文件大小: 63.4 MB
 Linux 文件大小: 79.29 MB

<u>说明</u>

说明 在安装 Java 后,需要在浏

O Linux x64 文件大小: 76.35 MB

<u>说明</u>

览器中启用 Java。

Linux x64 RPM 文件大小: 60.4 MB

说明

#### 帮助资源

- 安装 Java
- 删除旧版本
- 禁用 Java
- 使用 Java
- 一般问题
- Mobile Java
- 安全
- 支持选项

#### 如何开始使用 Java 开发工具包 (JDK) 开发 Java 程序?

编写 Java 小应用程序和应用程序需要类似于 JDK 的开发工具。JDK 包括 Java 运行时环境、Java 编译器和 Java API。无论是编程新手还是有经验的程序员,都很容易入门。



#### 从何处下载 JDK?

要下载最新版本的 Java 开发工具包 (JDK),请转到 <u>JDK 下载</u>。 开发人员还可以参阅<u>面向 Java 开发人员的 Oracle 技术网</u>以充分了解 Java 技术,其中包括<u>文档和培训。</u>

#### 如果我是 Java 新用户该如何做?

如果您是 Java 新用户并有兴趣着手开发 Java 程序,请参阅 new to Java 以找到适合初学者的有用信息。

#### 如何获得 Java 认证?

对于希望高薪聘请员工的雇主们来说,员工获得 Oracle Java 技术认证是技术技能、专业素养和工作进取的 有力体现。Oracle 推出的 Java 技术培训和认证计划倍受业界的赞誉,可以确保学员拥有必备的技能,轻松 应对 IT 公司的挑战。

» 了解 Java 认证详细信息

#### Java 开发人员大会

- JavaOne 是最重要的 Java 开发人员大会,与会者可以了解最新的 Java 技术,增进技术理解,并且可以直接向其他软件战略人员和开发人员提问。Oracle 每年都举办 JavaOne 大会,包括在旧金山举办的旗舰 JavaOne 大会和区域会议。有关即将到来的活动和地点的详细信息,请访问www.oracle.com/javaone。
- Oracle 技术网开发人员日是免费的 Java 开发人员实践研讨会,定期在全球召开。
- Oracle 还组织多项第三方 Java 技术会议和活动。要了解在您附近举行的即将到来的活动,请搜索 Oracle 活动目录。

#### Java 期刊

Java 期刊是公开发布的数字版双月刊,是关于 Java 技术、Java 编程语言和基于 Java 的应用程序的重要知识来源,面向在工作中使用 Java 或者希望使用 Java 的人员。其中提供了有关创新 Java 应用程序、Java 技术使用方法、Java 社区新闻以及有关新 Java 书籍、会议和活动的信息。

#### **Oracle Academy**

Oracle Academy 面向中小学、职业教育和高等院校提供了用于教学的全面的软件、课程、托管技术、员工培训、支持和认证资源组合。院校可以将这些资源灵活地融入计算机科学和商业课程内,确保学生在走上工作岗位之前熟练掌握行业相关的技能。Oracle Academy 在 95 个国家/地区有 1500 万名以上的学员。Oracle Academy 近期扩展了其课程以纳入 Java。如需了解详细信息,请访问 Oracle Academy Java 编程。



#### Oracle Technology Network / Java / Java SE / Downloads





Community Overview Downloads Documentation **Technologies** Training

#### Java SE Downloads







NetBeans with JDK 8

#### Java Platform, Standard Edition

#### Java SE 9.0.4

Java SE 9.0.4 includes important bug fixes. Oracle strongly recommends that all Java SE 9 users upgrade to this release.

Learn more >

- Installation Instructions
- Release Notes
- Oracle License
- Java SE Licensing Information User Manual
  - Includes Third Party Licenses
- Certified System Configurations
- Readme

**JDK** DOWNLOAD **±** 

Server JRE DOWNLOAD #

JRE DOWNLOAD 🛨

#### Which Java package do I need?

- Software Developers: JDK (Java SE Development Kit). For Java Developers. Includes a complete JRE plus tools for developing, debugging, and monitoring Java applications.
- Administrators running applications on a server: Server JRE (Server Java Runtime Environment) For deploying Java applications on servers. Includes tools for JVM monitoring and tools commonly required for server applications, but does not include browser integration (the Java plug-in), auto-update, nor an installer. Learn more >
- End user running Java on a desktop: JRE: (Java Runtime Environment). Covers most end-users needs. Contains everything required to run Java applications on your system.





Linux macOS

Windows

Solaris SPARC



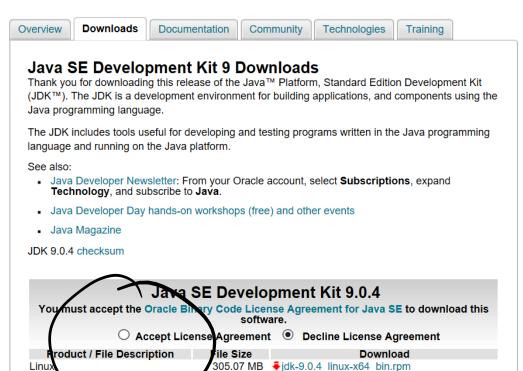






#### Oracle Technology Network / Java / Java SE / Downloads





206.97 MB ₱jdk-9.0.4\_solaris-sparcv9\_bin.tar.gz

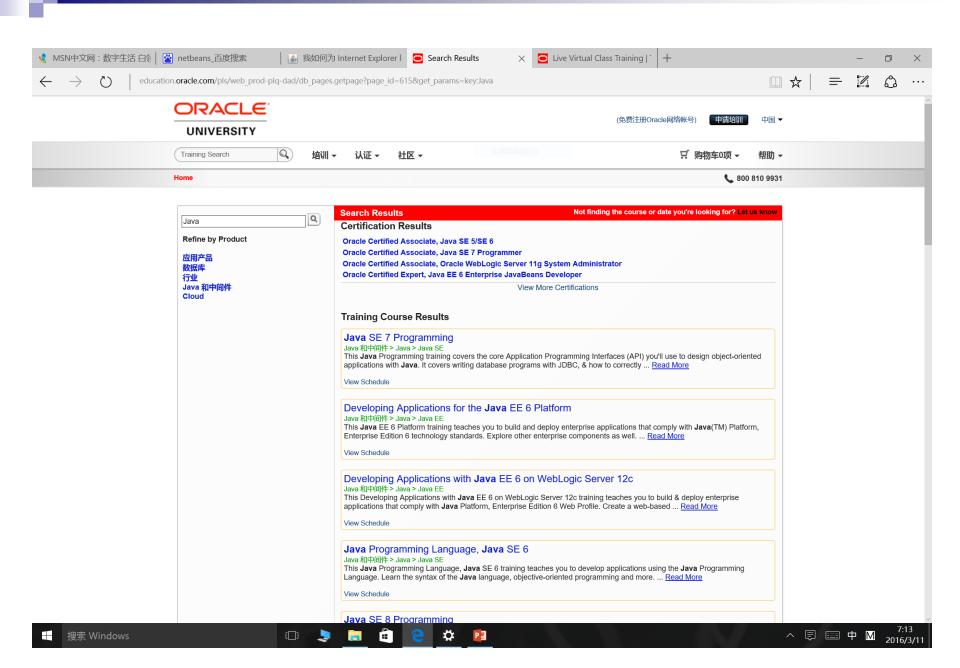
382.11 MB • jdk-9.0.4 osx-x64 bin.dmg

#### Java SDKs and Tools

- Java SE
- Java EE and Glassfish
- Java ME
- Java Card
- NetBeans IDE
- Java Mission Control

#### Java Resources

- Java APIs
- Technical Articles
- Demos and Videos
- Forums
- Java Magazine
- Developer Training
- Tutorials
- Java.com









#### ✓ 您已成功安装 Java

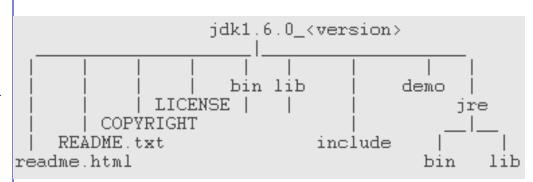
系統将提示您 Java 更新何时可用。请始终安装更新以获取最新的性 能和安全性改进。

有关更新设置的详细信息

单击 "关闭" 后, 浏览器将会打开以便您可以验证 Java 是否工作正 常。

关闭(C)

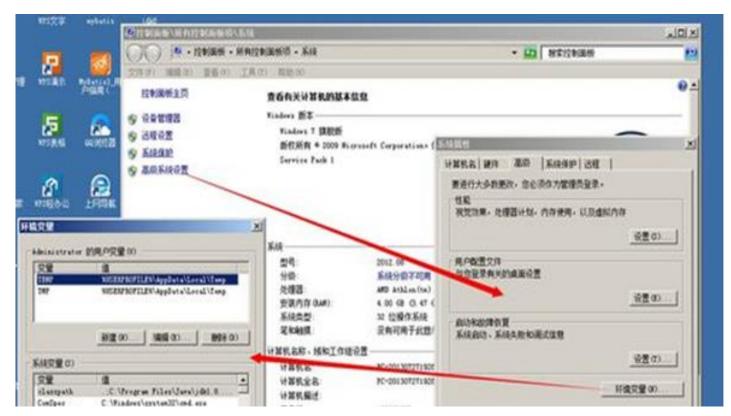
- bin目录是一些执行文件, Java的编译器、解释器和工具。
- demo 目录是各种演示的实例。
- lib目录保存库文件。
- include 目 录 下 是 Win32子目录,都是本 地方法文件。
- jre目录是Java程序运 行环境的根目录。



### Java SDK安装后的目录结构

## 设置环境变量

- 在Window下 (windows7-10), 控制面板
  - →高级系统设置→环境变量



有些IDE如eclipse可动态配置环境变量,无需人工设置

## 设置环境变量

■配置 classpath 变量 (tools.jar, dt.jar),新建,输入变量值:

.;D:\Java\jdk1.8.0\_05\lib\tools.jar;D:\Java\jdk1.8.0\_05\lib\dt.jar



## 设置环境变量

■配置path变量,选择jdk8的安装目录,输入变量值:

;D:\Java\jdk1.8.0 $_05$ \bin



### 测试 点击'开始-运行',输入cmd 命令 打开窗口,输入javac



- 如果将Java执行环境比喻为操作系统,
  - □设path变量是为了让操作系统找到指定的工具程序(以Windows来说就是找到.exe文件)
  - □设置classpath的目的就是让Java执行环境找到指定的Java程序(包括.class文件和jar中的classpath设置)。

- 例:
  - □ 在 "系统变量"下单击"新建"按钮,在"变量名" 文本框中输入Classpath,在"变量值"文本框中输入 Java类文件的位置。
  - ".;C:\Program Files\Java\jdk8\lib\tools.jar; C:\Program Files\Java\jdk8\lib\rt.jar"
- 每一路径中间必须以;作为分隔。 (根据平台的不同分隔符略有不同,类 unix 的系统基本上都是 ":", windows 多是 ";".)

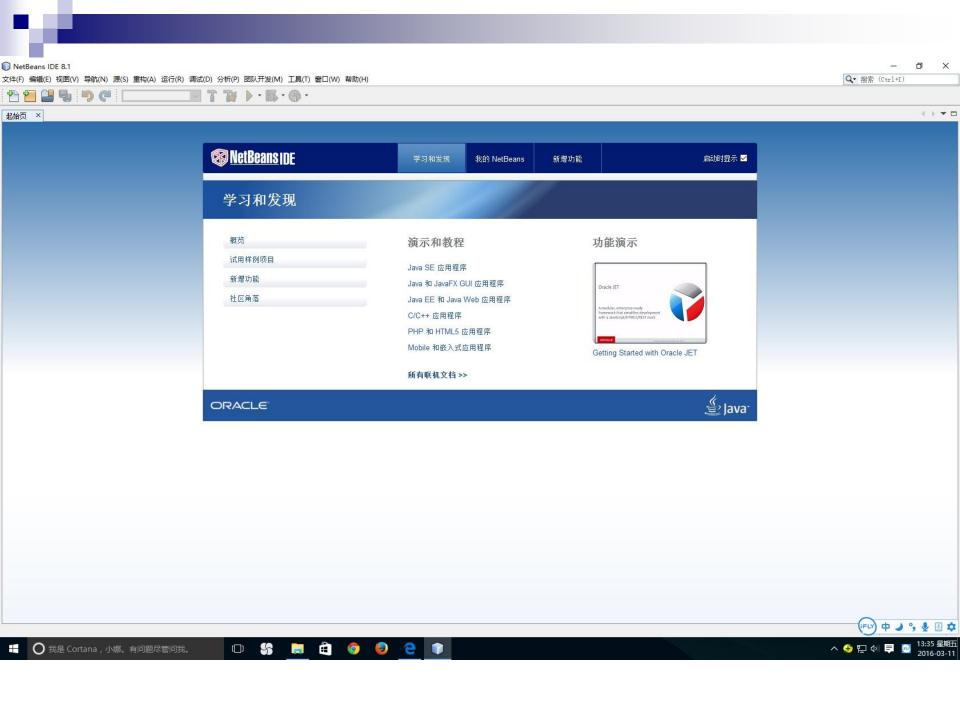
- JDK默认到当前工作目录(上面的.设置), 以及JDK的lib目录中寻找Java程序。所以 如果Java程序是在这两个目录中,则不必 设置Classpath变量也可以找得到.
- 非必须步骤,很多集成环境会自动搜寻设置 ,如netbean、eclipse等。



- JDK在安装的过程当中将在注册表会生成如下3个项目:
- HKEY\_LOCAL\_MACHINE\SOFTWARE\JavaSof t\Java Development Kit
- HKEY\_LOCAL\_MACHINE\SOFTWARE\JavaSof t\Java Plug-in
- HKEY\_LOCAL\_MACHINE\SOFTWARE\JavaSoft\ t\Java Runtime Environment

٧

■同时,JDK安装程序将会把java.exe,javaw.exe,javareg.exe这3个可执行文件拷贝到%SystemRoot%\system32目录下,由于%SystemRoot%\system32被操作系统缺省的设置为最高优先权的PATH搜索路径,因此可保证用户在命令行任何目录下可运行java.exe来启动JVM。



М

- ■.java是源程序代码文件
- .class文件是编译后的字节码文件
- ■.jar是JAVA程序打包以后的文件
- .manifest是资源配置文件

М

- ■学习好java,首先应该熟悉一些常用的jdk的命令,比如:
  - □ javac、java、javadoc、javap、jar、javaw、 appletviewer、htmlConverter、native2ascii、 serialver等。



### Java

□用来执行class文件的,关联控制台。

### Javac

- □java编程语言编译器。将类和接口的定义编译 成字节代码的class文件。
- □语法: javac [options] [sourcefiles] [@files]

## м

### ■ Javac可能的选项包括:

- □ -g 生成所有调试信息
- □ -g:none 生成无调试信息
- □ -g:{lines,vars,source} 生成只有部分调试信息
- □ -O 优化;可能妨碍调试或者增大类文件
- □ -nowarn 生成无警告
- □ -verbose 输出关于编译器正在做的信息
- □ -deprecation 输出使用了不鼓励使用的API的源程序位置
- □ -classpath <路径> 指定用户类文件的位置
- □ -sourcepath <路径> 指定输入源文件的位置
- □ -bootclasspath <路径> 覆盖自举类文件的位置
- □ -extdirs <目录(多个)> 覆盖安装的扩展类的位置
- □ -d <目录> 指定输出类文件的位置
- □ -encoding <编码> 指定源文件中所用的字符集编码
- □ -target <版本> 生成指定虚拟机版本的类文件
- -help Print a synopsis of standard options

M

- 如果所使用的JDK工具程序具有Classpath命令选项,则可以在执行工具程序时一并指定Classpath。例如:javac -classpath classpath1;classpath2...
   其中classpath1、classpath 2是实际要指定的路径。
- 也可以在命令符模式下执行以下的命令,直接设置目前的 环境变量,包括Classpath变量(这个设置在下次重新打开 命令符模式时就不再有效):

set CLASSPATH=%CLASSPATH%;classpath1;classpath3...



### Javaw

- □用来执行class文件的,但可以将dos窗口隐藏掉,不关联控制台。
- □写个批处理执行java程序时,会出现一个窗口,若不想让这个窗口出现,可在执行语句处将 java classname换成:

start javaw classname

便可以隐藏控制台窗口.

### Jar

□ 多用途的存档及压缩工具,是个java应用程序,可将多个文件合并为单个JAR归档文件。语法: jar [命令选项] [mainfest] destination input-file [input-files]

### □选项:

- -c 创建新的存档
- -t 列出存档内容的列表
- -x 展开存档中的命名的(或所有的)文件
- -u 更新已存在的存档
- -v 生成详细输出到标准输出上
- -f 指定存档文件名
- -m 包含来自标明文件的标明信息
- -0 只存储方式; 未用ZIP压缩格式
- -M 不产生所有项的清单(manifest)文件
- -i 为指定的jar文件产生索引信息
- -C 改变到指定的目录,并且包含下列文件:
- 如果一个文件名是一个目录,它将被递归处理。
- 清单(manifest)文件名和存档文件名都需要被指定,按'm'和 'f'标志指定的相同顺序。

## ٧

### Jar

- □ 示例:
  - jar tf file.jar //显示一个jar文件中的文件列表
  - jar cf file.jar \*.class //将当前目录下的所有Class文件打包成新的JAR 文件
  - jar cvf file.jar \* //将当前目录下的所有文件放进一个已存在的JAR文件中。
  - 示例:将两个class文件存档到一个名为 'classes.jar' 的存档文件中:
    - □ jar cvf classes.jar Foo.class Bar.class
  - 示例: 用一个存在的清单(manifest)文件 'mymanifest' 将 foo/ 目录下的所有文件存档到一个名为 'classes.jar' 的存档文件中:
    - □ jar cvfm classes.jar mymanifest -C foo/ .
- □ option的顺序决定了jar文件名和manifest文件名的顺寻,如使用'cfm',manifest文件应该紧随jar后面。正确顺序应该是: jar cfm aa.jar manifest.txt Files.class

r

- 解压 JAR 文件
  - □JAR 文件实际上就是 ZIP 文件,所以可以使用常见的一些解压 ZIP 文件的工具来解压 JAR 文件,如 Windows 下的 WinZip、WinRAR 等和 Linux 下的 unzip 等。

۲

- 使用 WinZip 或者 WinRAR 等工具创建 JAR 文件
  - □ JAR 文件就是包含了 META-INF/MANIFEST.MF 的 ZIP 文件,所以,只需要使用 WinZip、WinRAR 等工具创建所需要 ZIP 压缩包,再往这个 ZIP 压缩包中添加一个包含 MANIFEST.MF文件的 META-INF 目录即可。

## м

# 标明文件(manifest.mf)

- 不能用汉字,每行不能超过72字节,不是 key 的每行前面要留一个空格,最后一行为空。
- Manifest-Version: 1.0
- Main-Class: cn.com.DemForm1
- Class-Path: ./lib/swing-layout-1.0.3.jar
- ./lib/AbsoluteLayout.jar
- ./jar/poi-contrib-3.2-FINAL-20081019.jar
- ./jar/jcommon-1.0.17.zip
- ./jar/jxl.jar
- ./jar/swt-debug.jar
- ./jar/swt.jar

## м

## 标明文件(manifest.mf)

- 一般属性
- 1. Manifest-Version

用来定义manifest文件的版本,例如: Manifest-Version: 1.0

2. Created-By

声明该文件的生成者,一般该属性是由jar命令行工具生成的,例如: Created-By: Apache Ant 1.5.1

3. Signature-Version

定义jar文件的签名版本

4. Class-Path

应用程序或者类装载器使用该值来构建内部的类搜索路径

■ 应用程序相关属性

#### Main-Class

定义jar文件的入口类,该类必须是一个可执行的类,一旦定义了该属性即可通过 java -jar x.jar来运行该jar文件。

## .

## 标明文件(manifest.mf)

- 小程序(Applet)相关属性
- 1. Extendsion-List

该属性指定了小程序需要的扩展信息列表,列表中的每个名字对应以下的属性

- 2. <extension>-Extension-Name
- 3. <extension>-Specification-Version
- 4. <extension>-Implementation-Version
- 5. <extension>-Implementation-Vendor-Id
- 5. <extension>-Implementation-URL
- 扩展标识属性

#### **Extension-Name**

该属性定义了jar文件的标识,例如Extension-Name: Struts Framework

## w

# 标明文件(manifest.mf)

- 包扩展属性
- 1. Implementation-Title 定义了扩展实现的标题
- 2. Implementation-Version 定义扩展实现的版本
- 3. Implementation-Vendor 定义扩展实现的组织
- 4. Implementation-Vendor-Id 定义扩展实现的组织的标识
- 5. Implementation-URL: 定义该扩展包的下载地址(URL)
- 6. Specification-Title 定义扩展规范的标题
- 7. Specification-Version 定义扩展规范的版本
- 8. Specification-Vendor 声明了维护该规范的组织
- 9. Sealed 定义jar文件是否密封,值可以是true或者false

## м

# 标明文件(manifest.mf)

#### ■ 包扩展属性

包密封

密封 JAR 文件中的一个包意味着在这个包中定义的所有类都必须在同一个 JAR 文件中找到,可以增强打包类之间的版本一致性,还提供了防止代码篡改的手段。

在 JAR 的 manifest 文件中为包添加一个 Name 头,然后加上值为"true"的 Sealed 头。与可执行的 JAR 一样,可以在创建 JAR 时,通过指定一个具有适当头元素的manifest 文件密封一个 JAR,如下所示:

Name: com/samplePackage/

Sealed: true

Name 头标识出包的相对路径名。它以一个"/"结束以与文件名区别。在 Name 头后面第一个空行之前的所有头都作用于在 Name 头中指定的文件或者包。在上述例子中,因为 Sealed 头出现在 Name 头后并且中间没有空行,所以 Sealed 头将被解释为只应用到包 com/samplePackage 上。

如果试图从密封包所在的 JAR 文件以外的其他地方装载密封包中的一个类,那么 JVM 将 抛出一个 SecurityException。



# 标明文件(manifest.mf)

#### ■ 签名相关属性

Name: javax/mail/Address.class

Digest-Algorithms: SHA MD5

SHA-Digest: AjR7RqnN//cdYGouxbd06mSVfI4=

MD5-Digest: ZnTIQ2aQAtSNIOWXI1pQpw==

这段内容定义类签名的类名、计算摘要的算法名以及对应的摘要内容(使用BASE64方法进

行编码)

## 1

## 标明文件(manifest.mf)

### ■ 自定义属性

除了前面提到的一些属性外,也可以在MANIFEST.MF中增加自己的属性以及响应的值, 例如J2ME程序jar包中就可能包含着如下信息

MicroEdition-Configuration: CLDC-1.0

MIDlet-Name: J2ME\_MOBBER Midlet Suite

MIDlet-Info-URL: http://www.javayou.com

MIDlet-Icon: /icon.png

MIDlet-Vendor: Midlet Suite Vendor MIDlet-1: mobber,/icon.png,mobber

MIDlet-Version: 1.0.0

MicroEdition-Profile: MIDP-1.0

MIDlet-Description: Communicator

■ java.util.jar包提供了用于处理这些信息的API,可以通过给JarFile传递一个jar文件的路径,然后调用JarFile的getManifest方法来获取Manifest信息。

■课后:

□了解mainfest格式要求及不同要求的打包方法



### Javah

□C头文件和Stub文件生成器。产生可以调用 Java过程的C过程,或建立能被Java程序调用 的C过程的头文件。

### □语法:

- javah [ 命令选项 ] fully-qualified-classname. . .
- javah\_g [ 命令选项 ] fully-qualified-classname. . .

## 7

### Javah

- □-help 打印该帮助信息
- □ -classpath <path> 类的加载路径
- □ -bootclasspath <path> 自举类的加载路径
- □ -d <dir> 输出目录
- □ -o <file> 输出文件(仅能使用 -d 或 -o 之一)
- □-jni 生成 JNI 风格的头文件(缺省)
- □ -old 生成 JDK1.0 风格的头文件
- □ -stubs 生成 stubs 文件
- □ -version 打印版本信息
- □-verbose 输出有关本命令所做工作的信息
- □-force 始终写输出文件
  - 指定 <classes> 时必须使用全名(例如 java.lang.Object)。

## M

### Javap

- □Java反汇编器,显示编译类文件中的可访问功能和数据,同时显示字节代码含义
- □类文件解析器,其输出取决于所用选项。若没用选项,javap将输出传递给它的类的 public 域及方法,并输出到标准输出设备上。
- □语法:

javap [命令选项] class...

## M

### Javadoc

- □Java API文档生成器,解析Java源文件中的声明和文档注释,并产生API文档及相应的HTML页
- □描述公有类、保护类、内部类、接口、构造函数、方法和域,依赖于java编译器完成其工作
- □语法:

javadoc [ 命令选项 ] [ 包名 ] [ 源文件名 ] [ @files ]

## м

- □ -overview <file> 读取 HTML 格式的概述文档
- □ -public 仅显示 public 类和成员
- □ -protected 显示 protected/public 类和成员(缺省)
- □ -package 显示 package/protected/public 类和成员
- □ -private 显示所有类和成员
- □ -help 显示命令行选项
- □ -doclet <class> 通过候选 doclet 生成输出
- □ -docletpath <path> 指定 doclet 类文件的查找位置
- □ -sourcepath <pathlist> 指定源文件的查找位置
- □ -classpath <pathlist> 指定用户类文件的查找位置
- -exclude <pkglist> Specify a list of packages to exclude
- -subpackages <subpkglist> Specify subpackages to recursively load
- -breakiterator Compute 1st sentence with BreakIterator
- □ -bootclasspath <pathlist> 覆盖自举类加载器所加载的类文件的位置
- -source <release> Provide source compatibility with specified release
- □ -extdirs <dirlist> 覆盖已安装的扩展的位置
- □ -verbose 有关 Javadoc 所做工作的输出信息

## м

- □ -locale <name> 所用的 Locale,例如 en\_US 或 en\_US\_WIN
- □ -encoding <name> 源文件编码名称
- □-J<flag>将 <flag>直接传给运行时系统
- □ 由标准 doclet 提供:
- □ -d <directory> 输出文件的目标目录
- □-use 创建类和包的用法页
- □ -version 包含 @version 段
- □ -author 包含 @author 段
- docfilessubdirs Recursively copy doc-file subdirectories
- □-splitindex 将索引分为每个字母对应一个文件
- □ -windowtitle <text> 文档的浏览器窗口标题

## v

- □ -doctitle <html-code> 包含包索引页(首页)的标题
- □ -header <html-code> 包含每一页的页眉文本
- □ -footer <html-code> 包含每一页的页脚文本
- □ -bottom <html-code> 包含每一页的页底文本
- □ -link <url> Create links to javadoc output at <url>
- -linkoffline <url> <url>> Link to docs at <url> using package list at <url>>
- -excludedocfilessubdir <name1>:.. Exclude any docfiles subdirectories with given name.
- -group <name> <p1>:<p2>.. Group specified packages together in overview page

## м

| -nocomment Supress description and tags, generate only declarations.                             |
|--|
| -nodeprecated 不包含 @deprecated 信息   |
| -noqualifier <name1>:<name2>: Exclude the list of qualifiers from the output.</name2></name1>    |
| -nosince Do not include @since information   |
| -nodeprecatedlist 不生成不鼓励使用的列表  |
| -notree 不生成类层次   |
| -noindex 不生成索引   |
| -nohelp 不生成帮助链接  |
| -nonavbar 不生成导航栏   |
| -quiet Do not display status messages to screen  |
| -serialwarn Generate warning about @serial tag   |
| -tag <name>:<locations>:<header> Specify single argument custom tags</header></locations></name> |
| -taglet The fully qualified name of Taglet to register   |
| -tagletpath The path to Taglets  |
| -charset <charset> Charset for cross-platform viewing of generated</charset>                     |
| documentation.   |
| -helpfile <file> 包含帮助链接功能链接到目标的文件</file>   |
| -linksource Generate source in HTML  |
| -stylesheetfile <path> 改变所生成文档的样式的文件</path>  |
| -docencoding <name> 输出编码名称</name>  |

1

■ JavaDoc文档标记 javadoc注释以"/\*\*"开始,以"\*/"结束,里面 可以包含普通文本、HTML标记和javadoc 标记。javadoc只处理源文件中在类/接口定 义、方法、域、构造器之前的注释,忽略 位于其他地方的注释。

- ■课后:
  - □了解JavaDoc书写规范



### Appletviewer

- □调试applet的工具,可不用万维网浏览器环境
  - ,运行java的applet。
- □语法:
  - appletviewer [threads flag] [命令选项] urls...
    - 连接到url指向的文档或资源,在其自身的窗
  - 口显示引用的applet。



### htmlConverter

□命令转换工具,用来转换网页中的applet标签

0

□语法:

HtmlConverter [-option1 value1 [-option2 value2 [...]]] [-simulate] [filespecs]



### Native2ascii

- □将含有不是Unicode或Latinl字符的的文件转换 为Unicode编码字符的文件.
- □语法:

native2ascii [options] [inputfile [outputfile]]

如果省in/outputfile则使用标准输入/出设备。



### Native2ascii

□ Java 编译器和其它 Java 工具只能处理含有 Latin-1 和/或 Unicode 编码(udddd 记号)字符的文件。 native2ascii 将含有其它字符编码的文件转换成含 Latin-1 和/或 Unicode 编码字符的文件。若省略 outputfile,则使用标准输出设备输出。此外,如果也省略 inputfile,则使用标准输入设备输入。

### □命令选项

- -reverse 执行相反的操作: 将含 Latin-1 和/或 Unicode 编码字符的文件转换成含本地编码字符的文件。
- -encoding[encoding\_name] 指定转换过程使用的编码名称。 缺省的编码从系统属性 file.encoding 中得到。



### Serialver

- □返回serialverUID。
- □语法:

### serialver [show]

命令选项show是用来显示一个简单的界面。输入完整的类名按Enter键或"显示"按钮,可显示serialverUID。

serialVerUID 从 JDK 5开始强迫使用. 用于序列化时版本的跟踪与维护。