



内容提要

第1章 Java概述

Java简介

Java与C++的主要区别

JDK安装与配置

Java程序的编辑及编译运行

IDE—Eclipse的安装与配置



Java概述

- Java是一种编程语言，它提供了一个同时用于程序开发、应用和部署的环境。
- Java语言主要定位于网络编程，使得程序可以最大限度地利用网络资源。
- 名字来源

Java是印度尼西亚**爪哇岛**的英文名称，因盛产**咖啡**而闻名。Java语言中的许多库类名称，多与咖啡有关，如JavaBeans（咖啡豆）、NetBeans（网络豆）以及ObjectBeans(对象豆) 等等。SUN和JAVA的标识也正是一杯正冒着热气的咖啡。



Java的产生与发展

- 在Internet上出现的特别吸引人的事件就是Java语言和用Java编写的浏览器 HotJava。
- 1991年，SUN MicroSystem公司为在电视、控制烤面包箱等家用消费类电子产品上进行交互式操作而创造了一种名为Oak(橡树，Java之父James Gosling以他的办公室外的树而命名)的程序设计语言。Oak程序运行于虚拟机，同时允许程序在电视机、机顶盒等多平台上运行。
- 1994年，SUN用Oak语言开发了万维网浏览器 HotJava，Oak改名为Java，此后它逐渐成为 Internet上受欢迎的开发与编程语言，MS、IBM、Netscape、Novell、Apple、DEC、SGI等著名的计算机公司纷纷购买了Java语言的使用权。



Java的产生与发展

- 1995年，SUN在SunWorld大会首次公开介绍了Java和HotJava浏览器。
 - 重大突破—Netscape在它的浏览器中支持Java。
 - Oracle在其将发布的WWW网页系统中包含了一个兼容的Java浏览器。
 - 更广泛行业接受的信号，MS在其IE浏览器中支持Java。
- 1996年，SUN成立了Java业务集团，专门开发Java技术。



Java的产生与发展

- 1996年, JDK1.0。
- 1997年, JDK1.1, 主要特点是JDBC、RMI、内部类
- 1998年, JDK1.2 (通常被称为Java2), 包括J2SE (Standard Edition)、J2EE (Enterprise Edition)、J2ME (Micro Edition), 见证重大转变的最流行版本。主要特点是集合框架、JIT编译器、策略工具、Java基础类、Java二维类库和JDBC改进。
- 2000年, JDK1.3。
- 2002年, JDK1.4, 主要特点是XML处理、Java打印、支持日志、JDBC 3.0、断言和正则表达式处理。



Java的产生与发展

- 2004年，JDK1.5，主要特点是支持泛型、自动装箱、注释处理、Instrumentation。
- 2006年，JDK1.6，包括JavaSE、JavaEE、JavaME，主要特点是支持脚本语言、JDBC4.0、Java编译API并整合了Web服务。
- 2010年，Oracle收购Sun公司及其产品。现在Java由Oracle控制。
- 同年，Steve Jobs声称，苹果将来不再支持Java。



Java的产生与发展

- 2011年，JDK1.7，主要特点是支持动态语言、Java nio包、多重异常处理等。
- 2014年，JDK1.8，主要特点是默认接口方法、多重Annotation、API改进（比如流，日期API）等。
- 2017年，JDK1.9，主要特点是轻量级的JSON API、WebSockets、统一JVM日志、更多的编译器控制、移除过时的GC组合等。
- 2018年3月，JDK 10
- 2018年9月，JDK 11
- 2019年3月，JDK 12



Java 语言的特点

● 跨平台性

所谓的跨平台性，是指软件可以不受计算机硬件和操作系统的约束而在任意计算机环境下正常运行。这是软件发展的趋势和编程人员追求的目标。之所以这样说，是因为计算机硬件的种类繁多，操作系统也各不相同，不同的用户和公司有自己不同的计算机环境偏好，而软件为了能在这些不同的环境里正常运行，就需要独立于这些平台。



Java 语言的特点

● 跨平台性

而在Java语言中，Java自带的虚拟机很好地实现了跨平台性。Java源程序代码经过编译后生成二进制的字节码是与平台无关的，但是可被Java虚拟机（JVM）识别的一种机器码指令。JVM提供了一个字节码到底层硬件平台及操作系统的屏障，使得Java语言具备跨平台性。



Java 语言的特点

● 跨平台性

只要安装了 Java运行时系统 (JRE)，Java程序就可在任意的处理器上运行。这些字节码指令对应于 JVM中的表示，Java解释器得到字节码后，对它进行转换，使之能够在不同的平台运行。

- 在C、C++中，对于不同的平台，编译器对于简单数据类型如 int, float 等分别分配不同长度的字节数。例如：int在 IBM PC中为16位，在VAX-11中为32位，这导致了代码的不可移植性；
- 但在Java中，对于这些数据类型总是分配固定长度的位数，如对int型，它总占32位，这就保证了Java的平台无关性。



Java 语言的特点

● 面向对象

面向对象是指以对象为基本粒度，其下包含属性和方法。对象的说明用属性表达，而通过使用方法来操作这个对象。面向对象技术使得应用程序的开发变得简单易用，节省代码。Java是一种面向对象的语言，也继承了面向对象的诸多好处，如代码扩展、代码复用等。



Java 语言的特点

● 安全性

- 语言级安全性指Java的数据结构是完整的对象，这些封装过的数据类型具有安全性。
- 编译时要进行Java语言和语义的检查，保证每个变量对应一个相应的值，编译后生成Java类。
- 运行时Java类需要类加载器载入，并经由字节码校验器校验之后才可以运行。
- Java类在网络上使用时，对它的权限进行了设置，保证了被访问用户的安全性。



Java 语言的特点

● 多线程

多线程在操作系统中已得到了最成功的应用。多线程是指允许一个应用程序同时存在两个或两个以上的线程，用于支持事务并发和多任务处理。Java除了内置的多线程技术之外，还定义了一些类、方法等来建立和管理用户定义的多线程。

程序设计者可以分别用不同的线程完成特定的行为，而不需要采用全局的事件循环机制，这样就很容易实现网络上的实时交互行为。



Java 语言的特点

● 简单性

- Java略去了运算符重载、多重继承等模糊的概念，并且通过实现自动垃圾收集大大简化了程序设计者的内存管理工作。
- Java源代码的书写不拘泥于特定的环境，可以用记事本、文本编辑器等编辑软件来实现，然后将源文件进行编译，编译通过后可直接运行，通过调试则可得到想要的结果。
- 当然，用Eclipse等IDE开发程序更方便。



Java 语言的特点

- 解释执行

Java解释器直接对Java的字节码进行解释执行（转换成对应于特定CPU的机器码）。字节码本身携带了许多编译时信息，使得连接过程更加简单。

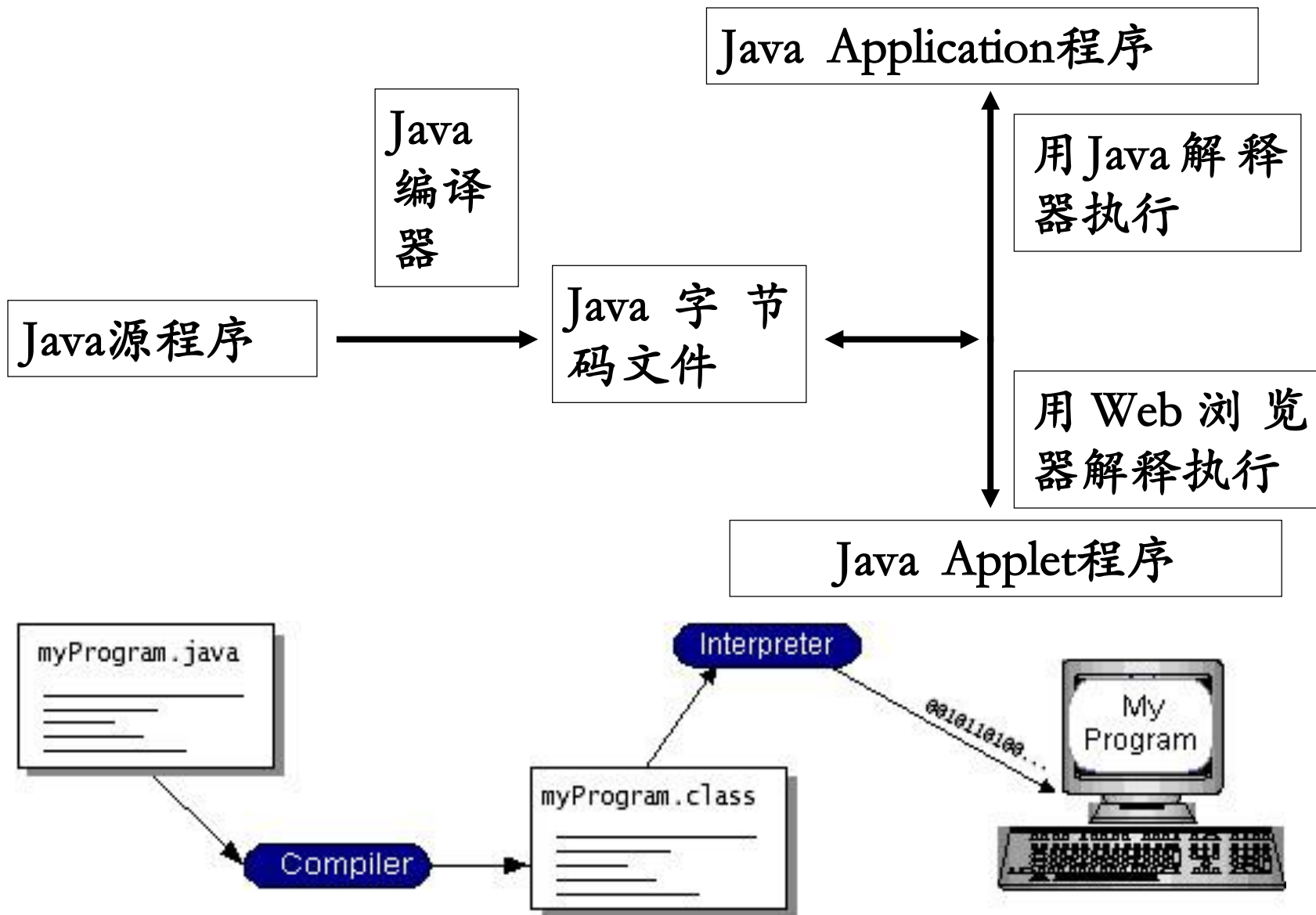
- 分布性

Java是面向网络的语言。它提供的类库（低级网络包）用于实现 Socket编程，处理 TCP/IP协议，用户可以通过 URL地址在网络上方便地访问其它对象。



Java 开发环境：程序编译执行过程

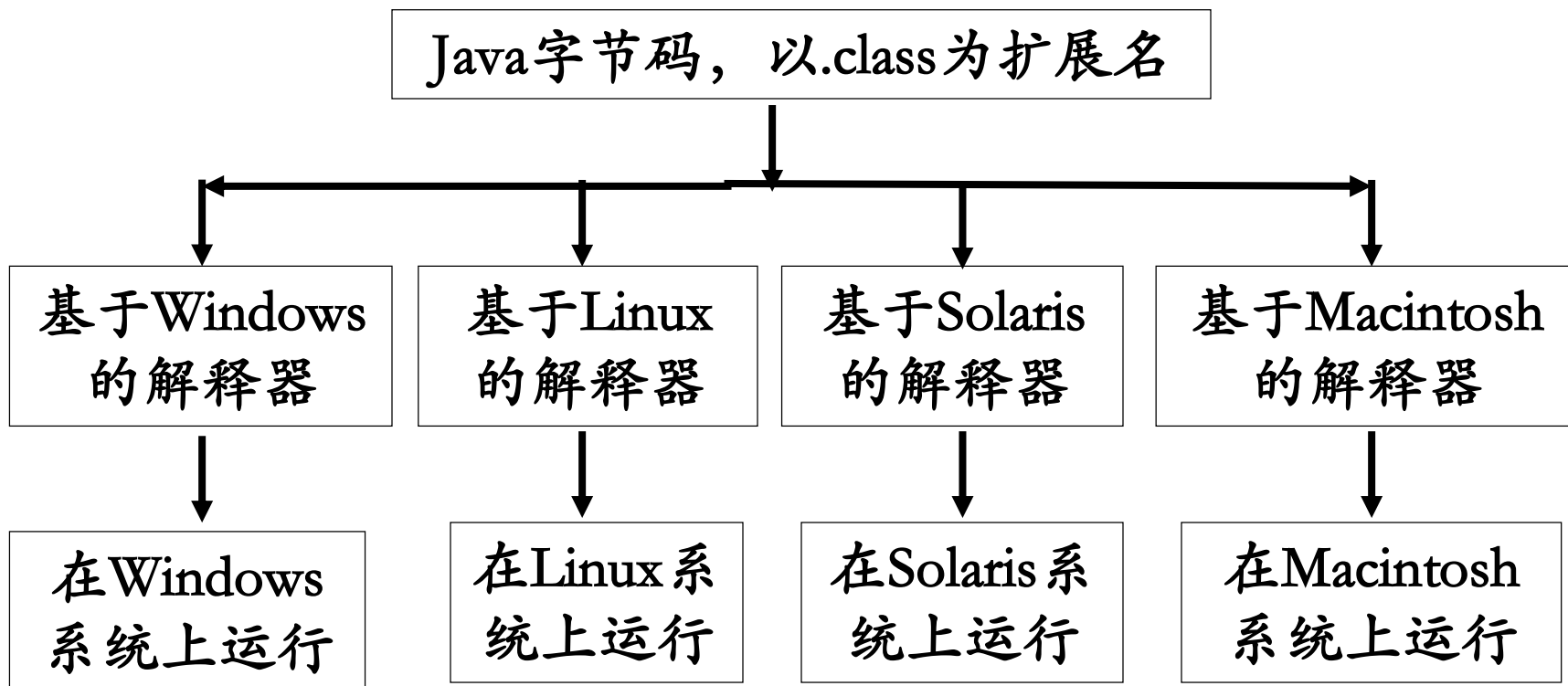
第1章 Java概述





Java 开发环境：一次编译，到处运行

第1章 Java概述



JVM把字节码解释成具体平台上的机器指令执行。

Java：“一次编译，到处运行”

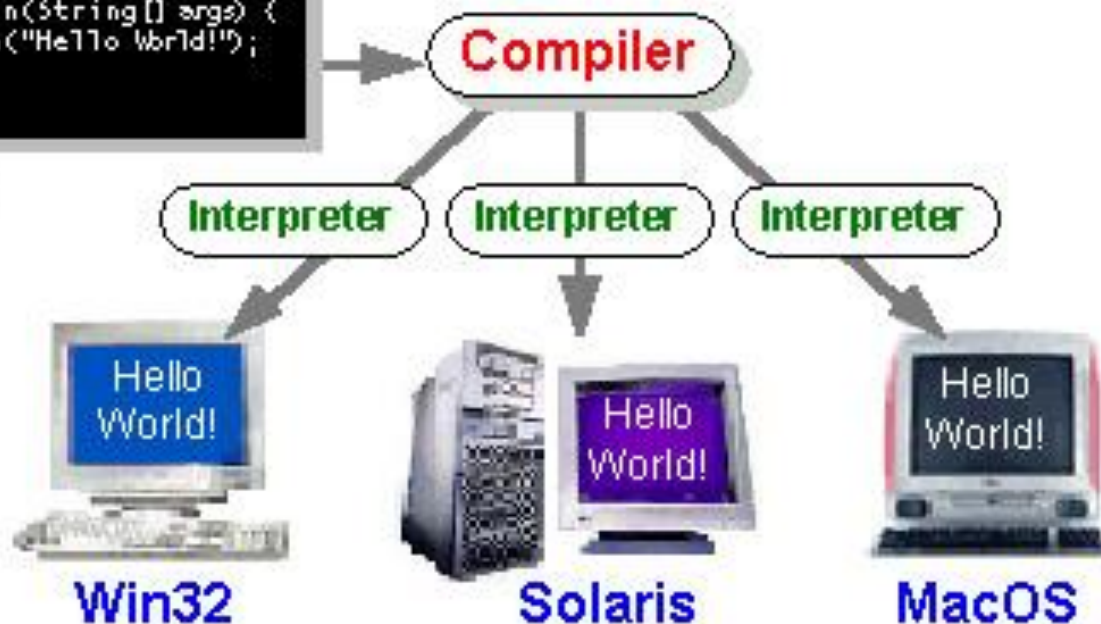


Java 开发环境：一次编译，到处运行

Java Program

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

HelloWorldApp.java

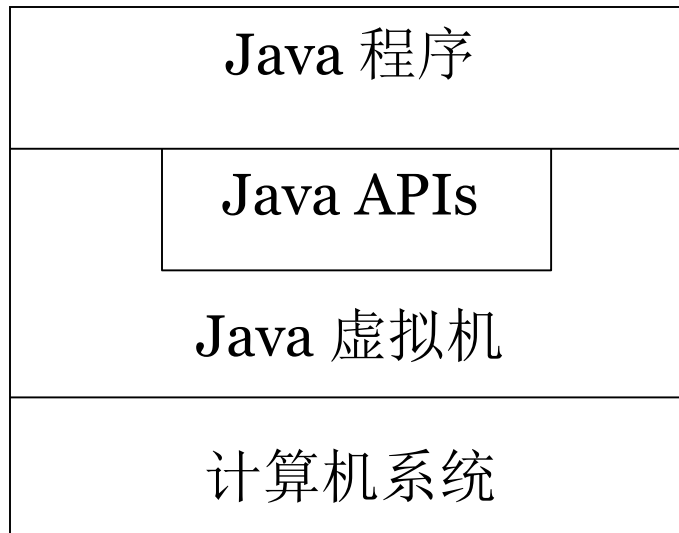


JVM把字节码解释成具体平台上的机器指令执行。

Java：“一次编译，到处运行”



Java 开发环境：Java平台



- Java APIs (应用程序接口)

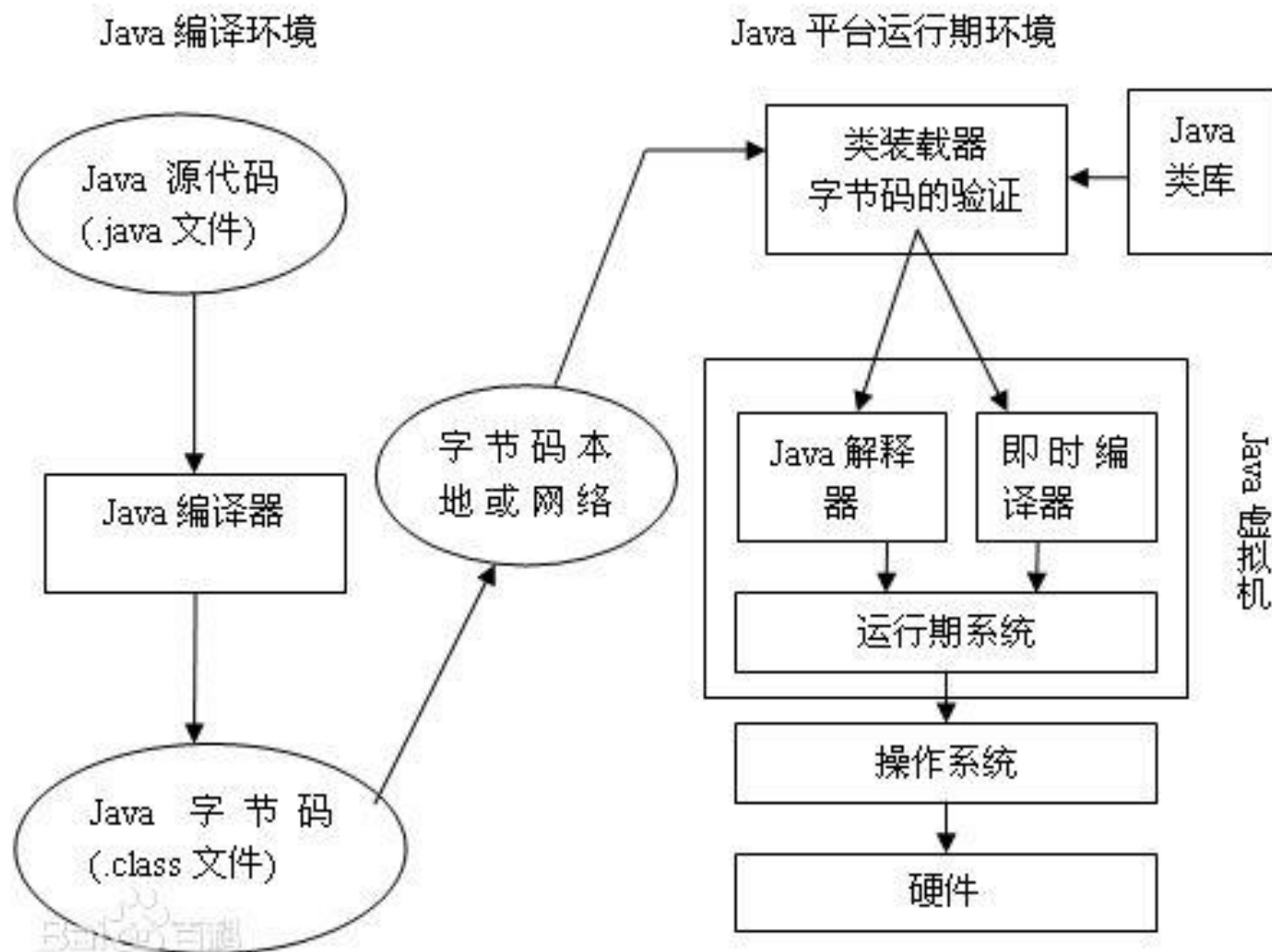
经过编译的，可在程序中使用的Java代码标准库。

- Java VM (虚拟机)

Java 程序由Java虚拟机程序执行（或解释执行）。



Java 开发环境：Java平台





Java 开发环境：Java平台

● Java虚拟机 (Java Virtual Machine, JVM)

- JVM是一种用于计算设备的规范，是一个虚构的计算机，是通过在实际的计算机上仿真模拟各种计算机功能来实现的。
- 一般的高级语言如果要在不同的平台上运行，需要编译成不同的目标代码。而引入JVM后，Java程序在不同平台上运行时不需要重新编译。
- Java语言使用JVM屏蔽了与具体平台相关的信息，只需编译成在JVM上运行的目标代码（字节码），就可以在多种平台上不加修改地运行。JVM在执行字节码时，把字节码解释成具体平台上的机器指令执行。



Java 开发环境：Java平台

- Java SDK(Software Development Kit)

- Standard Edition (JavaSE, 标准版)、Enterprise Edition (JavaEE, 企业版)、Micro Edition (JavaME, 微型版/移动版)

- Java SE——Java SDK的核心部分

- 开发工具：编译器、调试器、文档制作工具。
- 运行环境：Java 虚拟机、组成Java平台API的类、帮助文档。



Java 开发环境：Java平台

● JDK主要包括

- Javac: Java编译器, 用来将java程序编译成Bytecode。
- Jdb: Java调试器, 用来调试java程序。
- Java: Java解释器, 执行已经转换成Bytecode的java应用程序。
- Appletviewer: Applet解释器, 用来解释已经转换成Bytecode的java小应用程序。
- Javap: 反编译, 将类文件还原回方法和变量。
- Javadoc: 文档生成器, 创建HTML文件。
- Jar: 文件 (一般是类及其所在目录即包) 压缩打包。



Java 应用分类

● 应用程序(Application)

典型的应用程序可以在具备Java运行环境（JRE）的设备中独立运行，它又分为：

- GUI应用程序：即图形用户界面程序，可实现丰富的输入界面和输出显示。
- 命令行程序：无需界面，只需在命令行下运行，运行结果只在后台发生变化，可以将输出存放到文件中。
- 嵌入式应用程序：Java的平台独立性决定了它可以嵌入到不同的设备中，且只需具备必要的JRE即可。



Java 应用分类

● Servlet组件(即服务器端应用程序)

- Servlet是服务端的组件，用来收集客户端的数据输入（客户请求），对数据进行处理之后，返回相应的响应给客户。它主要用来实现与客户端的交互。
- Servlet组件为服务器和基于Web的客户（HTML页面中的表单或Applet）之间的通信提供了一条更为简单的途径。



Java 应用分类

● Applets小应用程序

- Applets应用于网络上，嵌入在HTML网页中，支持Java的浏览器都可以对它进行解释并运行。通常通过一个HTML标签<APPLET></ APPLET >来识别并运行Applets。
- 小应用程序的类在服务器端，当浏览器显示网页时，它随之下载到本地，由本地的浏览器载入运行。



Java 与 C++ 的主要区别

- 对于变量声明、参数传递、操作符、流控制等，Java使用了和 C++ 相同的传统，使得 C++ 的程序员能很方便地进行编程。
- 同时，Java为了 实现其简单、鲁棒、安全等特性，也摒弃了 C++ 中许多不合理的内容。



Java 与 C++ 的主要区别

● 全局变量

- Java程序中，不能在所有类之外定义全局变量，只能通过在一个类中定义公用、静态的变量来实现一个全局变量。例如：`Class GlobalVar{`

`public static global_var; }`

- Java对全局变量进行了更好的封装。而在C++中，依赖于不加封装的全局变量常常造成系统的崩溃。

● 类之外的函数

- Java中没有不属于类成员的函数。



Java 与 C++ 的主要区别

● 数据类型

- 在C++中对于不同的平台，编译器对于简单数据类型如 int, float 等分别分配不同长度的字节数，例如：int在 IBM PC中为16位，在VAX-11中为32位，这导致了代码的不可移植性。
- 但在Java中，对于这些数据类型总是分配固定长度的位数，如对int型，它总占32位，这就保证了Java的平台无关性。



Java 与C++的主要区别

● 结构、联合体和typedef

- C++中的结构和联合体中所有成员均为公有，这就带来了安全性问题。
- Java中不包含结构和联合，所有的内容都封装在类中。Java中也没有typedef。

● 异常处理

Java中不支持 C++中支持的goto(但goto还是保留的关键字)，而是通过异常处理语句 try、Catch、 final等来代替 C++中用goto来处理遇到错误时跳转的情况，使程序更可读且更结构化。



Java 与 C++ 的主要区别

● 指针

- 指针是C++中最灵活，也是最容易产生错误的数据类型。由指针所进行的内存地址操作常会造成不可预知的错误，同时通过指针对某个内存地址进行显式类型转换后，可以访问一个C++中的私有成员，从而破坏安全性，造成系统的崩溃。
- 而Java对指针进行完全的控制，程序员不能直接进行任何指针操作，例如把整数转化为指针，或者通过指针释放某一内存地址等。同时，数组作为类在Java中实现，良好地解决了数组访问越界这个C++中不作检查的错误。



Java 与 C++ 的主要区别

● 运算符重载、继承

- Java中只支持单重继承，没有多重继承，但可以实现多个接口；
- Java中没有运算符重载；

● 预处理

- C++中用宏定义来实现的代码给程序的可读性带来了困难。
- 在 Java 中，不支持宏，它通过关键字 `final` 来声明一个常量，以实现宏定义中广泛使用的常量定义。



Java 与 C++ 的主要区别

● 头文件

- C++ 中用头文件来声明类的原型以及全局变量、库函数等，在大的系统中，维护这些头文件是很困难的。
- 而 Java 不支持头文件，类成员的类型和访问权限都封装在一个类中，运行时系统对访问进行控制，防止对私有成员的操作。同时，Java 中用 `import` 语句来与其它类进行通讯，以便使用它们的方法。



Java 与 C++ 的主要区别

● 内存管理

- 在C++中，程序员通过运算符 `new` 和 `delete` 来分配和释放内存。再次释放已释放的内存块或未被分配的内存块，会造成系统的崩溃；同样，如果忘记释放不再使用的内存块也会逐渐耗尽系统资源。
- 而在Java中，所有的数据结构都是对象，通过运算符 `new` 为它们分配内存堆。通过 `new` 得到对象的处理权，而实际分配给对象的内存可能随程序运行而改变，Java对此自动地进行管理并且进行垃圾收集，有效防止了由于程序员的误操作而导致的错误，并且更好地利用了系统资源。



JDK的安装与配置

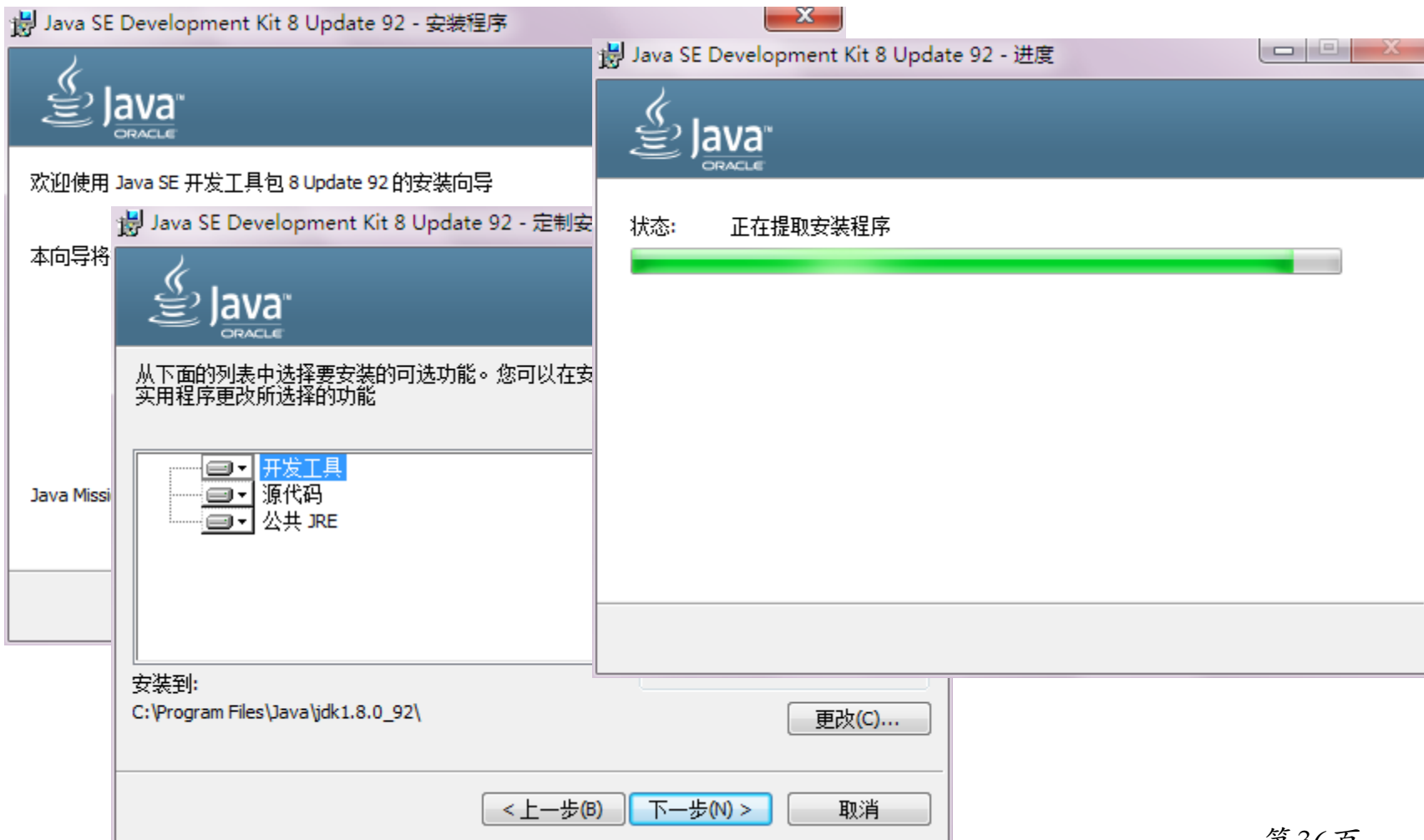
● JDK的下载与安装

- JDK(Java Development Kit) 是Oracle(1.0~1.6版本Sun) 免费发行的软件包，可以从Oracle网站免费下载：
<https://www.oracle.com/cn/downloads/>(需注册用户)
- 运用JDK，就可以对Java源程序进行编译和运行。
- 以Java SE开发工具包8 update 92为例，下载jdk-8u92-windows-i586.exe，安装到默认路径 C:\Program Files\Java\jdk1.8.0_92，也可以自定义路径，例如 C:\jdk1.8.0_92。



JDK的安装与配置

● JDK的下载与安装





JDK的安装与配置

第1章 Java概述

Java 安装 - 目标文件夹



目标文件夹

单击“更改”以将 Java 安装到其他文件夹。

安装到:

C:\Program Files\Java\jre1.8.0_92

更改(C)...

Java 安装 - 进度



状态: 安装 Java

ATMs, Smartcards, POS Terminals, Blu-ray Players
Set Top Boxes, Mobile Phones, Servers, St
Routers, Smart TVs, Servers, St
Automobiles, Smart Cars, Servers, St
Systems, Access Control Systems, Building Controls
Programs

3 Billion Devices Run Java



#1 Development Platform

ORACLE

Java SE Development Kit 8 Update 92 - 完成



Java SE Development Kit 8 Update 92 已成功安装

单击“后续步骤”访问教程, API 文档, 开发人员指南, 发布说明及更多内容, 帮助您开始使用 JDK。

后续步骤(N)

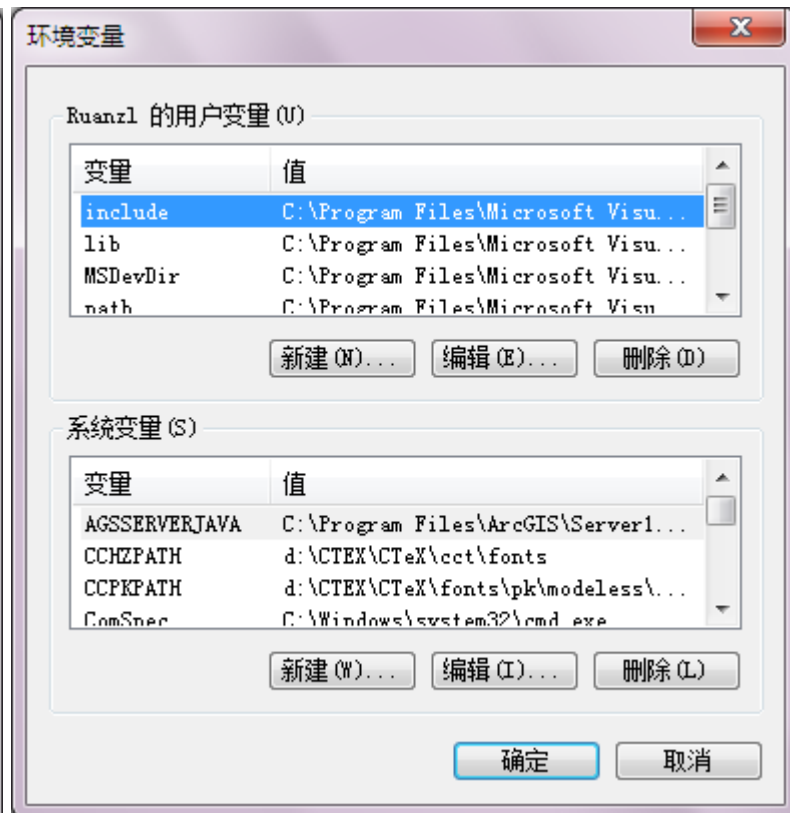
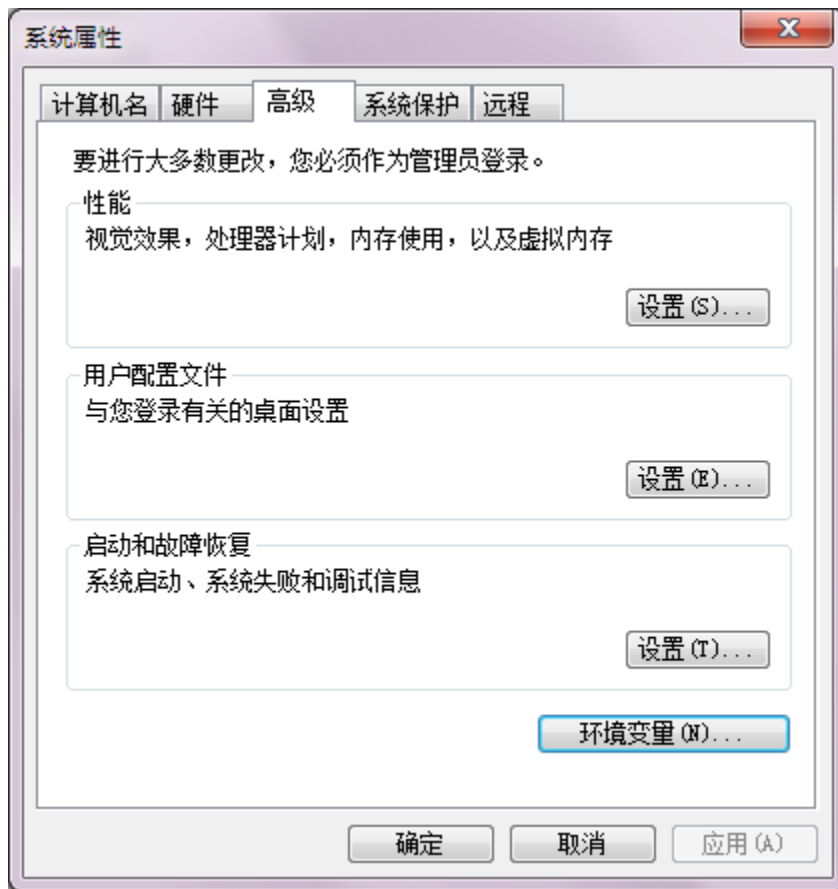
关闭(C)



JDK的安装与配置

● JDK的环境配置

JDK包安装完成后，需要设置计算机系统环境变量。

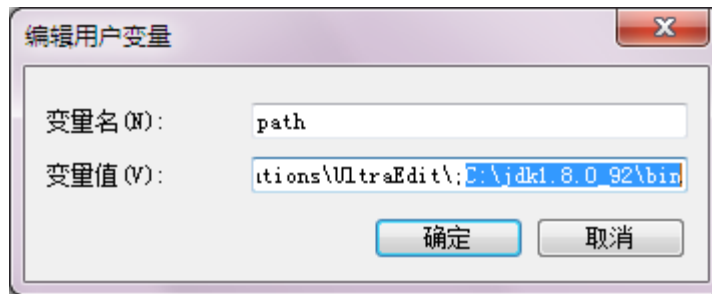
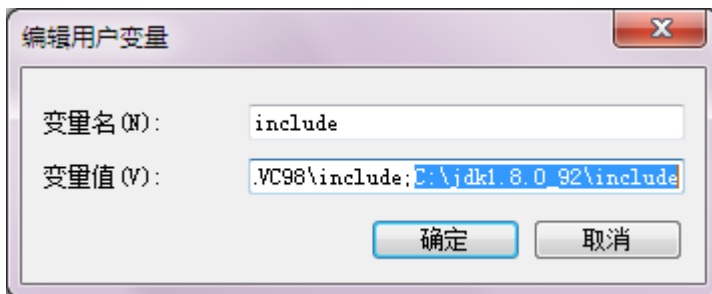




JDK的安装与配置

● JDK的环境配置

添加3个用户变量：include、lib和path，分别将JDK包安装之后的相应路径包含到这三个用户变量中。





Java程序的编辑及编译运行

● 编辑

- 用记事本或者专用的编辑工具如EditPlus等进行编辑，并将文件存为xxx.java。建议使用像EditPlus这样的编辑软件，~~编译~~使得代码更加清晰且风格良好。
- Eclipse等IDE



Java程序的编辑及编译运行

【例1】Java入门经典程序—HelloWorld.java

//程序文件名称为HelloWorld.java

```
public class HelloWorld{  
    public static void main(String args[]){  
        System.out.println("Hello World");  
    }  
}
```



Java程序的编辑及编译运行

● 编译

- 启动系统命令行cmd.exe，并进入程序所在目录（假定为D:\JavaExample），然后执行java编译命令：

```
javac HelloWorld.java
```

- 若编译不通过，会产生错误提示。
- 若编译通过，则没有任何提示，同时进入命令行等待状态。这时，命令行虽然没有提示，但在源程序的路径下生成一个新的文件为HelloWorld.class。



Java程序的编辑及编译运行

● 编译



```
管理员: 命令提示符

Microsoft Windows [版本 6.1.7600]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Ruanzl>D:

D:\>cd JavaExample

D:\javaExample>javac HelloWorld.java

D:\javaExample>
```



Java程序的编辑及编译运行

● 运行

- HelloWorld.class 文件就是编译后生成的类文件，运行此文件，需在命令行状态中键入命令：

`java HelloWorld`

- 按回车键，此时程序就会运行并输出“Hello World”。输出完毕，立即退出程序，进入命令行等待状态。



Java程序的编辑及编译运行

● 运行

```
C:\ 管理员: 命令提示符

Microsoft Windows [版本 6.1.7600]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Ruanzi1>D:

D:\>cd JavaExample

D:\javaExample>javac HelloWorld.java

D:\javaExample>java HelloWorld
Hello World

D:\javaExample>
```



Java程序的编辑及编译运行

- 命令Javac和Java都是JDK软件包自带的。
- 从JDK安装路径的bin目录下可以看到javac.exe，这是编译程序，源程序编译通过后就生成.class文件；而Java.exe就是载入类的运行程序，运行时根据源程序的指令要求产生正确的输出或结果。
- 如果没有进行环境配置，直接编译或者运行Java源程序，系统会提示找不到这些命令，所以必须进行环境配置后再使用。



Java程序的编辑及编译运行

【例2】 Applet小应用程序—HelloApplet.java

//程序文件名称为HelloApplet.java

```
import java.awt.Graphics;  
import java.applet.Applet;  
public class HelloApplet extends Applet{  
    public void paint(Graphics g ){  
        g.drawString ("Hello World!",50,25);  
    }  
}
```



Java程序的编辑及编译运行

【例2】Applet小应用程序—HelloApplet.java

小应用程序代码书写和编译完成后，无法独立运行，需要一个载体或者容器。下面的HTML网页代码就是小应用程序载入的容器。

```
<!-- 程序文件名称为HelloApplet.html -->
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> HTML Test Page</TITLE>
```

```
</HEAD>
```


<BODY>

HelloApplet will appear below in a Java enabled browser.

<APPLET

CODEBASE = "."

CODE = "HelloApplet.class"

NAME = "TestApplet"

WIDTH = 400

HEIGHT = 300

HSPACE = 0

VSPACE = 0

ALIGN = middle

>

</APPLET>

</BODY>

</HTML>



HelloApplet will appear below in a Java enabled browser.

Hello World!

今日特卖

今日直播

或在命令行启用Applet小
程序查看器appletviewer

管理员: 命令提示符 - appletviewer HelloApplet.h

```
D:\javaExample>appletviewer HelloApplet.html
```

小应用程序查看器: HelloApplet.class

小应用程序

Hello World!

已启动小应用程序。



IDE—Eclipse的安装与配置

● Eclipse简介

- Eclipse是著名的跨平台的可扩展的自由IDE。
- 最初主要用来Java语言开发，通过安装不同的插件Eclipse可以支持不同的计算机语言，比如C/C++、Python、COBOL、PHP、Android等开发。
- Eclipse的本身只是一个开源的框架平台，但是众多插件的支持使得Eclipse拥有其他功能相对固定的IDE软件很难具有的灵活性。
- 许多软件开发商以Eclipse为框架开发自己的IDE。



IDE—Eclipse的安装与配置

● Eclipse简介

- Eclipse最初是由IBM公司开发的替代商业软件Visual Age for Java的下一代IDE开发环境，2001年11月贡献给开源社区，现在它由非营利软件供应商联盟Eclipse基金会（Eclipse Foundation）管理。
- 版本：2007年3.3版->08年3.4版(代号Ganymede) -> 09年3.5版(Galileo) -> 10年3.6版(Helios) -> 11年3.7版(Indigo) -> 12年4.2版(Juno) -> 13年 4.3版(Kepler) -> 14年4.4版(Luna) -> 15年4.5版(Mars) -> 16年2月4.5.2版(Mars.2) -> 16年5月4.6版(Neon) RC1 -> 18-06 Oxygen (4.7) ->18-06 Photon (4.8, 光子) -> 18-09 (4.9)-> 18-12 (4.10)-> 19-03 (4.11)-> 19-06 (4.12)



IDE—Eclipse的安装与配置

● 下载

- 从官方网站下载 **Eclipse IDE for Java Developers**:
<https://www.eclipse.org/downloads/packages/>
- 解压缩后即可使用，免安装。



Eclipse IDE for Enterprise Java Developers

346 MB 567,337 DOWNLOADS

Tools for Java developers creating Enterprise Java and Web applications, including a Java IDE, tools for Enterprise Java, JPA, JSF, Mylyn, Maven, Git and more.
Click here to file a bug against Eclipse Web Tools Platform.
Click here to file a bug against Eclipse Platform.
Click here to file a bug against Maven integration for web projects.

Windows 64-bit
Mac Cocoa 64-bit
Linux 64-bit



Eclipse IDE for C/C++ Developers

235 MB 446,040 DOWNLOADS

An IDE for C/C++ developers with Mylyn integration.

Windows 64-bit
Mac Cocoa 64-bit
Linux 64-bit

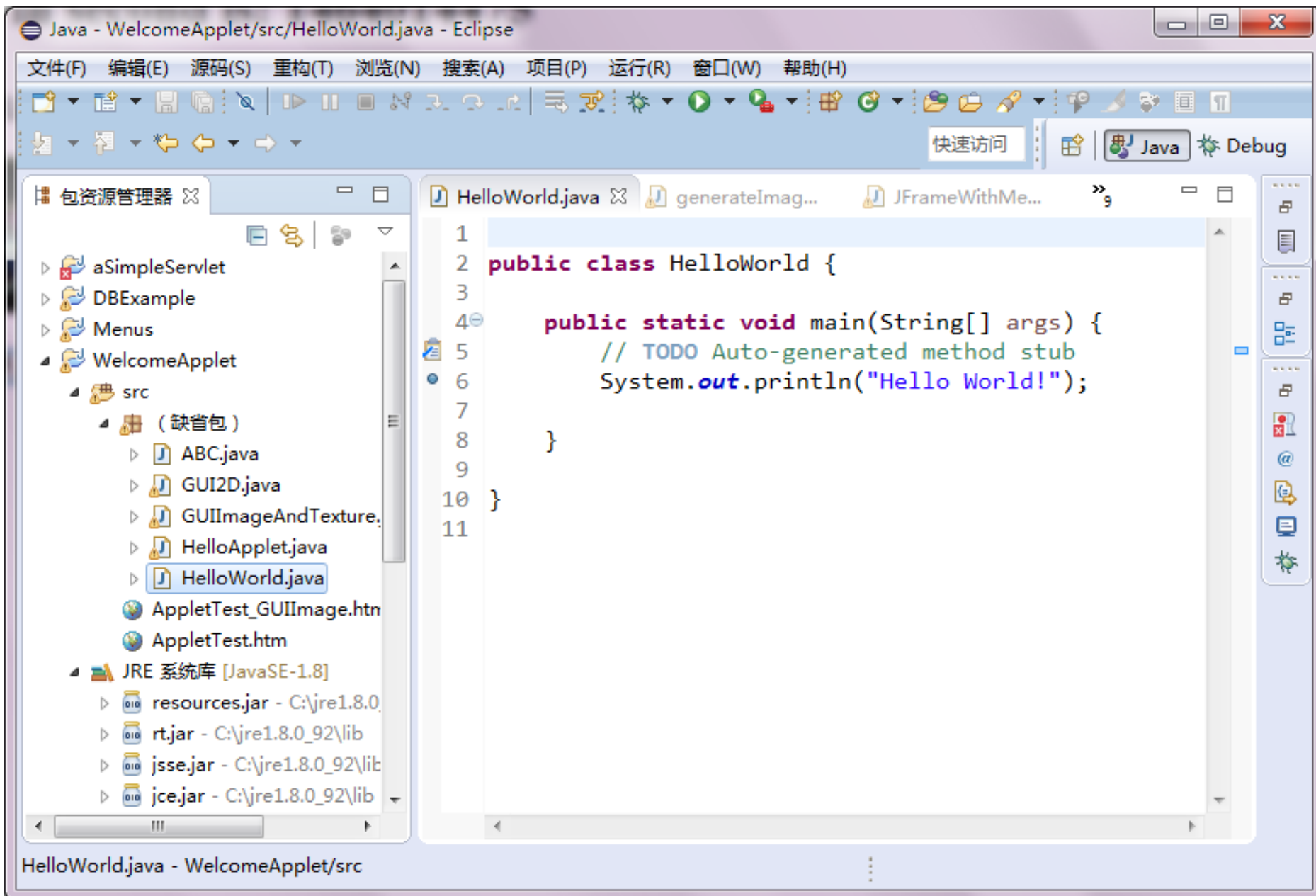


Eclipse IDE for Java Developers

195 MB 360,822 DOWNLOADS

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration

Windows 64-bit
Mac Cocoa 64-bit
Linux 64-bit





用IDE—Eclipse开发项目

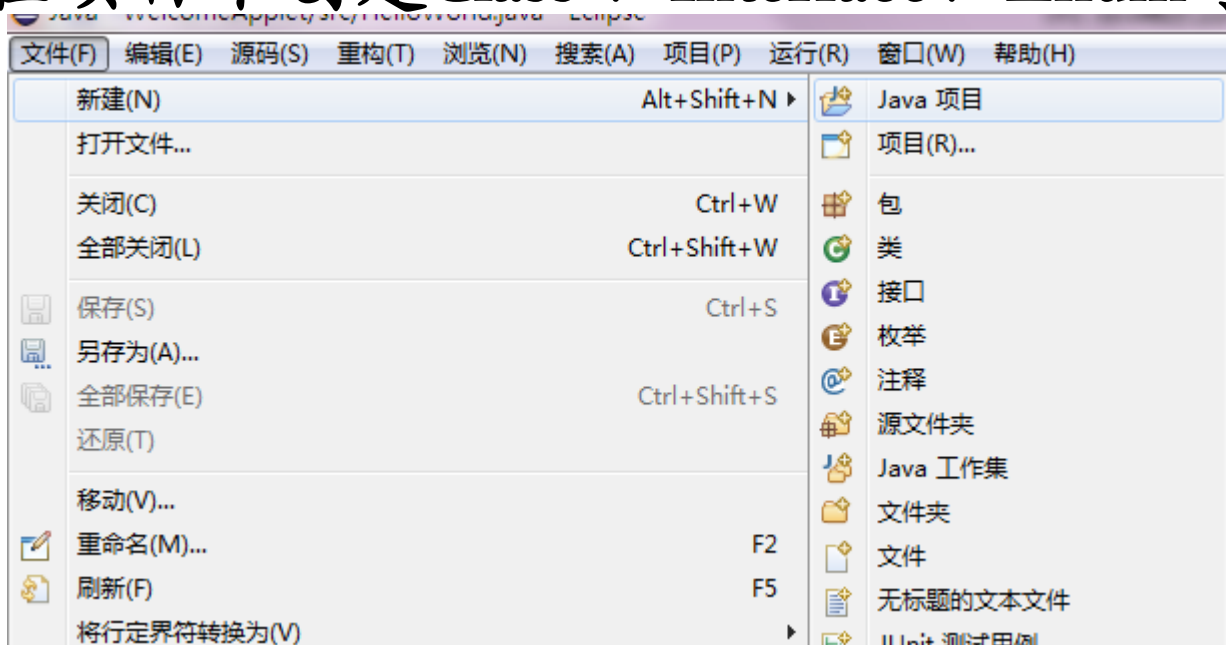
● 创建Java项目并配置JRE和项目布局 (Layout)



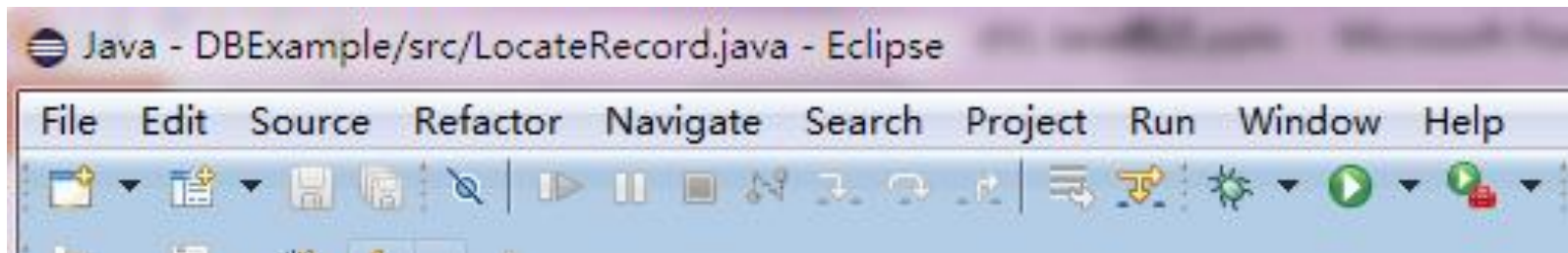


用IDE—Eclipse开发项目

- 在项目下创建Class、Interface、Enum等



- 调试 (Debug) 项目、运行 (Run) 项目





用IDE—Eclipse开发项目

● 代码编辑中的常用快捷键

- 块注释: Ctrl+Shift+/
 - 移除块注释: Ctrl+Shift+\
 - 自动对齐所选代码: Ctrl+I
 - 格式化所选代码或整个文档: **Ctrl+Shift+F**
 - 显示当前的包或类、接口、属性、方法的Java API帮助文档: **Alt+/**



习 题

- 简述Java的特点。
- 简述Java程序分类情况。
- 简述Java程序的运行机制。
- 安装和配置JDK、安装Eclipse。
- 编写应用程序，屏幕上输出“欢迎来到Java世界!”。
- 编写Applet，输出“欢迎来到Java世界!”。



谢谢大家!