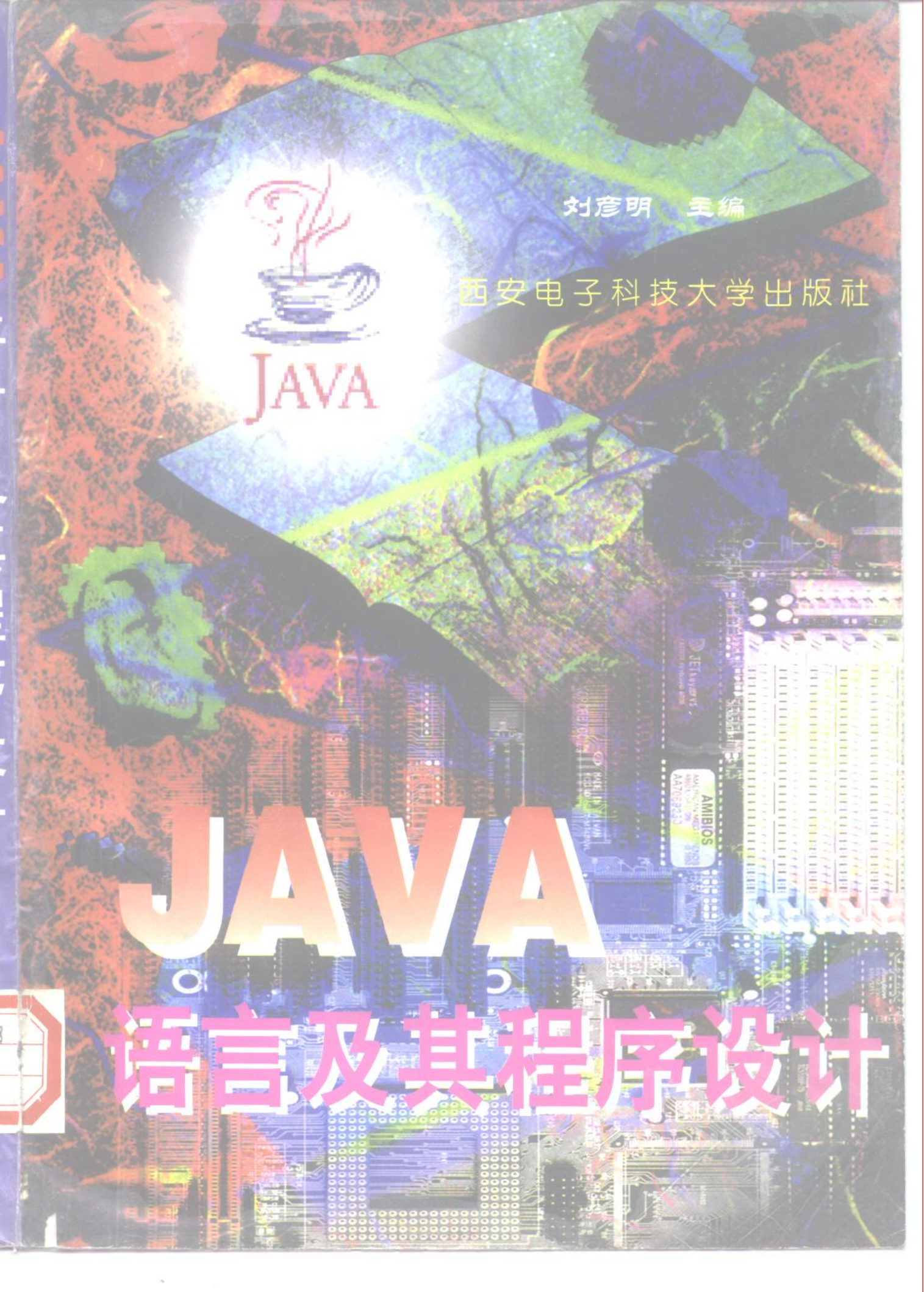
JAVA
刘彦明 主编
西安电子科技大学出版社
JAVA
语言及其程序设计

JAVA 语言及其程序设计

刘彦明　刘立群
芮　雨　赵　俊　李　鹏

西安电子科技大学出版社
1997

(陕)新登字 010 号

JS194/32

内容简介

本书介绍了新一代计算机语言——Java。Java 语言是一种基于面向对象的网络编程语言，它特别适合于 Internet 网上的应用开发，同时具有常用计算机程序设计语言的特点。

本书从面向对象的基本概念开始，由浅入深、按部就班地介绍了 Java 语言的语言要素、编程方法以及一些常用的特殊功能，同时用较多的篇幅介绍了网络环境下的程序设计和用户界面编程。

本书不仅是一本 Java 语言的入门书，而且兼顾了具有一定 Java 语言基础的程序设计人员的需要，也是一本 Java 语言的提高书。

本书注重实用性，在书中给出了大量实例程序，使读者能够通过例子和上机实践，很快掌握 Java 语言。

本书适合于计算机和通信等相关专业的本科生、研究生、大学教师和工程技术人员使用，特别适合于从事网络和 Internet 开发的科技工作者使用。

JAVA 语言及其程序设计

刘彦明　主编

责任编辑　杨兵　云立实

西安电子科技大学出版社出版发行

空军电讯工程学院印刷厂印刷

陕西省新华书店发行　新华书店经售

开本 787×1092　1/16　印张 31　4/16　字数 741 千字

1997 年 3 月第 1 版　1997 年 3 月第 1 次印刷　印数 1—6 000

ISBN 7-5606-0508-7/TP·0243　　定价：41.00 元

衷心感谢西安电子科技大学 ISN 国家重点实验室的朋友们

衷心感谢西安电子科技大学通信学院计算机通信教研室的朋友们

衷心感谢陕西海泰科技责任有限公司的大力支持

衷心感谢所有关心和支持我的朋友们

前　言

Java 语言是 Sun Microsystem 公司开发的新一代面向对象程序设计语言，特别适合于 Internet 网上的应用软件开发，因此有人也把 Java 语言称为新一代网络编程语言。Java 语言使得网络开发人员、WWW 应用开发人员可以设计具有高度交互能力和安全性的应用程序。

Java 的中文名字叫“爪哇”，实际上是印尼的一个小岛屿的名字，那里盛产咖啡。

Java 作为一种网络编程语言，在 Internet 网上是炙手可热的，大有“听取哇声一片”的气势。Java 作为软件开发的一种革命性技术，其地位已被确立，主要表现在以下几个方面：

(1) 计算机产业的许多大公司购买了 Java 的许可证，这一点说明 Java 已得到工业界的认可。

(2) 众多的软件开发商开始开发支持 Java 的软件产品。今天是以网络为中心的时代，不支持 Java 的应用是不敢想象的。

(3) Java 作为 Internet 和 Intranet 的主力开发语言，为信息产业提供了有力的支持。

因此，无论是做商业软件开发、企业信息系统，还是搞信息服务产业、科学研究，上 Internet，使用 Java 编程将是一个非常有前途的方向。可以这样说：Internet 是一个舞台，Java 语言是一种工具，企业家和科学家将在 Internet 舞台上，用 Java 导演一幕幕惊心动魄的故事。

鉴于此，我们把 Java 语言介绍给大家，以期帮助大家了解并掌握它，为我们的将来服务。

那末，为什么 Java 会如此成功？它有哪些特点呢？

有关 Java 语言的特点已有不少文章作了详细介绍，这里就不再重复了。这里给出几位科学家对 Java 的评价来说明它的成功所在。

Sun 公司副总裁、首席技术官员 Eric Schmidt：

“Java 的出现，正威胁着操作系统开发商。操作系统开发商不可能再通过“API 锁定”来把应用程序束定到特定的操作系统和 CPU 上，形成事实上的垄断，从而控制软硬件的销售。”

“Java 不仅是一种语言，而且是新的网络操作系统。Java 将会成为 90 年代的 DOS，虽然它不完善，但它会像以前的 DOS 一样无处不在。”

MIT 副教授 Danny Hills：

“DNA 代表着 Java 的产生。用 DNA 可以生产出原始微生物和病毒。多细胞生物是由许多互相通信的细胞组成的集合。Internet 已成为巨大的电子生态系统，在这个电子生态系统中生活的‘生物’是用 Java 编写的应用程序。”

Sun 公司研究副总裁 Bill Joy：

“Java 是一种广泛使用的网络编程语言，它是一种新的计算概念。Java 是一种软件虚拟机器，这种虚拟机器是网络加软件。Java 不依赖于机器结构，是面向对象的，可重用性好，

并有并发机制。”

从以上的言论我们可以得出：Java 将成为新的应用程序开发环境的标准。为此，我们把 Java 推荐给大家，希望能起到抛砖引玉的作用。

本书分六篇介绍了 Java 语言：

开篇　进入 Java。主要介绍了 Java 语言的开发环境。通过“ Hello World ”application 和“Hello world”applet 说明了 Java 程序设计的一般过程，包括 Java 程序的组成、编译和执行的整个过程。

第一篇　Java 编程。主要介绍了 Java 语言的编程要素，包括第 2～10 章的内容。

第二篇　编写 applet。主要介绍了 Java 环境下如何编写 applet 应用程序，包括第 11～14 章的内容。

第三篇　创建用户界面。主要介绍了在 Java 环境下如何创建应用程序的界面，包括第 15～18 章的内容。

第四篇　网络应用和安全。主要介绍了在 Java 环境下如何开发网络应用程序和开发符合安全要求的 Java 应用，包括第 19～22 章的内容。

第五篇　将本地方法集成到 Java 程序中。主要介绍了如何在 Java 程序中调用 C 语言函数，即把 C 函数作为 Java 程序中的方法，包括第 23、24 章的内容。

在本书的编写过程中，西安电子科技大学 ISN 重点实验室的硕士研究生高勋、马昕、伦观涛、朱兴全、张凯、付延增等同志做了大量的资料收集和整理工作，在此表示衷心感谢！

由于作者水平有限，书中难免有不妥之处，敬请读者批评指正。

刘彦明

谨识于西安电子科技大学通信学院计算机通信教研室

1996 年 8 月 1 日

目　录

开篇　进入 Java

第一篇　Java 编程

第二篇 编写 applet

第三篇 创建用户界面

第四篇 网络应用和安全

第五篇 将本地方法集成到 Java 程序中

开篇　进入 Java

0.1　Java 开发工具 JDK

Sun 公司提供的 Java 开发工具 JDK 有 Windows 95 版、Windows NT 版和 Solaris 2.x 版，它们都可以从 java.sun.com 处免费获得，国内的 Sun 镜象服务器及各大网络节点的 FTP 服务器上也有。

Windows95 版的 JDK 由下述六种工具组成。

0.1.1　javac——Java 语言编译器

Java 是一种编译语言，与其它语言不同的是：它编译后产生一种字节代码，这种字节代码与机器代码类似，但又不针对于某一具体机器，它对所有变量和方法的引用都使用名字而不是地址。在任何机器上，Java 语言的语义都相同，如 int 总是 32 位，long 总是 64 位，这也是 Java 语言移植性好的主要原因。Java 语言编译器的编译格式为：

javac　　[选项]文件名.java

javac -g　　[选项]文件名.java

javac 将扩展名为.java 的源程序编译成字节代码。传递给 javac 的文件中的每一个定义好的类，都产生各自的字节代码，并单独保存在各自的类名.class 文件中。javac 都把它们放在与.java 文件相同的子目录下。如果在编译的过程中，javac 找不到出现的某个类的定义，则按-classpath 选项提供的路径去查找。

javac -g 是 javac 的一个未优化版本，它适合于像 gdb 或 dbx 之类的调试器调试。

javac 中可使用的选项及含义如下：

-classpath <路径;路径;...>——定义 javac 去查找用户定义的某些类时的路径，各路径间以";"分隔；

-d <目录;目录;...>——定义产生的 class 层次的根目录；

-g——产生调试信息，使字节代码包含行号和局部变量信息，以供调试用。这是缺省方式；

-ng——不产生调试信息，字节代码不能用于调试；

-nowarn——不产生警告信息；

-O——代码优化；

-verbase——使编译和链接程序打印出源文件及字节代码文件的信息。

0.1.2　java——Java 语言解释器

编译后的 Java 字节代码是无法在机器上直接运行的，需要由解释器对其解释执行。Java 语言解释器的解释格式为：

java　　[选项] 类名 <参数>

java -g　[选项] 类名 <参数>

类名是要运行的类的名字，它必须包含所在的组件(package)的名字(例如 java.lang.String)，只有在未定义组件名，即类缺省地放在一个无名的组件内时，才可直接写类的名字。

类名后的参数是传递给类而不是传递给解释器的。

Java 对扩展名为.class 的字节代码解释执行。类名.class 文件中必须包含一个 main()方法，执行即从 main()开始。由于 main()在一个类内定义，但是却不能由这个类的任一实例来调用，因此必须将其定义为静态的。如果在 main()中或其它方法中创建了线程，则只有当 main()及所有线程都结束后，程序才终止运行。否则，main()退出后，程序即终止。

当在执行过程中，用到其他用户自己定义的类时，必须用-classpatch 选项说明它们的路径。当用户未指明时，系统将自动把系统类的路径加到其后部。

一般，我们都使用 javac 将 Java 的源程序编译成字节代码，然后用 java 对其解释执行。然而，若使用-cs 选项，则用 java 即可对源程序编译并解释执行。每个类被加载后，它的修改数据与源文件中的修改数据相比较，如果源文件最近修改过，则需重新编译并加载新的字节代码。

解释器可以通过检查机制确定一个类是否合法，这就使得解释执行的字节代码不会破坏语言的一些约定。

除了上面提到过的-cs 选项及-classpatch 选项外，一般用到的选项还有：

-ms x——设置内存分配池的大小为 x；

-prof——使 Java 解释器产生剖析数据，在当前目录下产生一个“java.prof”的文件；

-v——在每次加载一个类文件后，向标准输出打印信息；

-verbosegc——在每次自动回收无用单元后，向标准输出打印信息；

-vertify——检查所有代码；

-vertifyremote——对通过类加载器加载到系统的代码实行检查，这是缺省方式；

-novertify——关闭检查。

java -g 是 Java 的一个未优化版本，它还有一个专用的选项-t，用以打印跟踪信息。

0.1.3　javah

javah 实现从 Java 的类文件中产生 C 的头文件及 C 剩余文件，其格式为：

javah　　[选项] 类名

javah -g　[选项] 类名

javah 主要用于为实现类中本地的(native)方法而产生 C 的头文件及其它相关文件。其中头文件(.h)包含一个结构的定义，这个结构中的域与类中的实例变量相对应。这个头文件用于 C 程序中对类中变量的引用。

在省缺的情况下，javah 创建一个“CClassHeaders”子目录，并将产生的头文件放在这个子目录下。如果使用了-stubs 选项，javah 会创建一个“ stubs ”子目录并将剩余(stubs)文件放在这个子目录下。

除上面提到的-stubs 选项外，一般用到的选项还有：

-d ＜目录;目录;...＞——覆盖缺省的存放头文件的目录“CClassHeaders”；

-td ＜目录＞——覆盖缺省的存放临时文件的目录“\tmp”；

-verbose——向标准输出打印产生的文件的状态；

-classpath ＜路径;路径;...＞——定义 javah 用以查找类文件的路径，以“;”分隔。

javah -g 是 javah 的一个未优化版本，它适合于像 gdb 或 dbx 之类的调试。

0.1.4　javap——Java 类文件反汇编程序

javap 用以对类文件进行反汇编。它的输出依赖于所使用的选项，若不使用任何选项，javap 仅打印出传递给它的类内的公共域及方法，javap 向标准输出打印信息。Java 类文件反汇编程序的格式为：

javap [选项] 类 . . .

各选项含义如下：

-p——除打印公共的域及方法外，亦打印出私有的和受保护的域及方法；

-c——打印出反汇编码；

-classpath ＜路径;路径;...＞——定义 javap 用以查找类文件的路径，以“;”分隔。

0.1.5　javaprof——Java 剖析工具

javaprof 用于对由 Java 语言解释器产生的剖析数据格式输出。在用 Java 解释器解释执行.class 程序时，若加-prof 选项，则产生一个 java.prof 文件，它包含了 javaprof 用以进行大量统计的信息。javaprof 对其格式化后向标准输出打印，包括三方面的信息：

(1) 每个方法调用的次数和时间。

(2) 每个类调用的次数和时间。

(3) 每种数据类型所占的内存空间。

应注意，在多线程 application 中，剖析数据很不正确，剖析只能用于单线程 application。

Java 剖析工具的格式为：

javaprof [选项] java.prof

选项-v 使 javaprof 打印一些额外的剖析信息。

0.1.6　hotjava——WWW 浏览器

hotjava 允许在 WWW 网上浏览，并且提供通过平台的现场交互功能，交互内容以 applet 的形式存在。applet 是用 Java 语言编写的一类小程序，它只能在支持 Java 的 WWW 浏览器上执行。

通过编写 applet 程序，将其嵌入到 hotjava 的 Web 页中，便可集声音、动画等多种技术于一体，产生生动的页面。在翻阅页面的同时，applet 执行，从而实现交互。hotjava 使网络功能进一步扩大，充分体现了 Java 语言的强大功能。

0.2　"Hello World"application

如果你对编写独立运行的 application(能不依赖于任何浏览器而执行的 Java 程序)感兴趣，那么这里就是你开始的起点。遵循本节中的步骤，就能创建和使用独立运行的 Java application。

• 创建 Java 源程序

创建一个名为 HelloWorldApp. java 的文件，Java 代码是：

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

• 编译源文件

用 Java 编译器编译源文件：

javac HelloWorldApp. java

如果编译成功，编译器则创建名为 HelloWorldApp. class 的文件。如果编译失败，则需确认一下你是否输入并命名了完全如上面所示的程序。

平台特定细节：

在 UNIX 下：

javac HelloWorldApp. java

在 DOS Shell(Windows 95/NT)下：

javac HelloWorldApp. java

• 运行 application HelloWorldApp. java

用 Java 解释器运行程序：

java HelloWorldApp

应该看到"Hello World!"显示在标准输出上。

平台特定细节：

在 UNIX 下：

java HelloWorldApp

在 DOS Shell(Windows 95/NT)下：

java HelloWorldApp

注意：Java 解释器的参数是类名，而不是文件名。

你已经看到了一个 Java application(也许甚至编译并运行了它)，你可能想知道它是怎样工作的以及它与其它 Java application 在多大程度上类似。记住，Java application 是独立运行的 Java 程序(用 Java 语言写的)，是不依赖于任何浏览器运行的程序。

一、定义类

在 Java 语言中，所有方法(函数)和变量都存在于类和对象(类的实例)中。Java 语言不

支持全局变量和杂散变量，因此任何 Java 程序的基干都是类定义。

在上面所示的代码中，class HelloWorldApp 开始了一个类定义块。类，作为面向对象语言(例如 Java)的基本结构单元，是描述与类的实例相关联的数据与行为的模板。当你实例化一个类时，就创建了一个对象，外观和感觉上很像同一个类的其它对象。与类或对象相关联的数据存储在变量中，与类或对象相关联的行为由方法执行。

在编译世界中，一个传统例子是表示矩形的类。这个类包含表示矩形的原点、宽度和高度的变量。一个矩形类的实例包含一个特殊的矩形(如这一页纸)的维数信息。

在 Java 语言中，类定义的最简单形式是：

```
class name {
. . .
}
```

关键字 class 为名为 name 的类开始类定义的标志。这个类的变量和方法用开始和结束类定义块的花括号包含。“Hello World”application 没有变量，只有一个 main()方法。

为了获得更多有关面向对象的信息，请参见第 1 章“面向对象编程的基本概念”。为了学习有关用 Java 语言实现面向对象概念，请参见第 3 章“对象、类和接口”。

二、main()方法

每个 Java application 的首脑是 main()方法。当你用 Java 解释器运行一个 application 时，就确定了你想运行的类名。解释器执行在类中定义的 main()方法。mian()方法控制程序流程，分配所需资源，运行其它任何为 application 提供功能的方法。

每个 Java application 必须包含一个 main()方法，其标识是这样的：

```
public static void main(String args[])
```

main()方法的方法标记包含三个修饰符：

• public 表明 main()方法能被任何对象调用。3.2.5 节“控制对类成员的访问”中讲述了 Java 语言支持的访问限制符 public、private、protected 和隐式 friendly 的来龙去脉。

• static 表示 main()方法是一个类方法。3.2.6 节“实例成员和类成员”讨论了类方法和变量。

• void 表示 main()方法不返回任何值。

1. 调用 main()方法

在 Java 语言中，main()方法与 C 语言和 C++语言中的 main()函数类似。执行 C 或 C++程序时，系统通过首先调用它的 main()函数来启动程序，然后 main() 函数调用程序中的其它函数。类似地，当 Java 解释器执行 application(通过在 application 的控制类上请求)时，通过调用类的 main()方法开始。main()方法能调用所有其它方法。

如果试图在没有 main()方法的类上运行 Java application，解释器则会显示一条与之类似的错误信息并拒绝编译程序。提示信息如下：

```
In class classname: void main(String argv[] ) is not defined
```

其中，classname 是你试图运行的类名。

2. 传递给 main()方法的参数

如下面的代码段所示，main()方法接收一个参数——String 类型数组：

```
public static void main(String args[])
```