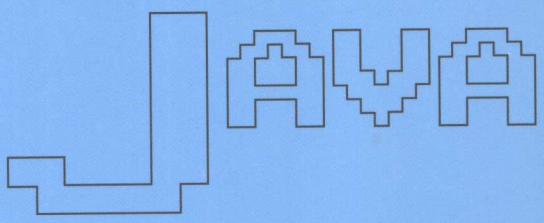
KH北京科海
25年·IT技术出版专家

21世纪高职高专计算机技能与应用系列规划教材

Java程序设计教程与实训

张 哲 主 编
孙 杰 吴 博 副主编



中国人民大学出版社
北京科海电子出版社
www.khp.com.cn

21世纪高职高专计算机技能与应用系列规划教材

Java程序设计教程与实训

张 哲 主 编
孙 杰 吴 博 副主编

中国人民大学出版社
·北京·

北京科海电子出版社
www.khp.com.cn

图书在版编目（CIP）数据

Java 程序设计教程与实训/张哲主编.
北京：中国人民大学出版社，2009
（21 世纪高职高专计算机技能与应用系列规划教材）
ISBN 978-7-300-10497-3
Ⅰ. J…
Ⅱ. 张…
Ⅲ. JAVA 语言—程序设计—高等学校：技术学校—教材
Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2009）第 045820 号

21 世纪高职高专计算机技能与应用系列规划教材
Java 程序设计教程与实训
张　哲　主编

出版发行	中国人民大学出版社　北京科海电子出版社		
社　　址	北京中关村大街 31 号	**邮政编码**	100080
	北京市海淀区上地七街国际创业园 2 号楼 14 层	**邮政编码**	100085
电　　话	（010）82896594　62630320		
网　　址	http: //www.crup.com.cn		
	http: //www.khp.com.cn（科海图书服务网站）		
经　　销	新华书店		
印　　刷	北京市鑫山源印刷有限公司		
规　　格	185 mm×260 mm　16 开本	**版　　次**	2009 年 5 月第 1 版
印　　张	20.75	**印　　次**	2009 年 5 月第 1 次印刷
字　　数	505 000	**定　　价**	32.00 元

内容提要

本书从实用的角度出发，立足于高职高专学生的学习需求，对 Java 语言中的各种语法结构、面向对象程序设计思想进行了详细分析和讲述，对 Java 语言在网络、数据库等方面的应用进行了重点介绍。全书共 12 章，每章均配有适量的习题和相应的实验，有利于读者巩固和检验所学知识。

本书内容编排合理，在重要基础知识的讲解上侧重深度，在实用类案例的讲解上侧重广度，通过本书的学习，读者可以在实践过程中真正掌握 Java 面向对象编程的思想和 Java 在网络编程中的一些重要技术。本书可作为各类高职高专院校、计算机培训学校等相关专业教材，也可作为软件编程人员和 Java 语言自学者的参考用书。

为方便读者学习和参考，书中所有实例的源代码均可到 http://www.khp.com.cn 中下载。

前言

由美国 Sun 公司开发的新一代面向对象的程序设计语言 Java，以其独有的、与网络紧密结合的特点，已经成为 Internet 领域功能最强大、最有前途的编程语言之一。目前，在我国高等院校计算机专业中，都陆续开设了 Java 程序设计课程。为了适应教学的要求，作者针对高等院校、高职高专学生和初学者的特点和要求，结合多年教学实践经验，编写了《Java 程序设计教程与实训》。

本教程在内容结构的组织方面作了精心安排，大部分章节都是以实例为驱动进行讲解，让学生在实际操作中真正掌握所学知识。全书共分为 12 章，前两章介绍了 Java 的基本编程知识，这些内容是学习任何一门编程语言都必须掌握的；第 3 章介绍了 Java 如何实现面向对象编程；第 4 章介绍了 Java 的输入/输出流与文件操作技术；第 5 章介绍了 Applet 编程技术；第 6 章介绍了 Java 异常处理技术；第 7 章介绍了如何在 Java 中实现多线程；第 8 章介绍了如何实现与平台无关的图形用户界面的设计；第 9 章介绍了 Java 语言的网络编程技术；第 10 章介绍了如何使用 JDBC 编程技术访问数据库；第 11 章介绍了 JavaScript 编程技术；第 12 章安排了与前面章节相对应的实验，可强化读者的实际应用能力。书中所有程序均调试运行通过。

本教程紧扣 Java 编程语言的特点，从平台独立性、面向对象、多线程和网络编程等方面逐一展开，从不同的方面展现了 Java 语言的精髓。由于本书面向高等院校、高职高专学生和 Java 语言的初学者，所以特别注重技术应用，在讲述文法上尽量做到简洁明了、通俗易懂。结合 NCRE（全国计算机等级考试），每章最后都有适量的习题，有利于读者复习、巩固所学知识，同时还配有相应实验内容，以培养读者的实际动手和应用能力。

为方便读者学习和参考，书中所有实例的源代码均可到 http://www.khp.com.cn 中下载。

由于时间仓促，加之编者水平有限，不足之处在所难免，敬请读者批评指正。

编者

2009 年 3 月

目　录

第1章

Java 语言概述

本章知识点概述

通过本章的学习，了解 Java 语言的发展、特点以及 Java 和 C++的不同之处，掌握 Java 应用程序的一般结构以及 Applet 小程序的生成方法。通过技能训练，掌握 Java 运行环境的基本使用方法。

本章所含知识点

- Java 语言的特点
- Java 和 C++的比较
- 面向对象编程的基本概念
- Java 编程原则
- Java 应用程序和 Applet 小程序
- Java 程序的运行环境

知识重点

- Java 编程原则
- JCreator 工具的使用
- Java 应用程序的一般结构以及 Applet 小程序的生成方法

知识难点

- 面向对象编程的基本概念
- Java 程序的运行环境

1.1 Java 语言简介

1.1.1 Java 语言的发展

Java 语言的前身称为 Oak 语言，它是由美国 Sun 公司于 1991 年开发的一个称为“Green”的软件项目，该项目的本意是开发一个用于消费类电子产品的、与平台无关的软件技术。从那以后，Oak 语言一直被认为是用来开发消费类电子产品和交互式电视控制器的工具。1994 年，Sun 公司的两个开发人员在 Oak 语言的基础上创建了 HotJava 的第一个版本，当时称为 Webrunner，即在 Web 上使用的图形浏览器，经过一段时间后才称为 Java。1995 年 5 月，Sun 公司对外正式发布了 Java 语言，随后立即得到了各 WWW 厂商的大力支持，纷纷在浏览器上加入 Applet 小程序（用 Java 语言编写的小应用程序），并通过 Internet 在世界各地进行传播。

自从 1995 年 Sun 公司正式发布 Java 1.0 以来，在全球范围内引发了经久不衰的 Java 热潮，Java 的版本也不断更新到 v1.1、v1.2、v1.3、v1.4、v1.5，最新的 Java 2 SDK 版本是 1.6，其内容也有了巨大的改进和扩充。随着 Java 技术的不断发展，Java 还分化出三个不同的版本以满足不同的需求：针对企业网应用的 J2EE（Java 2 Enterprise Edition）、针对普通 PC 应用的 J2SE（Java 2 Standard Edition）和针对嵌入式设备及消费类电器的 J2ME（Java 2 Micro Edition）。另外迅速发展的还有 JavaBean、其他的 Java 编译器和集成开发环境等第三方软件。

实际上，Java 语言是与 Internet 同步发展起来的一种新型网络语言，是近 20 年来计算机软件环境中的最有意义的进步之一。Java 语言在网络中的地位同超文本标记语言 HTML（Hypertext Markup Language）一样重要。

Java 语言是一种强有力的网络编程语言，它最大限度地利用了网络资源。其 Applet 小程序可以跨平台、跨操作系统、跨网络运行。Applet 代码小，易于在网络上快速地下载和发送，且具有不需要修改应用程序就可以增加 Web 页的新功能。Java 还为编程技术人员提供了许多公用的系统接口。随着 Internet 在全世界范围内的广泛流行，以及在各个领域的渗透，Java 语言已经被各行各业的人士所接受。

1.1.2 Java 语言的特点

Java 语言之所以被人们广泛认可，是因为它具有许多先进的技术特点。

1. 移植性好

Internet 上运行着各种各样的计算机系统，这些系统上的计算机软、硬件千差万别。要使一种应用软件在网络上任何一种计算机操作系统中都能正常地运行，就必须保证这种应用软件的移植性要好。也就是说，应用软件本身不受计算机硬件和操作系统的限制（即与平台无关），软件代码可以在不同的计算机环境中正常地运行。Java 语言就是一种与平台无关、移植性好的编程语言，在源程序级就保证了其基本数据类型与平台无关。Java 源程序经编译后产生的二进制代码是一种与具体指令无关的指令集合，通过 Java 虚拟机（Java Virtual Machine，JVM），可以在不同的平台上运行。

2. 纯面向对象技术

Java是一种完全面向对象的程序设计语言（Object Oriented Programming，OOP）。Java语言代码以类的形式组成。Java删除了C++中非面向对象的特性，使得应用程序的开发变得十分容易。Java面向对象技术的特征主要有封装性、继承性和多态性。

3. 分布式

Java语言中提供了一个支持HTTP和FTP等基于TCP/IP协议的类库，通过这些类库，Java应用程序可以通过URL打开并访问网络上的对象。因此，用Java语言可以很方便地编写适合Internet和分布式环境下的应用程序。

4. 安全性

分布式计算环境要求软件具有高度的稳定性和安全性。Java语言具有强大的安全结构和安全策略。Java之所以具有高质量的安全性，是因为采取了以下一些措施。

（1）取消了指针操作，不允许直接对内存进行操作。

（2）实现内存管理自动化，内存布局由Java决定。

（3）提供了字节码检验器，以保证程序代码在编译和运行过程中接受一层层的安全检查，这样可以防止非法程序或病毒的入侵。

（4）提供了文件访问控制机制，严格控制程序代码的访问权限。

（5）提供了多种网络软件协议的用户接口，用户可以在网络传输中使用多种加密技术来保证网络传输的安全性和完整性。

5. 编译和解释的结合性

Java是一种编译和解释相结合的语言。一个Java语言源程序要运行，必须先由Java编译器编译成字节码（ByteCode），即文件扩展名为.class。这个字节码文件不是最终的执行程序，不能在具体平台上运行，而必须再由运行系统上的字节解释器将其翻译成机器语言，达到边翻译边执行程序的目的。由于这个解释过程是在编译后的字节码上进行的，所以运行速度仍然相当快。

1.1.3 Java和C++的比较

C++语言适应了软件工程界的面向对象的新潮流，但是C++并不能真正满足面向对象的程序，它保留了许多非面向对象的特点。而Java语言实现了纯面向对象的特点，使得它较C++语言更为简单，更为人们所接受。

从Java语言的起源与发展来看，一方面，它由C++发展而来，其语言风格与C++十分相似。另一方面，Java又比C++简单，它删除了C++中难理解、易引起安全隐患的内容。Java与C++的区别主要表现在以下几个方面。

1. 指针

指针和内存地址是C++最有效也是最有害的特性。不正确的指针操作会引起许多错误。

在 Java 中不允许使用指针。

2. 内存分配

C++中的内存分配与指针操作有同样的危险，其内存分配是通过 malloc()和 free()库方法以及 new 和 delete 两个运行符来实现的，程序员需要自己释放空间。Java 语言中没有 malloc()和 free()方法。由于每个复杂的数据结构都是对象，它们通过 new 运算符在内存上分配空间，一旦不再访问对象，占据的内存空间就会自动被收回，根本不需要人为干预。

3. 全局变量

C++全局变量作为程序的状态信号没有很好地进行封装。Java 语言中只有类是全局的，不可能创建一个不属于任何类的全局变量。

4. 数据类型

不同的 C++编译器根据不同机器的实际配置，分配给数据类型以不同的字长。Java 语言所给定的基本数据类型确定了一个合理的字长并保持不变，Java 语言解释器的这种严格与硬件无关的数据类型很难对代码进行优化和实现，Java 语言的数据类型是很脆弱的，但这是唯一能够保证跨平台实现的途径。

5. goto 语句

在 C++引入异常处理之前，goto 语句经常被用来在异常处理中跳出循环。Java 语言没有 goto 语句，它严格定义的异常处理机制使 goto 语句没有再存在的必要，取消这种随意跳转的语句有利于优化代码及保持系统的稳定性和安全性。

6. 头文件

C++有由 include 引用的头文件，Java 语言没有头文件，但 Java 语言有由 Java 引入的程序包。

7. 多重继承

C++具有多重继承的特性，Java 语言不直接支持多重继承。

1.1.4 面向对象编程的几个基本概念

面向对象编程技术是当今软件开发中常用的技术之一。Java 语言是一种新型的纯面向对象的程序设计语言，它使用一种称为“类（class）”的软件对象，其代码可重用和可扩展，可以将这些由变量和方法组成的类作为一个模板，在这些模板上增加其他功能又可以创建用户自己需要的新类，无须重写许多代码，因此，应用程序的开发变得容易。面向对象编程常常涉及的几个主要概念如下。

1. 对象

对象（object）可以是现实生活中的任何物体。例如，书、人或动植物、计算机等都是现

实生活中的对象，一些看不见的事务、规则也是客观存在的对象。任何一个物体都包括两个基本方面，一个是物体的内部构成（例如，组成飞机的各个部件），另一个是物体的行为，即对该物体内部构成成分的操作或与外界信息的交换（例如，飞机的发动、起飞等）。与物体的两个基本面相对应，OOP中的对象也有两个基本的成分：数据和方法。数据表示对象的构成，方法表示对象的行为。

2. 类

类（class）是一组具有相同基本成分的对象集合。例如，大学由许多具体的学校组成，大学就是一个类，而具体的北京大学就是大学类的对象。

类是一种最基本、最重要的数据类型。作为一种数据类型，在程序中必须定义相应类型的数据，类所定义的数据被称为这个类的实例（instance）或对象。类有两种基本成分：数据和方法，它们被封装在类的体内，与外界分隔开。数据是类的成员，它可以是基本类型的数据（例如，int、char 等），也可以是一个类的实例。方法也是类的成员，它用于处理该类的数据，类的方法与其他语言中的函数很相似，具有方法的名称、参数、方法体和返回值等。

3. 封装

封装（encapsulation）是把类（或对象）的基本成分（数据和方法）封装在类体（或对象体）内，使之与外界分隔开来。

一般地，类中具有隐藏复杂性的机制。类中的每个方法或变量都可以被定义为公有或私有。类公有部分可以让外界的用户知道或必须知道，公有的实例变量和方法是一个类与外部的接口，应用程序通过这个接口使用类的功能。定义私有的方法和实例数据则不能被类外部的其他代码所访问。

在进行 OOP 编程时，首先，需要定义类，并指出类的名称、数据成员和方法，确定类成员被访问的权限等。其次，利用类创建相应的实例（或对象）。最后，在程序运行时，由系统根据需要与对象交换信息。

4. 继承

继承（inherit）是指在已有类的基础上建立一个新类。新类自动拥有父类的所有方法和实例变量，然后再根据需要，添加新任务所需的方法或实例变量。

一个不由其他类派生来的类称为基类，一个派生类的最近的父亲（其最近的上层类）为该类的父亲，从某一类派生出来的类称做该类的子类。在 Java 语言中，一个父类可以有多个子类，但一个子类只能从一个父类继承而来，即指一个子类不能有一个以上的父类，这就是说 Java 语言不支持多重继承。为了解决多重继承的问题，Java 语言中引入了接口的概念，并克服了多重继承的复杂性。

合理使用继承可以减少很多的重复劳动。如果在一个类中实现了一个特别的功能，那么在其派生类中就可以重复使用这些功能，而不需要重新编程来实现。可以用 Java 语言的内置类创建派生类，也可以对用户自己创建的类建立派生类。

继承和封装具有很好的合作性。如果一个给定的类封装了某些属性，那么任何子类将继承这些属性。

5. 多态性

多态性（polymorphism）是指在一个类中可以有多个名字相同，但参数的类型和数量不同的实现方法。调用这些方法时，其方法的参数类型和数量决定了是调用哪一个方法，这样不仅简化了方法的实现和调用，也便于记忆。

方法的覆盖（或重写）是指在继承的过程中，子类可以重新定义父类的某些方法，实现自己需要的功能。

在 Java 语言中，通过方法重载（overloading）和覆盖（overriding）实现多态性。方法重载是指多个方法具有相同的名称，但各个方法的参数不同。调用方法时，系统根据实际参数自动选择相应的方法。

1.1.5 Java 语言的用途

Java 语言由于具有与平台无关的特点及完备的面向对象特性，因此越来越受到开发人员的喜爱和认可，也决定了 Java 语言的特殊用途。Java 语言应用的领域包括很多，但它的侧重面与其他语言不同，希望读者通过本书的学习后，掌握与下面主要应用有关的知识。

1. Java 语言可以方便地增加 Internet/Intranet 网页的功能

用 Java 语言编制的 Applet 小程序执行结果的表现形式是基于 Web 页面的网页的，它可以方便地嵌入到 HTML 代码中，并且很容易被调用并实现其功能。目前，人们对网页十分熟悉，可以利用一些流行工具来制作网页，利用 Java Applet 小程序可以扩充网页的功能，丰富网页内容。

2. Java 语言可以方便地实现多种任务的并行工作

许多语言都具备实现多任务并行工作的功能，但 Java 语言以其网络系统应用研究的优势，再加上 Java 语言提供的多线程编程技术，使得利用 Java 语言编制基于网络环境的多任务程序十分容易。例如，完成在网页中同时对多个任务进行监视等操作任务。

3. Java 语言可以方便地访问网络资源

利用 Java 语言可以很容易地编制出访问网页资源的程序。例如，要访问某网页的信息，包括域名、IP 地址、信息终止日期、信息修改日期、信息的长度等内容；要获取某网络服务器上某个文件的内容；要设计网络客户端与服务器端的应用程序（如邮信服务、对话服务等）等。Java 语言呈现出很好的网络语言的特征。

4. Java 语言可以方便地实现基于 Web 的数据库访问

Java 语言提供了方便的数据库访问的技术。利用 Java 语言中的 JDBC 技术，用户能很方便地开发出基于 Web 网页的数据库访问程序，从而扩充网络应用的功能。

5. Java 语言可以方便地实现分布式应用

Java 语言提供了许多技术以方便实现分布式应用。利用 Java 语言中的 JSP、RMI、CORBA

和EJB等技术、用户可以方便地实现分布式应用。

1.2 Java程序简介

Java源程序是由类定义组成的，因此在源程序中只能以类定义的形式来进行编程。一个Java源程序可以包含一个类定义，也可以包含多个类定义。Java源程序以.class为后缀名的字节码输出到文件中。在开始学习Java语言编程之前，应该先了解Java源程序的基本构成，以及如何编写、编译和运行Java程序，以便建立一个总体的印象。

一般地，Java程序分为Java应用程序和基于Java的Applet小程序两种。

1.2.1 一个简单的Java应用程序

用Java语言编写应用程序，再到编译源程序，最后得到结果，需要经过3个过程，即编写源程序、编译和运行。

1. 编写源程序

一个简单的Java应用程序如【边学边练1.1】所示。

【边学边练1.1】 一个简单的Java应用程序，文件名为HelloChina.java。

```
public class HelloChina
{
    public static void main(String[] args)
    {
        //输出一字符串
        System.out.println("您好,中国!");
    }
}
```

通过这个程序，可以看到Java应用程序还是比较简单的，结构并不复杂。之所以称其为应用程序，其标志就是含有main()方法，只有含有main()方法的Java程序才能应用Java命令来运行。编写Java程序时必须遵循以下几条Java语言的编程原则。

（1）Java程序是无格式的纯文本文件，可以用任何文本编辑器（例如，Word、写字板）编写Java程序。

（2）文件名必须与类名一致，且两者的大小写要一致。例如，类名为HolloChina，那么存盘时的文件名应该为HelloChina.java。

（3）首先要使用关键字class来定义一个新类，类名紧跟其后。这里public表明是一个公共类，可以省略，HelloChina是【边学边练1.1】所定义的类名。

（4）一个Java程序可以有多个类，每个类可以有多个方法，但是最多只有一个公共类。

（5）对于一个应用程序来讲，还必须有一个main()方法，且只能有一个main()方法。该方法标志着执行应用程序时的起始点。其中，关键字public表明所有的类都可以调用该方法，关键字static表明该方法是一个静态方法，关键字void表示main()方法无返回值。

（6）任何方法中可以有多条语句。【边学边练 1.1】的 main()方法中只有一条语句：

```
System.out.println("您好,中国!");
```

该语句用来在屏幕上输出一个“您好，中国!”字符串，System.out.println()的功能与 C 语言中的 printf()函数相同。

（7）Java 程序中的每条语句都要以分号“;”结束（包括以后程序中出现的类型说明等）。

（8）为了增加程序的可读性，程序中可以加入一些注释行，例如，用“//”开头的行。关于 Java 语言的注释定义符说明详见第 2 章。

【边学边练 1.1】仅仅是一个最简单的 Java 程序，只包含了一个类，类中包含了一个方法，并没有包含数据，如果类中要包含数据，结构中还需要有类说明等语句。

2. 编译

当 Java 程序编写完成后，必须经过 Java 编译器把.java 源程序翻译成以.class 为后缀的、与处理器无关的二进制字节码文件。在 Java 语言环境中，执行下面的命令编译 HelloChina.java 程序，编译后将自动得到 HelloChina.class 字节码文件。

```
javac HelloChina.java
```

3. 运行

以.class 为后缀的字节码文件，并不是一个可执行程序，它必须通过字节码检验器和 Java 解释器翻译后才能运行，通过特殊的运行环境，可形成.exe 的可执行文件。在 Java 语言环境中，执行下面的命令运行 HelloChina.class 程序：

```
java HelloChina
```

如果一切正常，运行结果将如图 1.1 所示。

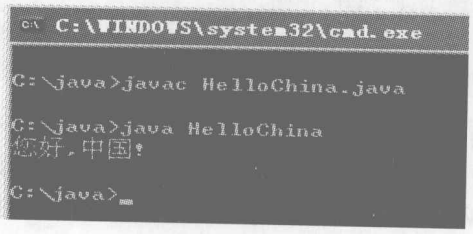


图 1.1　HelloChina 程序运行结果

1.2.2　将应用程序编写为 Applet 小程序

基于 Java 语言的 Applet 小程序是可以嵌入到 HTML 语言中，并由带有 Java 解释器的 WWW 浏览器（如 Internet Explorer、Netscape Communicator 等）来解释执行的程序，它不可以用 Java 解释器直接执行。将 Applet 称为小程序的原因就是因为其代码较小，易于通过 Internet/Intranet 下载。

一般地，用 Applet 小程序实现一个功能，得到最后的结果，需要经过编写 Applet 小程序、编译小程序、编写 HTML 文件、执行 HTML 代码程序 4 个过程。执行 HTML 程序有两种方

式，一种是通过运行浏览器执行 HTML 程序，另一种是通过 Java 软件包提供的 appletviewer 命令执行 HTML 程序。

1. 编写 Applet 小程序

【边学边练 1.2】 Applet 小程序示例，文件名为 HelloWorld.java。

```
import java.applet.Applet;
import java.awt.Graphics;
public class HelloWorld extends Applet
{
   public void init()
   {
      resize(400,300);
   }
   public void paint(Graphics g)
   {
      g.drawString("您好,世界!",60,40);
   }
}
```

从【边学边练 1.2】中，可以看到基于 Java 的 Applet 小程序其基本结构与 Java 应用程序相似，不同地方说明如下。

（1）小程序开始处，必须引入两个系统类：java.applet.Applet 和 java.awt.Graphics，程序中需要使用这两个类中的方法。引入系统提供的类时，应该用关键词 import，它与 C 语言中的#include 功能一样。

（2）关键词 extends 表示 HelloWorld 类是由 Applet 类继承而来的。

（3）init()方法和 paint(Graphics g)方法分别是 Applet 类和 Graphics 类中定义的公有方法，分别用于小程序的初始化和显示输出。

（4）所有定义的 Applet 小程序都必须是从 Applet 类中派生出来的，且不能有 main()方法。关于 Applet 小程序详细介绍请看第 5 章。

2. 编译 Applet 小程序

Applet 小程序 HelloWorld.java 编写完成后，需要对其进行编译，自动产生字节码文件 HelloWorld.class，其编译方法与 Java 应用程序相同。

3. 编写 HTML 文件

对于 Applet 小程序的字节码程序 HelloWorld.class 必须嵌入到 HTML 代码中，才可以完成小程序的功能。所以，还必须为 HelloWorld.class 编写一个 HTML 的代码文件，将字节码程序引入其中。下面是一个 HTML 代码文件，文件名为 HelloWorld.html。关于 HTML 语言，不是本书的主要内容，有兴趣的读者请参阅相关书籍。

【边学边练 1.3】 嵌入 HelloWorld.class 代码的 HTML 程序示例，文件名为 HelloWorld.html。