免费提供电子教案

高等院校规划教材
计算机科学与技术系列

Java 程序设计实用教程

主　编　董洋溢

机械工业出版社
CHINA MACHINE PRESS

高等院校规划教材·计算机科学与技术系列

Java 程序设计实用教程

主　编　董洋溢
副主编　周军妮　崔　岩
参　编　辛向丽　薛　萍　王萍利

机械工业出版社

本书从 Java 语言的发展历程、面向对象程序设计的基本概念入手，由浅入深地介绍了 Java 语言的编程方法。全书共分为 10 章，涉及 Java 语言概述、数据类型和基本语句、类和对象、异常处理、Java 常用 API、图形用户界面程序设计、Applet 小程序和多媒体、Java 多线程编程、Java 网络编程及 Java 综合项目实例。内容基本涵盖了 J2SE（Java2 Standard Edition，Java2 标准版）的基本内容，为进一步学习 Java 高级开发技术奠定了基础。

本书内容翔实，实例丰富，在前 9 章的最后均给出了配套习题和上机实验；在第 10 章给出了 4 个实用的 Java 综合项目开发实例，使读者能够尽快熟悉 Java 应用程序的开发过程。

本书适合作为高等学校计算机类相关专业的教材，也可以作为初学者及 Java 开发技术人员的参考书。

本书配套授课电子课件，需要的教师可登录 www.cmpedu.com 免费注册，审核通过后下载，或联系编辑索取（QQ：2399929378，电话：010-88379753）。

图书在版编目（CIP）数据

Java 程序设计实用教程 / 董洋溢主编. —北京：机械工业出版社，2014.4
高等院校规划教材 • 计算机科学与技术系列
ISBN 978-7-111-45939-2

Ⅰ. ①J… Ⅱ. ①董… Ⅲ. ①JAVA 语言－程序设计－高等学校－教材
Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2014）第 031843 号

机械工业出版社（北京市百万庄大街 22 号　邮政编码 100037）
责任编辑：郝建伟　师沫迪
责任印制：李　洋
三河市宏达印刷有限公司印刷
2014 年 4 月第 1 版 • 第 1 次印刷
184mm×260mm • 18.75 印张 • 465 千字
0001－3000 册
标准书号：ISBN 978-7-111-45939-2
定价：39.90 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务	网络服务
社 服 务 中 心：（010）88361066	教 材 网：http://www.cmpedu.com
销 售 一 部：（010）68326294	机工官网：http://www.cmpbook.com
销 售 二 部：（010）88379649	机工官博：http://weibo.com/cmp1952
读者购书热线：（010）88379203	封面无防伪标均为盗版

出版说明

计算机技术在科学研究、生产制造、文化传媒、社交网络等领域的广泛应用，极大地促进了现代科学技术的发展，加速了社会发展的进程，同时带动了社会对计算机专业应用人才的需求持续升温。高等院校为顺应这一需求变化，纷纷加大了对计算机专业应用型人才的培养力度，并深入开展了教学改革研究。

为了进一步满足高等院校计算机教学的需求，机械工业出版社聘请多所高校的计算机专家、教师及教务部门针对计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了教材的体系架构与编写原则，策划开发了“高等院校规划教材”。

本套教材具有以下特点：

1）涵盖面广，包括计算机教育的多个学科领域。

2）融合高校先进教学理念，包含计算机领域的核心理论与最新应用技术。

3）符合高等院校计算机及相关专业人才培养目标及课程体系的设置，注重理论与实践相结合。

4）实现教材“立体化”建设，为主干课程配备电子教案、教材和实验实训项目等内容，并及时吸纳新兴课程和特色课程教材。

5）可作为高等院校计算机及相关专业的教材，也可作为从事信息类工作人员的参考书。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢广大读者的支持与帮助！

机械工业出版社

前　言

Java 程序设计语言自从 1995 年诞生之日起就受到了广泛的关注，并掀起了全球范围内的学习热潮。时至今日，作为第一个完全的面向对象的程序设计语言，Java 语言的风靡热度并没有消退，反而随着新应用的不断出现越来越流行。这主要归功于 Java 语言不断完善的功能以及自身特点的优势，比如良好的可扩展性、与平台无关性等。因此，Java 语言也就自然而然地成为了程序员开发的首选，占领了软件开发领域的半壁江山。

近年来，国内外越来越多的高等学校计算机类及其相关专业都开设了 Java 程序设计这门课程，也陆续出版了很多教材。但有些教材理论性过强，实例较少，不易上手，往往“吓退”初学者。也有些教材过多地讨论实例，理论知识过于零散，初学者并不适用。正是基于这个考虑及多年一线教学实践的经验，笔者编写了此书。希望用适当的篇幅介绍理论，同时辅以恰当的实例来进行实验，帮助初学者快速、准确地掌握面向对象程序设计的相关概念和方法。在每章的最后都编写了和该章内容相关的习题和上机实验，帮助初学者在掌握理论知识的基础上，举一反三地进行练习，从而更好地学会运用知识解决实际问题。同时在最后一章，本书还特别编写了 4 个实用的 Java 应用程序，启发初学者如何综合运用 Java 语言来设计应用程序。从这个意义上讲，本书是一本非常实用的理论和实践相结合的 Java 教材，特别适合初学者以及注重应用性的开发者参考使用。

本书从 Java 语言的发展历程着手，由浅入深地介绍了 Java 语言的基本概念及编程方法。内容涵盖了 Java 标准版（J2SE）的基本内容，不涉及数据库编程（JDBC）及 J2EE。全书共分为 10 章，涉及 Java 语言概述、数据类型和基本语句、类和对象、异常处理、Java 常用 API、图形用户界面程序设计、Applet 小程序和多媒体、Java 多线程编程、Java 网络编程及 Java 综合项目实例。

本书的第 1、2 章由辛向丽老师编写，第 3、10 章由董洋溢老师编写，第 4、7 章由周军妮老师编写，第 5 章由薛萍老师编写，第 6 章由王萍利老师编写，第 8、9 章由崔岩老师编写，石硕、李立勇两位同学对部分代码进行了调试，全书由董洋溢老师统稿。本书在编写的过程中，也得到了机械工业出版社郝建伟编辑的全力支持和帮助，在此一并表示感谢！

由于作者水平有限，书中难免存在错误和疏漏，恳请广大读者及专家们批评指正！

编　者

目　录

第1章 Java语言概述

Java 程序设计语言是一种完全的面向对象的高级程序设计语言，自诞生之日起就受到了广泛的关注。其系统类库不断丰富，性能不断提高，应用领域也不断扩展，目前已成为最流行的软件开发语言之一。

本章主要介绍面向对象的基本概念、Java 语言的发展概况、Java 程序的分类、Java 程序的基本结构、Java 程序的开发环境等内容。

1.1 程序设计语言概况

1.1.1 程序设计语言的发展历程

计算机程序设计语言的发展，经历了从机器语言、汇编语言到高级语言的历程。

1. 机器语言

机器语言，是第一代计算机语言。计算机所使用的语言是由“0”和“1”组成的二进制数，二进制是计算机语言的基础。在计算机发明之初，人们使用二进制数所组成的数字代码来操控计算机，简单地说，就是写出一串串由“0”和“1”组成的指令序列交由计算机执行，这种语言，就是机器语言。

机器语言的运行效率是计算机语言中最快的。但是对于程序员来说使用机器语言是十分痛苦的，因为所谓的语言都是由一些没有意义的“0”和“1”组成的。由于要表达复杂的逻辑，这些代码的位数还非常多。记忆、理解和表达的难度都很大，特别是在程序需要修改时，更是如此。所以机器语言对于程序员来说，学习难度大，难掌握，不容易使用。此外，由于每台计算机的指令系统往往各不相同，在一台计算机上执行的程序，要想在另一台计算机上执行，必须另编程序，造成了重复工作。因此虽然机器语言的运行效率最快，但是它的编写效率是很低的。

2. 汇编语言

为了提高计算机语言的编写效率，降低计算机语言的使用难度，人们进行了改进。用一些简洁的英文字母、符号串来替代特定的指令的二进制串，比如，用“A D D”代表加法，“M O V”代表数据传递，等等。这样一来，人们很容易读懂并理解程序，程序纠错及维护都变得方便。这种程序设计语言就称为汇编语言，即第二代计算机语言。

汇编语言同样依赖于机器硬件，移植性不好，但汇编语言编写效率高。针对计算机特定硬件而编制的汇编语言程序，能准确发挥计算机硬件功能。虽然使用有意义的字母和符号代替了数字，使得该语言便于学习，但是它的语法仍然是按照机器语言的表达方式来进行表达，在语法逻辑上与人的表达习惯有很大不同，仍然不好理解。由于汇编语言编写的

程序保持了执行的高效，所以至今在一些直接面向底层硬件控制的应用中仍在使用汇编语言进行编程。

3. 高级语言

随着科技的进步，计算机科学家们渐渐意识到应该设计一种这样的语言：这种语言接近于数学语言或人的自然语言，同时又不依赖于计算机硬件，编出的程序能在所有机器上通用。经过不断的努力，1954 年，第一个完全脱离机器硬件的高级语言——Fortran 问世了，60 多年来，共有几百种高级语言出现。影响较大、使用较普遍的有 Fortran、ALGOL、COBOL、BASIC、LISP、SNOBOL、PL/1、Pascal、C、Prolog、Ada、C++、VC、VB、Delphi、Java 等。

高级语言的发展经历了从早期语言到结构化程序设计语言，从面向过程到非过程化程序语言的演变过程。20 世纪 60 年代中后期，软件越来越多，规模越来越大，而软件的生产基本上是程序员各自为战，缺乏科学规范的系统规划与测试评估标准。其带来的后果是大批耗费巨资建立起来的软件系统由于含有错误而无法使用，甚至带来巨大损失。软件给人的感觉是越来越不可靠，以致人们认为几乎没有不出错的软件。这一切，极大地震动了计算机界，史称“软件危机”。人们认识到：大型程序的编制不同于编写小程序，它应该是一项新的技术。应该像处理工程一样处理软件研制的全过程。程序的设计应易于保证正确性，也便于验证正确性。1969 年，提出了结构化程序设计方法，1970 年，第一个结构化程序设计语言——Pascal 语言出现，标志着结构化程序设计时期的开始。

20 世纪 80 年代初开始，在软件设计思想上，又产生了一次革命，其成果就是面向对象的程序设计的诞生。在此之前的高级语言，几乎都是面向过程的，程序的执行是流水线似的，在一个模块被执行完成前，人们不能干别的事，也无法动态地改变程序的执行方向。这和人们日常处理事物的方式是不一致的。对人而言是希望发生一件事就处理一件事，也就是说，不能面向过程，而应是面向具体的应用功能，也就是对象（Object）。其方法就是软件的集成化，如同硬件的集成电路一样。生产一些通用的、封装紧密的功能模块，称之为软件集成块，它与具体应用无关，但能相互组合，完成具体的应用功能，同时又能重复使用。对使用者来说，只关心它的接口（输入量、输出量）及能实现的功能，至于如何实现的，那是它内部的事，使用者完全不用关心，C++、Java 就是这方面的典型代表。

1.1.2 面向对象的相关概念

面向对象程序设计（Object-Oriented Programming，OOP），是一种新兴的程序设计方法，或者是一种新的程序设计规范（Paradigm），其基本思想是使用对象、类、继承、封装、消息等基本概念来进行程序设计。从现实世界中客观存在的事物（即对象）出发来构造软件系统，并且在系统构造中尽可能运用人类的自然思维方式。

1. OOP 基本概念

（1）对象（Object）

对象是系统中用来描述客观事物的一个实体，它是构成系统的一个基本单位。一个对象由一组属性和对这组属性进行操作的一组服务组成。从更抽象的角度来说，对象是问题域或实现域中某些事物的一个抽象，它反映该事物在系统中需要保存的信息和行为；它是一组属性和有权对这些属性进行操作的一组服务的封装。客观世界是由对象和对象之间的

联系组成的。

（2）类（Class）

类是具有相同属性和服务的一组对象的集合，它为属于该类的所有对象提供了统一的抽象描述，其内部包括属性和服务两个主要部分。在面向对象的编程语言中，类是一个独立的程序单位，类有一个类名并包括属性说明和服务说明两个主要部分。类与对象的关系就如“模具”和“铸件”的关系，类的实例化结果就是对象，而对同一类对象的抽象就是类。例如：“学生”和“张三”。“学生”是类，“张三”是对象。

（3）消息（Message）

消息就是向对象发出的服务请求，它应该包含下述信息：提供服务的对象标识、服务标识、输入信息和回答信息。服务通常被称为方法或函数。

2. OOP 基本特征

（1）封装性（Encapsulation）

封装性包含两层含义，一是指把对象的全部属性和全部服务结合在一起，形成一个不可分割的独立单位（即对象）；二是指信息隐藏，即尽可能隐蔽对象的内部细节，对外形成一个边界（或者说形成一道屏障），只保留有限的对外接口使之与外部发生联系。

封装原则要求外部对象不能随意存取对象的内部数据（属性），从而有效地避免了外部错误对对象的“交叉感染”，使软件错误局部化，从而减少查错和排错难度。

（2）多态性（Polymorphism）

对象的多态性是指在一般类中定义的属性或服务被特殊类继承之后，可以具有不同的数据类型或表现出不同的行为。这使得同一个属性或服务在一般类及其各个特殊类中具有不同的语义。简单来说，所谓多态就是指相同的信息给予不同的对象会引发不同的动作。

（3）继承（Inheritance）

如果一个类 A“继承自”另一个类 B，就把这个 A 称为“B 的子类”，而把 B 称为“A 的父类”。继承可以使得子类具有父类的各种属性和方法，而不需要再次编写相同的代码。在令子类继承父类的同时，可以重新定义某些属性，并重写某些方法，即覆盖父类的原有属性和方法，使其获得与父类不同的功能。

1.2 Java 语言简介

1.2.1 Java 语言的发展史

1. Java 语言发展简介

Java 语言是由 Sun Microsystems 公司（现已并入 Oracle 公司）的 James Gosling 和同事们在一个名为“Green 计划”的项目中开发设计的。Java 最初被命名为 Oak，目标设定为家用电器等小型系统的程序开发语言，Java 所开发的程序主要应用在电视机、电话、闹钟、烤面包机等家用电器的控制和通信中。由于这些智能化家电的市场需求没有预期的高，Sun 公司放弃了该项计划。随着 20 世纪 90 年代互联网的发展，Sun 公司看到了 Oak 在互联网上应用的前景，于是改造了 Oak，于 1995 年 5 月以 Java 的名称正式发布。Java 伴随着互联网的迅猛发展而发展，逐渐成为重要的网络编程语言。

2．Java 版本历史

Java 语言及 JDK（Java Development Kit，Java 软件开发工具包）的各种版本发展：

- 1995 年 5 月 23 日，Java 语言诞生。
- 1996 年 1 月，第一个 JDK——JDK1.0 诞生。
- 1997 年 2 月 18 日，JDK1.1 发布。
- 1998 年 12 月 8 日，Java2 企业平台 J2EE 发布。
- 1999 年 6 月，SUN 公司发布 Java 的三个版本：标准版（J2SE）、企业版（J2EE）和微型版（J2ME）。
- 2000 年 5 月 8 日，JDK1.3 发布。
- 2000 年 5 月 29 日，JDK1.4 发布。
- 2001 年 9 月 24 日，J2EE1.3 发布。
- 2002 年 2 月 26 日，J2SE1.4 发布，自此 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。
- 2005 年 6 月，JavaOne 大会召开，SUN 公司发布 Java SE 6。此时，Java 的各种版本已经更名，各种版本取消了其中的数字“2”：J2EE 更名为 Java EE，J2SE 更名为 Java SE，J2ME 更名为 Java ME。
- 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，Oracle 公司发布 Java SE 7。

3．Java 平台分类

Java 平台共分为三个主要版本 Java SE、Java EE 和 Java ME。

Java SE：标准版的 Java 平台是一个 Java2 的平台，为用户提供一个程序开发环境。这个程序开发环境提供了开发与运行 Java 软件的编译器等开发工具、软件库及 Java 虚拟机。它也是 Java2 平台、企业版本和 Java 网页服务的基础。

Java EE：Java 平台企业版（Java Platform Enterprise Edition），是 Sun 公司为企业级应用推出的标准平台。

Java ME：以往称作 J2ME（Java Platform, Micro Edition）是为机顶盒、移动电话和 PDA 之类嵌入式消费电子设备提供的 Java 语言平台，包括虚拟机和一系列标准化的 Java API。它和 Java SE、Java EE 一起构成 Java 技术的三大版本，并且同样是通过 JCP（Java Community Process）制订的。在目前，Java ME 最流行的应用是游戏软件。与其他嵌入式语言和开发平台（如索尼等）相比，Java ME 程序不需要昂贵的专用设备和开发工具，而是可以直接在 PC 机上开发和仿真运行，然后很容易地部署到目标机器或设备上，从而使其开发、测试和发布的整个过程都变得容易和廉价。

1.2.2 Java 语言的特点

在 1.1.1 节中已经介绍过，尽管所有高级语言的语法结构和表达逻辑更接近人类的表达方式。但无论什么样的语言，对于计算机来讲，只懂得以二进制为基础的机器语言。那么，如何让计算机可以“懂得”高级语言呢？这就需要一个翻译机制把程序员所写的高级语言翻

译成计算机可以识别的机器语言。按照不同的翻译方式，翻译语言大致可以分成两种类型：编译型语言和解释型语言。

编译型语言在执行源程序时，首先通过专门的编译器将源文件编译成为目标文件也就是机器语言，然后通过对目标文件中机器语言的执行来完成程序的任务。该程序在第一次执行后生成的目标文件将长期独立存在。从此之后每一次执行该程序时，系统将直接执行目标文件而不需要再去编译源代码了。因此编译型语言除了第一次执行时需要的时间较长，之后的执行速度都比较快。

解释型语言没有目标文件，在需要执行源程序时，本地计算机中的解释器会把代码读取一行，然后解释成机器语言进行执行，之后再读取一行源代码，再重复上面的步骤，直到读到源代码的末尾。由于没有目标文件的存在，所以每次执行相同的源程序时都是读一行执行一行，因此相对编译型语言，执行的效率较低。

这两种类型的高级语言，各有特色和优劣。编译型语言，执行效率较快，系统资源占用相对较小。但由于其编译的目标文件是针对第一次编译时所处的计算机软硬件平台生成的机器指令代码，因此目标文件一般只能在相同的计算机平台上运行，在不同的计算机平台上往往得不到相同的执行结果，也就是跨平台性、可移植性不强。另外修改编译型语言的源代码后必须重新编译后修改效果才可以体现出来，否则目标文件永远是未修改前的机器指令。典型的编译型语言有：C、C++、Pascal、Delphi 等。解释型语言，执行效率较之编译型要慢，但是由于每次执行都是由本地解释器完成读取、解释和执行，因此其跨平台性、可移植性要强于编译型语言。典型的解释型语言有：Perl、BASIC 等。

Java 语言是一种比较特殊的语言类型，可以称作混合型语言或者半编译半解释型语言。在执行 Java 源程序时，首先 Java 程序也会对源代码进行编译，但是并不是编译成内容为机器指令代码的目标文件，而是编译成称作字节码的文件，然后在 Java 虚拟机（Java VM）上用解释方式执行字节码。Java 虚拟机的解释过程其实就是把字节码文件的代码转换成本地系统的机器指令代码来进行执行。通过以上过程的描述，读者应该发现，Java 语言结合了编译型和解释型语言的一些优势，既提高了执行的效率，又做到了良好的跨平台性和可移植性。真正实现一次编程，到处执行的理想状态！

除此之外，Sun 公司对 Java 语言还有这样的解释："Java 编程语言是个简单、面向对象、分布式、解释性、健壮、安全与系统无关、可移植、高性能、多线程和动态的语言"。Java 编程语言的风格十分接近 C++语言，继承了 C++语言面向对象技术的核心，与此同时 Java 舍弃了 C++语言中容易引起错误的指针（改以引用取代）。同时，Java 移除了 C++中运算符重载以及多重继承（改用接口取代），增加了垃圾回收器功能。在 Java SE 1.5 版本中引入了泛型编程、类型安全的枚举、变长参数和自动装/拆箱特性。

1.2.3 Java 程序的分类

Java 程序按运行方式可以分为两种类型：Java 应用程序（Application）和 Java 小程序（Applet）。这两种程序的开发流程是类似的，但是在运行环境和发布上却有着显著的不同。

1. Java 应用程序

Java 应用程序（Application）是独立完整的程序，它可以在 Java 平台上独立运行，通常在命令行使用独立的解释器（Java 命令）即可运行。另外，Java 应用程序的主类一般必

须有一个主方法 main()，该方法的定义格式如下：

```
public static void main(String args[]) { … //方法体}
```

main()方法是 Java 应用程序的标志，和 C/C++语言类似，main()方法也是 Java 应用程序执行的入口点。

【例 1-1】 最简单的 Java 应用程序 HelloWorld.java。

```
1 //HelloWorld.java
2 public class HelloWorld    //类名
3 {
4     public static void main(String args[])        //主方法 main()
5     {
6         System.out.println("Hello, World!");      //屏幕输出字符串“Hello,World!”
7     }
8 }
```

程序的运行结果如图 1-1 所示。

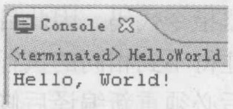

图 1-1 【例 1-1】的运行结果

在【例 1-1】中，程序的第 2 行定义一个类 HelloWorld，第 3 行为类开始的大括号{，第 8 行为类结束的大括号}，大括号必须成对匹配。第 4 行到第 7 行定义了一个 main 函数，和 C 语言一样，是 Java 应用程序的入口点。在 Java 中 main 函数也称 main 方法，其函数原型是固定的格式。第 6 行 System.out.println("Hello, World!");语句向屏幕输出字符串"Hello, World!"后换行。

2. Java 小程序 Appet

Java Applet 是一种嵌入在 HTML 网页文档中的 Java 程序，不能独立运行，也没有 main()方法，必须通过网页浏览器来运行，因此称为小程序。我们知道，WWW 浏览器是 Internet 上遵循 WWW 协议的软件。HTML 语言是 WWW 浏览器中的通用语言。具体的方法是把 Java 的小程序编译生成的字节码文件（.class 文件）嵌入在 HTML 文件中，当这些文件传送到 WWW 浏览器中时，由浏览器中内置的 Java 解释器来解释执行该 Java 小程序。由此可见，当运行一个 Java 小程序时，也要为它写一个 HTML 文件，然后在 WWW 浏览器中观看这个 HTML 文件，就可以激活浏览器中的 Java 解释器，来运行 Java 小程序。

Java 小程序同 Java 应用程序之间在编写组成、计算结构和运行方式上都有着较大的差别。【表 1-1】列出了小程序与应用程序之间的主要差别。

表 1-1 小程序与应用程序的差别

功 能 要 求	应 用 程 序	小 程 序
使用图形	可选	固定用图形
装载	主要从文件系统装入	通过 HTML 连接
内存要求	最低 Java 应用程序要求	Java 程序加 Web 浏览器要求
环境输入	命令行参数	嵌入 HTML 文档的参数
执行过程	主函数(main)启动过程	init 初始化过程 start 启动过程 stop 暂停/关闭过程 destroy 终止过程 paint 绘图过程

【例 1-2】 最简单的 Java Applet 小程序 HelloApplet.java。

```
//HelloApplet.java
import java .awt.Graphics;
import java.applet.Applet;
public class HelloApplet   extends Applet          //继承自父类 Applet
{
    public void paint(Graphics g)                  //小程序的 paint()方法
    {
        g.drawString("Hello, Applet!", 20,20);     //屏幕输出字符串“Hello,World!”
    }
}
```

【例 1-2】中的小程序不能独立运行，必须嵌入到 HTML 文件中，可以将对应的 HelloApplet.html 文件编写如下：

```
//HelloApplet.html
<html>
  <head>
    <title>Applet 对应的 HTML</title>
  </head>
  <body>
    <applet code="HelloApplet.class" width=200   height=100>
    </applet>
  </body>
</html>
```

程序的运行结果如图 1-2 所示。

在 HelloApplet.html 文件中，<applet code="HelloApplet.class" width=200 height=100>是进行加载 HelloApplet 小程序类的标记语句，程序执行到此句时就会加载已经编译好的小程序结果，并将要现实的内容显示在 HTML 页面上。</applet>标记为<applet>对应的结束标记。

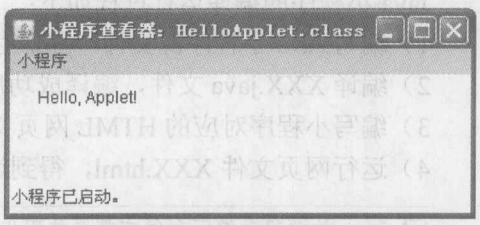

图 1-2 【例 1-2】的运行结果

1.2.4 Java 程序的基本结构

通过 1.2.3 节的两个例子，可以看到简单的 Java 程序的结构。

一个完整 Java 源程序结构通常包括下面的语句：

```
package 语句;              //该语句至多一句
import 语句;               //该部分可以有若干 import 语句或者没有，必须放在所有的类定义之前
public classDefinition;    //公共类定义部分，至多只有一个公共类的定义
                           //Java 语言规定该 Java 源程序的文件名必须与该公共类名一致
classDefinition;           //类定义部分，可以有 0 个或者多个类定义
interfaceDefinition;       //接口定义部分，可以有 0 个或者多个接口定义
```

其中，

- package 语句：Java 编译器为每个类生成一字节码文件，且文件名与类名相同，同名的类有可能发生冲突。为了解决这一问题，Java 提供包来管理类名空间。在 Java 的系统类库中，把功能相似的类放到一个包（package）中，例如，图形界面的类放在 java.awt 这个包中，与网络功能有关的类放到 java.net 这个包中。
- import 语句：引入包，包中定义了要使用的系统类库或程序员自定义过的类。
- 类（class）的定义：一个 Java 源程序中除了开头的 package 语句和 import 语句之外，接下来一般就是一个或者多个类的定义。一个源文件中只能包含一个 public 类型的类，源文件名必须和 public 型的类名同名，非 public 型的类可以定义多个。类中通常包含若干个变量和函数（方法）的定义。
- 接口（interface）的定义：一个 Java 源程序中除了可以定义多个类之外，还可以定义一个或多个接口，接口是一种特殊的类。

观察一下【例 1-1】和【例 1-2】中的 Java 源程序文件 HelloWorld.java 和 HelloApplet.java，文件中均包含 package、import 及类的定义等部分。

1.2.5 Java 程序的编译运行过程

1. Java 应用程序的编译运行过程

Java 应用程序的编译运行过程如下：

1）编写 Java 应用程序源文件，并保存为 XXX.java 的格式。

2）编译 XXX.java 文件，编译成功后生成二进制的字节码文件 XXX.class。

3）运行字节码文件，得到运行结果。

2. Java 小程序的编译运行过程

Java 小程序的编译运行过程如下：

1）编写 Java 小程序源文件，并保存为 XXX.java 的格式。

2）编译 XXX.java 文件，编译成功后生成二进制的字节码文件 XXX.class。

3）编写小程序对应的 HTML 网页文件（嵌入 XXX.class）并保存为 XXX.html。

4）运行网页文件 XXX.html，得到运行结果。

> 📖 Java 的源程序编写和保存需要使用相关工具来完成，Java 源程序进行编译和执行也要借助于相关开发工具来完成。

1.3 Java 程序的开发环境

1.3.1 JDK+EditPlus

1. JDK

（1）JDK 简介

Java Development Kit（简称 JDK）是 Sun 公司针对 Java 开发人员发布的免费软件开发工具包。其版本号也在不断更新中，从 JDK1.5 开始增加了泛型类库等功能。JDK1.6 是目前