



普通高等院校“十一五”规划教材

Java 程序设计 实例教程

主编 刘丽华 付晓东



国防工业出版社

National Defense Industry Press

普通高等院校“十一五”规划教材

Java 程序设计实例教程

主编 刘丽华 付晓东

国防工业出版社

· 北京 ·

内 容 简 介

Java 语言是由美国 Sun 公司开发的一种功能强大的多线程动态计算机编程语言,具有简单、面向对象、分布式、可移植等特点。Java 语言非常适合于企业网络和 Internet 环境,现已成为 Internet 中最受欢迎、最有影响的编程语言之一。本书使用 Sun 公司发布的 Java2 SDK 1.4.2 作为开发工具,对 Java 语言进行介绍。

全书共分为 15 章,采用实例带动知识点学习的方法进行讲解,通过学习实例掌握软件的操作方法和操作技巧,以及程序设计方法和设计技巧。本书按节细化了知识点,并结合知识点介绍了相关的实例。

本书可作为大学非计算机专业的教材,也可作为普通高等学校及高职高专院校计算机专业的教材,还可作为初学者的自学用书。

图书在版编目(CIP)数据

Java 程序设计实例教程/刘丽华,付晓东主编. —北京:国防工业出版社,2010.11

ISBN 978-7-118-07090-3

I. ①J... II. ①刘... ②付... III. ①JAVA 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 201088 号

※

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

涿中印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 19 1/4 字数 443 千字

2010 年 11 月第 1 版第 1 次印刷 印数 1—4000 册 定价 36.00 元

(本书如有印装错误,我社负责调换)

国防书店: (010)68428422

发行邮购: (010)68414474

发行传真: (010)68411535

发行业务: (010)68472764

前 言

Java 语言是由美国 Sun 公司开发的一种具有面向对象、分布式和可移植等性能的功能强大的多线程动态计算机编程语言。同时,Java 语言还是一种跨平台的程序设计语言,可以在各种类型的计算机和操作系统上运行。Java 语言非常适合于企业网络和 Internet 环境,现在已成为 Internet 中最受欢迎、最有影响的编程语言之一。另外,它还因其可移植、安全性能高和网络移动性等逐渐成为一种行业标准。Java 语言以其独有的开放性、跨平台性和面向网络的交互性风靡全球,是目前最常用的计算机编程语言,也是主要的网络开发语言之一。

本书使用 Java2 SDK 1.4.2 作为开发工具介绍 Java 语言。本书的特点是内容广泛且通俗易懂,以实例为主,全面介绍了用 Java 语言编程所需的各方面知识。全书以 Java 程序设计实例操作为主线,以实例带动知识点的学习,通过大量实用、经典的编程实例介绍 Java 语言,使读者可以快速掌握、应用所学的 Java 语言编程知识。

本书内容由浅及深、由易到难、循序渐进、图文并茂,理论与实际制作相结合,读者在阅读学习时不但能够快速入门,而且可以达到较高的水平。教师在使用本教材进行教学时,可以一边指导学生练习各章的实例,一边讲解各实例知识中的知识和概念,将它们有机地结合在一起,以达到事半功倍的效果。本书的作者由大学的计算机教师和计算机公司的程序设计员组成,他们有丰富的教学和实践经验,总结出了一套任务驱动式的教学方法。采用这种方法学习的学生将更容易掌握 Java 语言的编程方法和编程技巧。本教材由刘丽华、付晓东担任主编。参加编写工作的人员有高艳、吴艳和余冀辉,全书由刘丽华统稿。

采用实例形式编写本书,是新的尝试,我们诚恳希望读者在使用过程中提出宝贵的意见和建议,以便于我们在以后的工作中加以改进。由于作者水平有限,加上时间仓促,书中难免有不妥之处,恳请广大读者批评指正。

编 者

2010 年 8 月

目 录

第 1 章 Java 概述	1
1.1 Java 语言简介	1
1.2 Java 运行系统	2
1.3 Java 运行环境	4
1.3.1 JDK 的下载与安装	4
1.3.2 JDK 的基本操作命令	4
1.3.3 Java 开发工具	5
1.4 两类 Java 程序	5
1.4.1 Java Application 程序	5
1.4.2 Java Applet 程序	6
1.4.3 Java 程序的基本规则	7
1.5 实践解析	8
习题	9
第 2 章 Java 程序设计基础	10
2.1 数据类型概述	10
2.2 Java 语法基础	11
2.3 运算符与表达式	15
2.3.1 算术运算符与算术表达式	15
2.3.2 关系运算符与关系表达式	17
2.3.3 逻辑运算符与逻辑表达式	17
2.3.4 位运算符	18
2.3.5 赋值运算符	19
2.3.6 条件运算符	19
2.3.7 运算符优先级	20
2.4 实践解析	21
习题	24
第 3 章 Java 程序设计控制结构	28
3.1 顺序结构	28
3.2 选择结构	28
3.2.1 单分支选择结构(图 3-1)	28
3.2.2 双分支选择结构(图 3-2)	30
3.2.3 多分支选择结构	31

3.3 循环结构	34
3.3.1 while 语句(图3-3)	34
3.3.2 do—while 语句(图3-4)	36
3.3.3 for 语句(图3-5)	37
3.3.4 循环跳转语句	39
3.4 实践解析	41
习题	43
第4章 方法	45
4.1 方法的概念及作用	45
4.2 方法的定义与调用	46
4.2.1 方法的定义	46
4.2.2 方法的调用	48
4.3 实践解析	53
习题	55
第5章 数组与字符串	57
5.1 数组	57
5.1.1 什么是数组	57
5.1.2 一维数组	57
5.1.3 二维数组	59
5.1.4 数组应用举例	60
5.1.5 数组参数与相关系统方法	62
5.2 字符串	65
5.2.1 字符串常量与String类	66
5.2.2 字符串变量与StringBuffer类	72
5.3 实践解析	74
习题	78
第6章 面向对象程序设计	81
6.1 面向对象的概念	81
6.2 类的概念	82
6.3 类的定义	82
6.4 类的成员变量	83
6.5 类的方法	85
6.6 实践解析	91
习题	94
第7章 类的封装	100
7.1 封装	100
7.2 继承	101
7.3 多态	105
7.4 实践解析	109

习题	111
第8章 接口	113
8.1 接口	113
8.2 抽象类	116
8.3 包	117
8.4 内部类	120
8.5 Java 基本类库	123
8.6 实践解析	130
习题	132
第9章 异常处理	134
9.1 Java 异常处理的基本概念	134
9.2 异常类的产生、捕获和处理	135
9.3 抛出异常	137
9.4 自定义异常类	140
9.5 实践解析	141
习题	144
第10章 Java 图形界面的设计	145
10.1 GUI 概述	145
10.2 AWT 概述	145
10.3 容器和版面配置	146
10.4 事件处理	156
10.5 AWT 组件	163
10.6 AWT 的高级组件	177
10.7 AWT 的基本绘图	186
10.8 实践解析	195
习题	200
第11章 多线程	202
11.1 线程的基本概念	202
11.2 线程的基本结构与使用方法	203
11.3 线程的管理	207
11.4 实践解析	209
习题	214
第12章 输入、输出流	215
12.1 Read 类(java.io.Reader)	215
12.2 InputStream 类(java.io.InputStream)	216
12.3 File 类(java.io.File)	217
12.4 FileReader 类	219
12.5 BufferedReader 类	219
12.6 Writer 类(java.io.Writer)	222

12.7	OutputStream 类(java.io.OutputStream)	223
12.8	FileWriter 类	223
12.9	RandomAccessFil 类(java.io.RandomAccessFile)	225
12.10	管道流类	227
12.11	实践解析	230
	习题	234
第 13 章	Applet	236
13.1	Applet 简介	236
13.1.1	Applet 及其工作原理	236
13.1.2	第一个 Applet 小程序	236
13.1.3	Applet 的安全性	238
13.2	Applet 类	238
13.2.1	java.applet 包与 Applet 的继承关系	238
13.2.2	Applet 的生命周期	239
13.2.3	Applet 与 Application 的区别	241
13.2.4	与 Applet 程序有关的 HTML 文件标记	242
13.3	Applet 的应用	243
13.4	实践解析	258
	习题	263
第 14 章	JDBC 编程	264
14.1	JDBC 概述	264
14.2	JDBC 的工作原理	267
14.2.1	JDBC 的结构	267
14.2.2	常用的 JDBC 类与方法	268
14.3	实践解析	272
	习题	281
第 15 章	网络编程	282
15.1	TCP/IP 概述	282
15.2	统一资源定位器(URL)	283
15.3	Java 和 URL	284
15.4	UDP 基础	287
15.5	基于 TCP/IP 协议 Socket 通信	291
15.6	实践解析	296
	习题	298

第 1 章 Java 概述

1.1 Java 语言简介

一、Java 语言的发展

Java 语言诞生于 20 世纪 90 年代,其前身是 Sun 公司开发的一种智能化家电语言 Oak。

1995 年正式推出了 Java 语言,随后立即得到了各 WWW 厂商的大力支持,从此 Java 走上了快速发展的轨道。

1996 年, Sun 公司正式推出 Java 开发工具包 JDK 1.0(Java Development Kit)。

1997 年, Sun 公司发布了 JDK 1.1。

1998 年, Sun 公司发布了 Java 2 平台及 JDK 1.2。Java 2 平台是 Java 技术发展新的里程碑,标志着 Java 技术发展的新阶段。

目前, Sun 公司重新定义了 Java 技术的架构,将 Java 2 平台分为三个版本,即 Java 2 平台标准版(J2SE)、Java 2 平台企业版(J2EE)、Java 2 平台微缩版(J2ME)。

二、Java 语言的特征

1. 简单性

由于 Java 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,如最典型的指针操作等,降低了学习的难度;同时 Java 还有一个特点就是它的基本语法部分与 C 语言几乎一模一样,容易让人们接受。

2. 面向对象

Java 是一种完全面向对象的程序设计语言,它去除了 C++ 语言中为了兼容 C 语言而保留的非面向对象的内容,在编程过程中全部使用类和对象。同时,Java 语言支持静态和动态风格代码的继承和重用,甚至一些基本数据类型,如整型、字符型等在 Java 语言中都是可以作为对象处理。

3. 分布式

Java 支持分布式计算的特征。分布式计算中的“分布”指的是数据分布和操作分布。数据分布即应用系统所操作的数据可以分散存储在不同的网络节点上;操作分布即应用系统的计算可由不同的网络节点完成。Java 能实现两种层次上的分布。

4. 安全性

由于 Java 是应用于网络的开发语言,因而安全性变得至关重要。Java 在语言机制和运行环境中都引入了多级安全措施,其主要措施如下:

(1) 内存分配及布局由 Java 运行系统规定。

① 取消了指针的操作,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 JVM 决定。
- (2) 运行系统提供代码认证、字节码验证与代码访问权限控制的安全控制模型。
 - ① 提供字节码检验器, 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。
 - ② 提供了文件访问控制机制, 严格控制程序代码的访问权限。
 - ③ 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 应用程序由编译器编译成字节码, 这个字节码不是最终可执行的程序, 不能在具体的平台上运行, 还需要由运行系统上的字节码解释器将其解释成机器语言, 从而达到边解释边执行的目的。

6. 可移植性

Java 的最大特点“一次编程, 处处使用”。任何机器只要配备 Java 虚拟机, 便可以运行 Java。主要原因是 Java 语言不是针对某个具体平台结构设计的, Java 源程序经编译后产生的字节码是一种与具体指令无关的指令集合, 通过 Java 虚拟机就可以在任何的平台上运行, 这样可以使开发人员大大降低了开发、维护 and 管理的开销。

7. 多线程

多线程技术是允许应用程序并行执行, 即同时做多件事, 满足了一些复杂软件的要求。Java 不但内置多线程功能, 而且提供语言级的多线程支持, 即定义了一些用于建立、管理多线程的类和方法, 使得开发具有多线程功能的程序变得简单、容易和有效。

1.2 Java 运行系统

一、Java 运行系统的构成

1. Java 运行系统

Java 运行系统包括五部分: 类装配器、字节码检验器、解释器、代码生成器、运行支持库, 如图 1-1 所示。

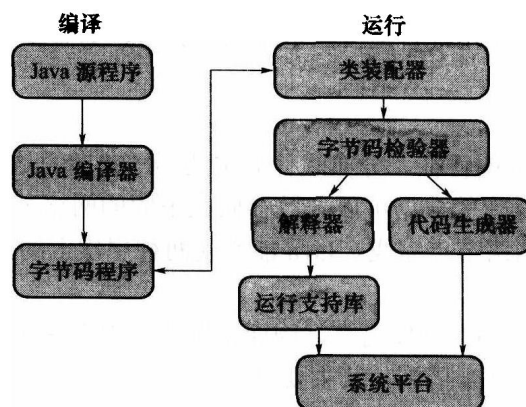


图 1-1 Java 运行系统组成

2. Java 运行系统对字节码的解释分为三个部分

(1) 装入代码。Java 程序的基本单位是类，类装配器是装入运行时所需的 Java 程序代码即程序所需的所有的类，包括代码中的类所继承的类和被其所调用的类，当装入运行程序所需的类后，运行系统便可以确定整个可执行程序的内存在布局。

(2) 验证代码。Java 代码通过字节码检验器对代码进行检验，防止 Java 运行环境运行有害的类文件，并且还能发现是否有操作数栈溢出、非法数据类型转换等逻辑性错误。

(3) 执行代码。

对字节码的执行可以有以下两种方式：

① 即时编译方式：由代码生成器将字节码解释成机器码，然后再执行本机代码。

② 解释执行方式：解释器每次把一小段代码解释成机器码并执行，如此反复完成 Java 字节码的解释操作。这种方法有足够的灵活性，通常采用这种方法。

二、Java 虚拟机(JVM)

Java 的运行之所以与平台无关，其原因是 Java 虚拟机屏蔽了与具体平台相关的联系。一般高级语言的编译器编译生成的代码是为某一个特定硬件平台运行而产生的机器码，但 Java 语言的编译器是将代码编译成 Java 虚拟机的目标代码即字节码而不是针对特定平台的机器代码，所以用户只需在所使用的平台上安装 Java 虚拟机(JVM)，Java 虚拟机中的 Java 解释器负责将字节码文件解释成特定的机器码进行运行。

JVM 其实是在实际的计算机上通过软件模拟来实现的虚拟机器。Java 虚拟机有自己想像的硬件，共包括以下 5 个抽象的逻辑组件。

(1) 指令集：用来执行基本的 CPU 运算。

(2) 寄存器：用来存放地址信息，比如执行指令的地址。

(3) 栈结构：栈用来保存操作参数、返回结果和为方法传递参数。

(4) 堆：Java 运行时的数据区，程序初始化时所有对象都置于堆上。

(5) 存储区：用来存放字节码的方法代码、符号表。

JVM 对这些组件进行了严格的规定，尤其对字节码格式做了更加严密的规定，但它并没有对这些组件具体的实现技术加以规定，也就是它们可以用任何一种技术实现 JVM 的功能，用软件或是芯片均可。但是 Java 虚拟机的功能必须是统一的，只能执行 JVM 规范的统一的字节码。

三、Java API

Java API 也是一个很大的 Java 类库集合，它给程序员提供了大量的可重用的类，如同 C 语言中的库函数，给编程人员提供了丰富的功能。这些类以包的形式组织，所谓包，可以理解为一个文件夹，它把类库中各种功能相近或者相关的类组织在一起，下面介绍最常用的 6 个包。

(1) java.lang: Java 语言的核心类库，包含基本数据类型和出错处理方法。

(2) java.io: Java 语言的标准输入/输出类库，包含 Java 程序与操作系统、用户界面等进行交互时使用的类。

(3) java.awt: Java 语言用来构建图形图像类库，包含创建用户界面和绘制图形图像的所有类。

(4) java.applet: 实现运行 Internet 浏览器中的 Java Applet 的工具类库，包含创建 Applet

以及实现 Applet 所需的所有类。

(5) java.util: Java 语言的一个集合包, 如 Map、Set、日期与时间相关的类等。

(6) java.sql: 支持通过 JDBC 对数据库进行访问操作。

编写 Java 程序时, 如能熟练自如地运用类库, 将在很大程度上提高编程的效率和质量。

1.3 Java 运行环境

1.3.1 JDK 的下载与安装

1. 下载

Sun 公司为所有的 Java 程序员提供了一套免费的 Java 开发和运行环境, 取名为 Java2 JDK, 可以从 <http://java.sun.com> 网站下载。安装的时候可以选择安装到任意的硬盘驱动器上, 如安装到 C:\j2sdk1.4.2_04 目录下。

2. 安装

对于 Windows 98 系统, 需要修改系统根目录下的 autoexec.bat, 使用文本编辑器打开 autoexec.bat, 在该文件最后增加如下命令:

```
set path=%path%; C:\j2sdk1.4.2_04\bin
```

```
set classpath=.;C:\j2sdk1.4.2_04\lib
```

对于 Windows 2000/NT/XP 系统设置如下:

进入“控制面板”→“系统”→“高级”→“环境变量”→“系统变量”。双击 path, 在变量值上输入“C:\j2sdk1.4.2_04\bin”。如图 1-2 所示。

再新建 classpath 变量, 变量值为“.;C:\j2sdk1.4.2_04\lib”。如图 1-3 所示。

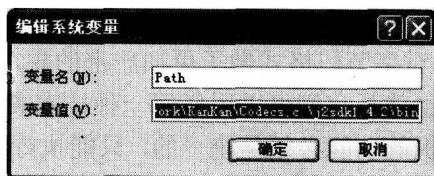


图 1-2 系统变量设置

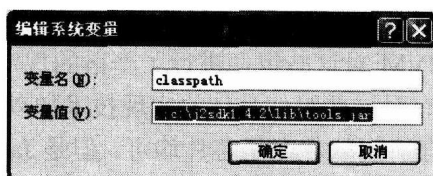


图 1-3 classpath 变量

1.3.2 JDK 的基本操作命令

JDK 的操作命令分为 6 类: 基本命令、RMI 命令、国际化命令、安全控制命令、Java IDL 和 RMI-IIOP 命令、Java Plug-in 命令。下面主要介绍一些常用的基本命令。

基本命令包括 javac、java、javadoc、appletviewer 等。

(1) javac: Java 语言的编译器即 Java 源代码编译成字节码。

基本语法: javac [选项] 源文件名

(2) java: Java 语言的解释器, 解释运行 Java 字节码。

基本语法: java [选项] 类名

(3) javadoc: Java API 生成器。解释 Java 源程序中类的定义和文档注释, 并生成相应

的 HTML 格式的文档。

(4) appletviewer: Java applet 浏览器。

基本语法: applet viewer [-debug] urls

1.3.3 Java 开发工具

JDK 不是集成的开发环境,它采用命令行的方式对程序进行编译和运行。所以除使用 JDK 外,还需要程序编辑软件。可以采用以下两种方式:

(1) 文本编辑器。如 Edit、Notepad 等作为 Java 的编辑软件。

在编辑器中将程序编辑好并保存后,用命令行方式对 Java 程序进行编译、运行。

(2) 用能够与 JDK 配合使用的具有能开发与调试 Java 程序的编辑软件。如 JCreator。

JCreator 提供了图形化的编辑环境,可以在该环境中直接编写程序,程序编译后的错误提示或运行结果将在该环境最底下的窗口显示(图 1-4)。所以程序员可以在 JCreator 中直接进行调试或查看结果。

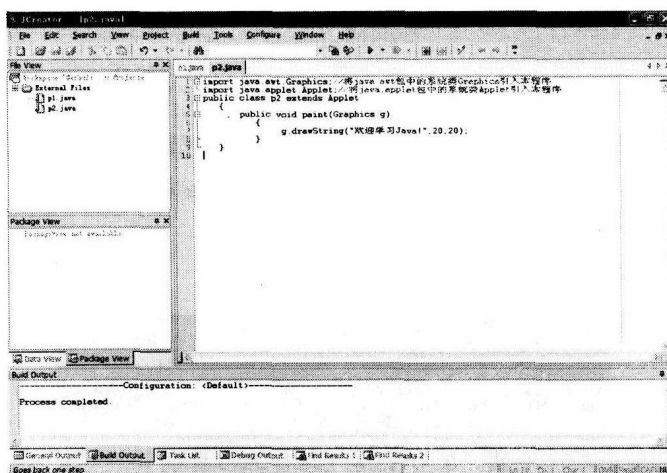


图 1-4 JCreator 界面

1.4 两类 Java 程序

Java 程序分为 Java 应用程序(Java Application)和 Java 小应用程序(Java Applet)两类。对于 Application,它是一种独立的程序,通过解释器就可以即时运行,不依赖浏览器;对于 Applet,它不是独立的程序,必须嵌入 WWW 页面中,在支持 Java 的 WWW 浏览器上才能执行。

1.4.1 Java Application 程序

Java Application 程序由编程开始,到最后看到结果,需要经过 3 个过程,即编写源代码、编译和运行。

1. 编写源程序

【例 1-1】一个简单的 Java 程序。

源程序

```
public class appfirst
{
    public static void main(String args[])
    {
        System.out.println("欢迎使用 java");
    }
}
```

2. 编译

编译的作用就是经过编辑器将 Java 源程序编译成 Java 字节码文件。首先进入 Windows 系统的 DOS 环境中,再进入 D:\source 路径后,键入如下命令:

```
javac appfirst.java
```

回车后,如果程序没有错误,屏幕上则不显示错误信息并将源文件 appfirst.java 生成了 appfirst.class。

3. 运行

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 解释器解释后才能运行。在上述编译操作后,执行 Java 命令运行 Java 程序。命令行如下:

```
java appfirst
```

编译过程演示如图 1-5 所示。

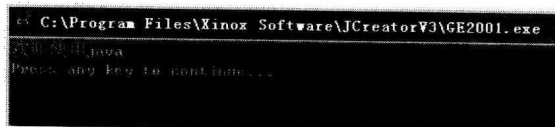


图 1-5 Java 命令运行 Java 程序

1.4.2 Java Applet 程序

Java Applet 程序实现一个功能需要经过如下 4 个过程:编写 Applet 程序、编译 Applet 程序、编写 HTML 文件、运行带 Applet 程序的 HTML 文件。

1. 编写 Applet 程序

【例 1-2】一个简单的 Applet 程序。

源程序

```
import java.awt.Graphics;    //将 java.awt 包中的系统类 Graphics 引入本程序
import java.applet.Applet;  //将 java.applet 包中的系统类 Applet 引入本程序
public class firstapplet extends Applet
{
    public void paint(Graphics g)
    {
        g.drawString("欢迎学习 Java!",20,20);
    }
}
```

```
}  
}
```

2. 编译 Applet 程序

3. 编写 HTML 文件

【例 1-3】一个嵌入 Applet 程序的 HTML 文档。

 源程序

```
<html>  
  <head>  
    <title>A simple Applet</title>  
  </head>  
  <body>  
    <applet code="firstapplet.class" width="200" height="70" >/applet>  
  </body>  
</html>
```

4. 运行

运行带 Applet 程序的 HTML 有两种方式：

(1) 在浏览器中执行运行 HTML。

完成 AppletIn.html 保存后，可以使用 IE 浏览器解释执行 Applet 程序。

(2) 使用 appletviewer 运行 HTML 文件。

在命令行上使用 appletviewer 运行 AppletIn.html 的命令(图 1-6)：

appletviewer AppletIn.html

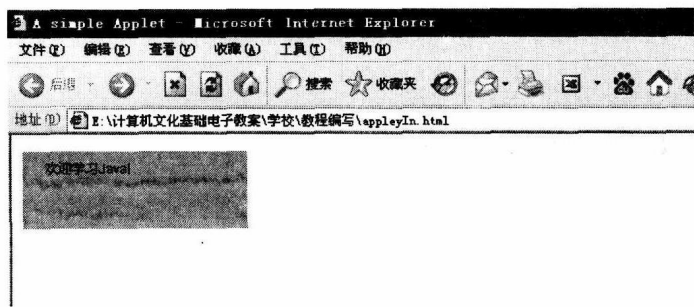


图 1-6 运行 AppletIn.html

注意：用 IE 执行带 Applet 的 HTML 程序较慢，因此，不适合在这种环境下进行测试。但是 appletviewer 工具对 HTML 语言中的一些标志不能识别，所以两种方式对程序运行出的结果可能不同。

1.4.3 Java 程序的基本规则

通过以上两类 Java 程序，可以看出 Java 程序的基本规则：

(1) Java 程序是由类构成的。Java 程序类中有系统提供类库，也可以是根据需要自定义的类。

(2) 每一个 Java 程序都有一个主类，且只有一个主类。无论主类的位置如何，程序

都从主类开始执行。

(3) 调用 Java 程序中的系统类库要在程序的开头用关键字 `import` 引入所需类库的成员。

(4) Java 程序中的每条语句都用一个分号结尾。

(5) Java 程序区分大小写，同一字母的大小写作为两个不同的字符。

(6) Java 程序的文件名必须与主类名取成一致，且大小写相同。

1.5 实践解析

实践 1 请按照模板要求，完成下列代码并转换成 Java 程序代码，并运行。

```
Hello.java
public class Hello
{
    public static void main(Stringp[] args)
    {
        【代码 1】//该处输出“你好，很高兴学习 Java.
        A a = new A();
        a.fA();
    }
}
class A
{
    Void fA()
    {
        【代码 2】//该处输出“We are student.”
    }
}
```

操作步骤：

(1) 打开一个文本编辑器，编辑源文件，把编辑好的文件存成 `.java` 文件。

(2) 在“命令提示符”下进行编译，在命令行中输入“`javac Hello.java`”，当编译结束以后，在 Java 源文件中的每一个类都会生成相应的“`.class`”文件，这时候会生成“`Hello.class`”和“`A.class`”。

(3) 可以在“命令提示符”下输入“`java Hello`”进行执行，该“`java`”命令会启动 Java 虚拟机，并读入“`Property.class`”文件进行执行。

该程序的运行结果直接在命令行下进行输出。

实践 2 Java Applet 小应用程序。

完成下列代码功能并转换为 Java 程序代码并执行。

```
import java.applet.*;
```

```

import java.awt.*;
public class FirstApplet extends Applet
{
    public FirstApplet extends Applet
    {
        public void paint(Graphics g)
        {
            g.setColor(Color.blue);
            【代码 1】//该处输出“这是一个 Java Applet 应用程序”
            g.setColor(Color.red);
            g.setFont(new Font(“宋体”,Font.BOLD,36) );
            【代码 2】//该处输出“我改变了字体”
        }
    }
}

```

操作步骤:

- (1) 将源文件命名为 FirstApplet.java 存放到 C 盘。
- (2) 编译源文件。
- (3) 编写一个 HTML 文件 FirstApplet.html。
- (4) 用浏览器打开 FirstApplet.html。

习 题

1. 完成在 Windows 2000 操作系统下环境变量的配置。
2. 创建一个 Java 程序, 输出自己的基本信息: 名字和年龄。
3. 概述 Java 语言的版本发展过程。
4. 简述 Java 语言的运行机制。
5. 在自己的机器上用 JDK 开发运行环境完成第一个 Java 程序的编辑、编译、运行过程。
6. 简述 Java 语言的特点, Java 虚拟机的概念及作用。