



应用型本科计算机类专业“十三五”规划教材

JAVA 程序设计实训教程

郑 豪 王 峥 王 洁 主编



南京大学出版社

副主编 王小正 侯 青 朱 杰 李 青

南 京 大 学 出 版 社

图书在版编目(CIP)数据

JAVA 程序设计实训教程 / 郑豪, 王峥, 王洁主编

— 南京: 南京大学出版社, 2017. 8

应用型本科计算机类专业“十三五”规划教材

ISBN 978-7-305-19206-7

I. ①J… II. ①郑… ②王… ③王… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 193558 号

出版发行 南京大学出版社

社 址 南京市汉口路 22 号 邮编 210093

出版人 金鑫荣

丛 书 名 应用型本科计算机类专业“十三五”规划教材

书 名 JAVA 程序设计实训教程

主 编 郑 豪 王 峥 王 洁

责任编辑 王秉华 蔡文彬 编辑热线 025-83597482

照 排 南京理工大学资产经营有限公司

印 刷 南京鸿图印务有限公司

开 本 787×1092 1/16 印张 15.75 字数 404 千

版 次 2017 年 8 月第 1 版 2017 年 8 月第 1 次印刷

ISBN 978-7-305-19206-7

定 价 36.80 元

网 址: <http://www.njupco.com>

官方微博: <http://weibo.com/njupco>

官方微信号: njupress

销售咨询热线: (025)83594756

* 版权所有, 侵权必究

* 凡购买南大版图书, 如有印装质量问题, 请与所购
图书销售部门联系调换

前 言

Java 程序设计是计算机学科重要的编程语言之一,Java 程序设计实践教程也成为计算机学科最为重要的实践课程。为深化教育教学改革,进一步推动专业综合改革,依据国家十二五规划,瞄准 Java 发展前沿,面向经济社会发展需求,借鉴国内外教学改革成果,充分利用信息技术,由高校一线教师和知名企业工程师一起编写了这本 Java 程序设计实训教程。

本书以项目实战为主线,深入浅出,把 Java 的基础知识融入到各个实验的项目中,各个实验先以小项目训练读者基础知识,最后再以综合项目实战使用户深入理解。使读者真正的得到了项目实训,掌握了 Java 的程序开发,理解了 Java 的内在机制,也积累了项目开发经验。

全书分为 9 个实验,本书各章内容介绍如下:

实验 1 是初识 Java 程序设计,指导读者掌握 Java 的开发环境、基础语法等;

实验 2 是类的封装和继承,指导读者掌握类的继承性和封装性;

实验 3 是类的多态性,指导读者掌握类的多态性;

实验 4 是接口,指导读者掌握接口,并进一步深入了解多态性;

实验 5 是图形用户界面设计,指导读者能够用 swing 组件编写图形用户界面;

实验 6 是输入输出流,指导读者能够使用输入输出流对文件进行读写;

实验 7 是多线程设计,指导读者编写多线程的程序;

实验 8 是综合性实验 1,指导读者开发一个图像浏览器系统。

实验 9 是综合性实验 2,指导读者开发一个图书馆信息管理系统。

本书的所有实验代码均在 Eclipse 和 JDK 1.7 上通过编译和正常运行。实验 1 是由朱杰老师编写;实验 2 是由王洁老师编写;实验 3 是由王小正老师编写;实验 4, 9 是郑豪老师编写;实验 5, 6 是由侯青老师编写;实验 7 是由李青老师编写;实验 8 是由王峥老师编写。本书编写过程中参考了 Java 程序设计的相关文献,同时还查阅了大量的网络资料,在此对所有的作者表示感谢。

由于编者水平有限,书中不妥和错误之处还望读者批评指正。

编 者

2017 年 5 月

目 录

实验 1 初识 Java 程序设计	1
1.1 知识点回顾	1
1.2 实验练习	4
1.3 项目实战	13
1.4 实验习题	16
实验 2 类的封装和继承	18
2.1 知识点回顾	18
2.2 实验练习	24
2.3 项目实战 1	28
2.4 项目实战 2	30
2.5 实验习题	32
实验 3 类的多态性	34
3.1 知识点回顾	34
3.2 实验练习	38
3.3 项目实战 1	43
3.4 项目实战 2	47
3.5 实验习题	51
3.6 小结	54
实验 4 接口	56
4.1 知识点回顾	56
4.2 实验练习	58
4.3 项目实战 1	70
4.4 项目实战 2	73
4.5 实验习题	77
实验 5 图形用户界面设计	81
5.1 知识点回顾	81
5.2 实验练习	93
5.3 项目实战	102
5.4 综合项目实战	107

5.5 实验习题	144
实验 6 输入输出流(I/O)	146
6.1 知识点回顾	146
6.2 实验练习	150
6.3 实验习题	156
实验 7 多线程	159
7.1 知识点回顾	159
7.2 实验练习	164
7.3 项目实战	173
7.4 实验习题	179
实验 8 综合项目 1—图像浏览器的实现	181
8.1 项目概述	181
8.2 项目需求分析	181
8.3 总体设计	181
8.4 项目文件结构说明	183
8.5 主要代码分析	184
8.6 项目总结	200
实验 9 综合项目 2—图书馆信息管理系统	201
9.1 项目概述	201
9.2 项目需求分析	201
9.3 项目数据库设计	202
9.4 项目总体设计	205
9.5 主要代码分析	208

实验 1 初识 Java 程序设计

1.1 知识点回顾

1. Java 简介

Java 是一种可以撰写跨平台应用软件的面向对象的程序设计语言,是由美国 Sun Microsystems 公司于 1995 年 5 月开发的 Java 程序设计语言和 Java 平台的总称。Java 语言是从 C++ 程序语言发展而来的,但比 C++ 语言简单,它是当前比较流行的网络编程语言。Java 语言的出现是源于对独立于平台语言的需要,即这种语言编写的程序不会因为芯片的变化而发生无法运行或出现运行错误的情况。

Sun 公司对 Java 编程语言的解释是: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、高效、多线程和动态的语言。主要特性如下:

(1) Java 语言简单易用。一方面,Java 语言的灵感主要来自于 C++ 语言,其语法与 C 语言和 C++ 语言接近,有 C++ 基础的开发者会感觉 Java 很熟悉,另一方面,Java 语言摒弃了 C++ 语言中很少使用的、难以理解且令人迷惑的那些特性,如操作符重载、多继承、自动的强制类型转换等。

(2) Java 语言是一种面向对象的程序设计语言。Java 语言只支持单继承,它的多继承是通过实现多接口来完成。

(3) Java 语言是分布式的。Java 拥有广泛的能轻易地处理 TCP/IP 协议的运行库,例如 HTTP 与 FTP 类库等等。这使得在 Java 中比在 C 或 C++ 中更容易建立网络连接。Java 应用程序可以借助 URL 通过网络开启和存取对象,就如同存取一个本地文件系统一样简单。

(4) Java 语言是健壮的。Java 语言的强类型机制、异常处理、内存垃圾的自动收集等是程序健壮性的重要保证;对指针的摒弃是 Java 语言的明智选择。

(5) Java 语言是安全的。

(6) Java 语言是平台无关的。所谓平台无关是指编译后的 Java 程序可直接在不同的平台上运行而不用重新编译,这一特性使得 Java 随着 Web 应用的普及而迅速普及起来。因此只要在操作系统中配有 JVM,就可以运行编译后的 Java 程序,也就是“一次编写,随处运行”,因而轻松实现跨平台。

(7) Java 语言是可移植的。Java 语言规定同一种数据类型在各种不同的实现中,必须占据相同的内存空间。

(8) Java 语言是解释型的。在运行时,Java 平台中的 Java 解释器对这些字节码进行解释执行,执行过程中需要的类在连接阶段被载入到运行环境中。

(9) Java 语言是高性能的。Java 语言是一种半编译半解释执行的语言。

(10) Java 语言是多线程的。在 Java 语言中,线程是一种特殊的对象,它由 Thread 类或

子(孙)类来创建,或者实现 Runnable 接口来创建。Java 语言支持多个线程的同时执行,并提供多线程之间的同步机制来保证对共享数据的正确操作。

(11) Java 语言是动态的。Java 语言需要的类能够动态地被载入到运行环境,也可以通过网络来载入所需要的类。

2. Application 与 Applet

Application(应用程序)和 Applet(小应用程序)是 Java 提供的两种不同类型的程序。Java Application 是一种独立完整的程序,与其他应用程序类似,是可以在计算机操作系统中运行的程序;Java Applet 不是一种独立完整的程序,需要在浏览器这种特定环境下运行。

Java Application 程序的结构特点是:程序是由一个或多个文件组成的,每个文件又是由一个或多个类组成的,每个类是由若干个方法和变量组成的。

Java Applet 应用程序是内嵌于 HTML 文档中,并使用<APPLET>标记的可执行 Java 字节码,是具有动态、安全、跨平台特性的网络应用程序,通过主页发布到 Internet。Java Applet 可以在 Internet 中传输,通过因特网下载并且能在所有支持 Java 的浏览器中运行,它的最大特点是能对用户做出反应,并进行相应的变化。Applet 不能独立在 JVM 中运行,而是由浏览器或 Applet 阅读器(applet viewer)执行。

3. Java 开发运行环境介绍

Java 开发工具分为两大类:一类是基本开发工具,即 Sun 公司免费提供的 Java 2 SDK;另一类是专业开发工具,当今最流行的企业级开发工具有 Eclipse、MyEclipse、JBuilder2008、JDeveloper 等,功能强大,满足企业级开发的需要。

JDK(Java Development Toolkit)是 Sun 公司开发的 Java 开发工具包,它是一个简单的命令行工具,主要包括软件库、编译 Java 源代码的编译器、运行 Java 字节码的解释器,以及测试 Java Applet 的 Applet 阅读器,还有其他一些有用的工具。它主要是通过 DOS 命令行,在 DOS 环境下进行 Java 程序的编译和运行。可以到 Oracle 公司的 Java 语言官方网站上下载最新版的 JDK 软件。安装完 JDK 后,需要设置 3 个系统环境变量,JAVA_HOME 的路径、JDK 开发工具的路径和 CLASSPATH 的路径(具体配置见实验任务 1)。

JCreator 是由 Xinox 软件公司所开发的一个可视化的 Java 程序集成开发环境,它给用户提供了包括工程管理、工程模板、代码实现、代码调试器、高亮语句编辑以及完全客户化的用户界面等广泛的功能。

JBuilder 是由 Borland 公司开发的一款功能强大的可视化 Java 集成开发环境,可以快速开发包括复杂企业级应用系统的各种 Java 程序,包括独立运行程序、Applet 程序、Servlet、JSP、EJB、Web Service 等。

Eclipse 是由 IBM 公司开发的一款开放源码的通用工具平台,它专注于为高度集成的工具开发提供一个全功能的、具有商业品质的工业平台,由 Eclipse 项目、Eclipse 工具项目、Eclipse 技术项目和 Eclipse Web 工具平台项目组成。最常用的 Java 开发功能实际上是 Eclipse 一个主要的插件 JDT(Java Development Tools, Java 开发工具)所提供的,它随 Eclipse SDK 一同发行。Eclipse 是开发源代码的项目,用户可以到<http://www.eclipse.org/downloads/>去免费下载 Eclipse 的安装程序。图 1.1 所示的界面布局是在 Java 视图下的工作主界面。工作台是一个桌面开发环境,提供了使用 Eclipse 工具必需的用户界面,以 Java 视图为例,用户的工作界面包含窗口、菜单栏、工具栏等。在资源管理窗口中文件以“文件/目录”的形式进行管理,可以在该窗口中查找所需的文件资源并可以右键单击文件名进行相关操作。

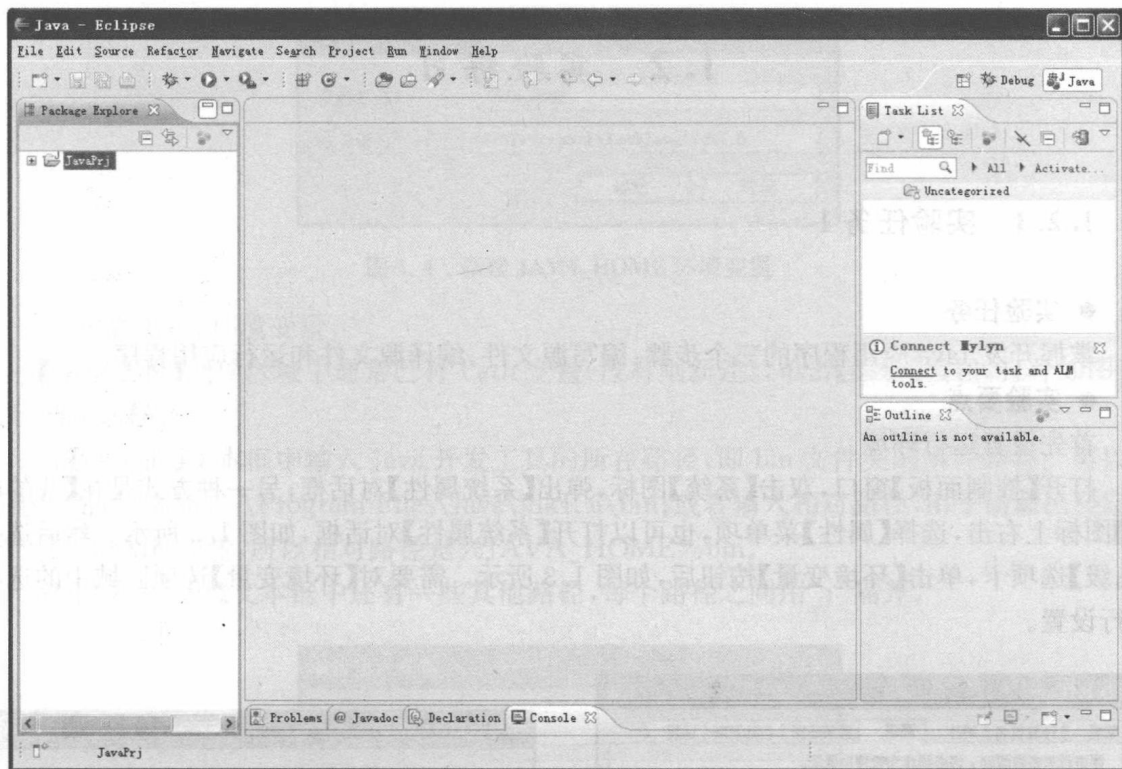


图 1.1 Eclipse 的工作界面

4. Java 核心包

(1) java.lang 包。封装所有编程应用的基本类,如 Object、Class、String、System、Integer、Thread 等。其中 Object 是所有类的根,它所包含的属性和方法被所有类继承。

(2) java.awt 包。封装抽象窗口工具包,提供构建和管理图形用户接口(GUI)设计工具,包含构件、容器和布局管理器。

(3) java.io 包。提供输入/输出文件操作的类。

(4) java.applet 包。为 Applet 提供执行所需的所有类。

(5) java.net 包。提供程序执行网络通信应用及 URL 处理的类。

(6) java.util 包。提供实用程序类和集合类,如系统特性定义和使用、日期方法类、集合 Collection、List、Arrays、Map 等常用工具类。

(7) java.sql 包。提供访问和处理标准数据源数据的类。

(8) java.rmi 包。提供程序远程方法调用所需的类。

(9) javax.naming 包。提供命名服务所需的类和接口。

(10) javax.swing 包。提供构建和管理程序的图形界面的轻量级的构件。

(11) javax.transaction 包。提供事物处理所需的基本类,除此之外,Java 扩展包还提供 Rmi、Sound、Accessibility 等包。

1.2 实验练习

1.2.1 实验任务 1

● 实验任务

掌握开发 Java 应用程序的三个步骤:编写源文件、编译源文件和运行应用程序。

● 实验要点

首先设置运行环境。

打开【控制面板】窗口,双击【系统】图标,弹出【系统属性】对话框;另一种方式是在【我的电脑】图标上右击,选择【属性】菜单项,也可以打开【系统属性】对话框,如图 1.2 所示。然后选择【高级】选项卡,单击【环境变量】按钮后,如图 1.3 所示。需要对【环境变量】选项区域中的选项进行设置。

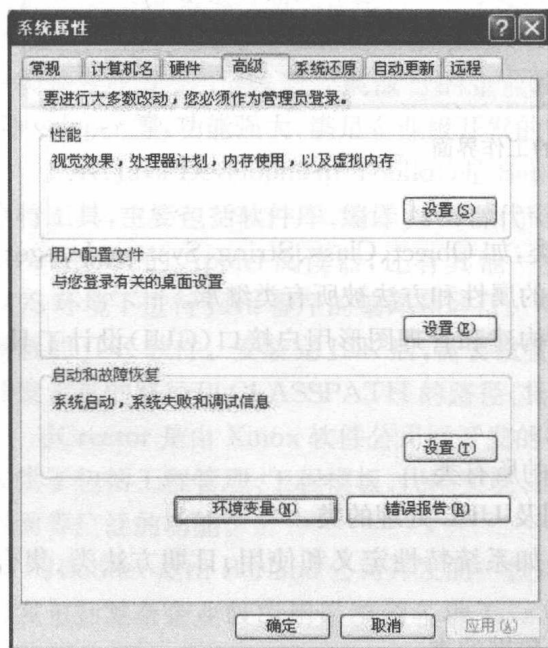


图 1.2 【系统属性】对话框

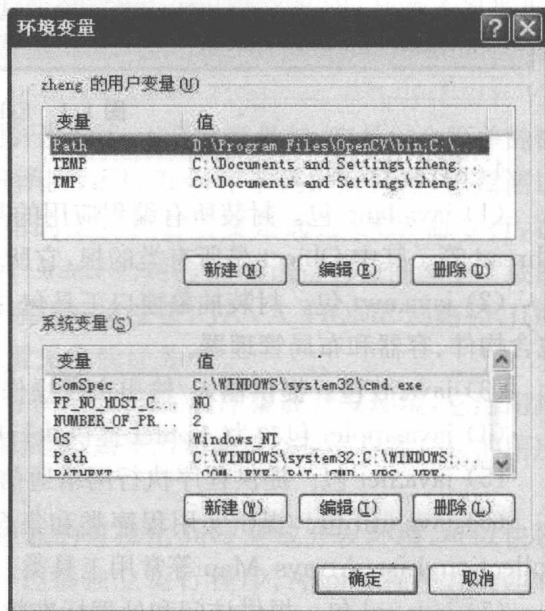


图 1.3 【环境变量】对话框

① 设置 JAVA_HOME 环境变量

【系统变量】选项区域中通常没有 JAVA_HOME 变量,可以单击【新建】按钮,打开如图 1.4 所示的对话框。

在【变量名】文本框中输入 JAVA_HOME,【变量值】文本框中输入安装路径,例如:C:\Program Files\Java\jdk1.6。

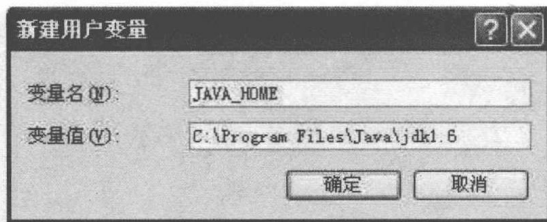


图 1.4 新建 JAVA_HOME 环境变量

② 设置 Path 环境变量

【系统变量】选项区域中通常已有 Path 变量(没有则新建),单击【编辑】按钮,打开如图1.5所示的对话框。

在【变量值】文本框中输入 Java 开发工具的所在路径,即 bin 文件夹的所在路径。可以填写绝对路径,例如 C:\Program Files\Java\jdk1.6\bin;或者输入相对路径,由于前面已经定义 JAVA_HOME 变量,所以相对路径是 %JAVA_HOME%\bin。

注意:【变量值】文本框中还有一些其他路径,每个路径之间用“;”隔开。

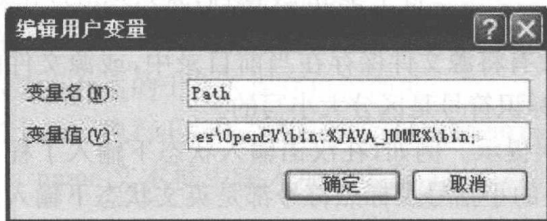


图 1.5 编辑 Path 环境变量

③ 设置 CLASSPATH 环境变量

【系统变量】选项区域中通常已有 CLASSPATH 变量(没有则新建),单击【编辑】按钮,打开如图 1.6 所示的对话框。

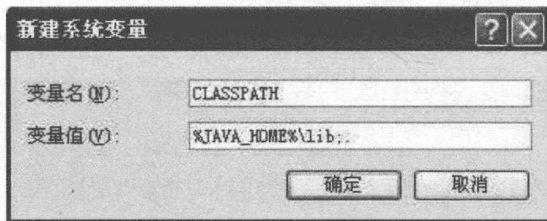


图 1.6 编辑 CLASSPATH 环境变量

Java 源程序一般用.java 作为扩展名,是一个文本文件,用 Java 语言写成,可以用任何文本编辑器来编辑。可以通过“程序”→“附件”→“记事本”来打开 Windows 自带的文本编辑器。程序编写完成后,保存文件。注意:文件名保存为 FirstJava.java。

使用 DOS 命令进入 FirstJava.java 文件所存放的目录(如 E:\java)下,使用 javac 命令编译该应用程序,通过编译该文件夹下会生成一个名为 FirstJava.class 的 class 文件,该文件包含程序的字节码,Java 字节码中包含的是 Java 解释程序将要执行的指令码,使用 java 命令执行该 class 文件,可在控制台下看到执行结果,如图 1.7 所示。

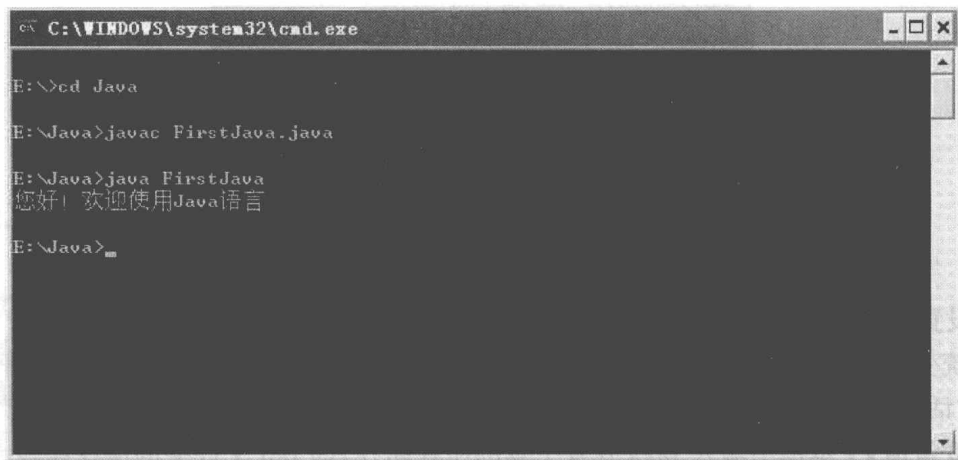


图 1.7 程序 FirstJava.java 在控制台下的输出结果

● 实验分析

可能会遇到下列错误提示:

① Command not Found:没有设置好系统变量 Path。

② File not Found:没有将源文件保存在当前目录中,或源文件的名字不符合有关规定。

要特别注意:Java 语言的标识符号是区分大小写的。

③ 出现一些语法错误提示。例如,在汉语输入状态下输入了程序中需要的语句符号等。Java 源程序中语句所涉及的小括号及标点符号都是英文状态下输入的,比如“你好,欢迎学习 Java”中的引号必须是英文状态下的引号,而字符串里面的符号不受汉语或英语的限制。

④ Exception in thread "main" java.lang. NoClassFoundError:没有设置好系统变量 ClassPath,或运行的不是主类的名字,又或程序没有主类。

● 实验主要代码

```
/**
```

此类用于在屏幕上显示消息。

```
** /
```

```
public class FirstJava
{
    public static void main(String args[])
    {
        System.out.println("您好! 欢迎使用 Java 语言");
    }
}
```

该程序的功能是在显示器屏幕的当前光标处输出显示字符串:你好! 欢迎使用 Java 语言。

(1) 在 Java 中源程序的文件名并不是任意的,它必须和程序中定义的 public 类名相同,扩展名必须是 java。另外,由于 Java 是区分大小写的,所以也应确保文件名的大小写字母和类名一致。

(2) 所有的 Java 应用程序都通过调用 main()方法开始执行,因此在 Application 的声明中对 main()方法的定义是必不可少的。修饰符 static 说明该方法是静态的,它可以在创建对

象之前被调用。

(3) `public class FirstJava` 是一个类,是 Java 程序的基本组成部分。在 Java 中,所有功能都是以类的方式实现的。每一个类都是由关键字 `class` 和类名组成。其中类名是自己定义的,且含有 `main()` 方法的类的类名和源程序文件的文件名必须一致。

(4) 可以添加注释。注释用来解释程序,使程序更好理解,它不会被执行。可以用 `//` 加单行的注释,用 `/* ... */` 或者 `/** ... */` 加多行的注释。

(5) 使用的变量名、对象名、方法名等标识符要有意义,尽量做到“见名知义”。

1.2.2 实验任务 2

● 实验任务

使用 Eclipse 编写 Java 应用程序 `FirstJava.java`,根据命令行参数输出相应信息(如“您好! 欢迎使用 Java 语言”)。

● 实验要点

(1) 打开 Eclipse。双击 `eclipse.exe`,启动 Eclipse 平台。

(2) 配置 Eclipse 的 Workspace。初次打开 Eclipse,会要求用户配置 Eclipse 的 Workspace,点击 `Browse`,选择合适的工作区保存位置即可。

(3) 创建 Java 项目名 Java。执行 `File`—`New`—`Java Project` 菜单命令,打开 New Java Project 对话框。在 Project name 文本框中输入项目名称 Java,然后单击 `Finish` 按钮,如图 1.8 所示。



图 1.8 New Java Project 对话框

(4) 创建 Java 类 FirstJava。执行 File→New→Class 菜单命令,打开 New Java Class 对话框。在 Name 文本框中输入项目名称 FirstJava,选中 public static void main(String[] args)复选框,以自动创建 main()方法框架,然后单击 Finish 按钮,如图 1.9 所示。



图 1.9 New Java Class 对话框

(5) 在工作区编辑 FirstJava 代码。

(6) 运行程序。执行 Run→Run 菜单命令,也可使用快捷键 Ctrl+F11 或单击调试工具栏中的启动运行按钮,运行结果如图 1.10 所示,提示用户输入参数。

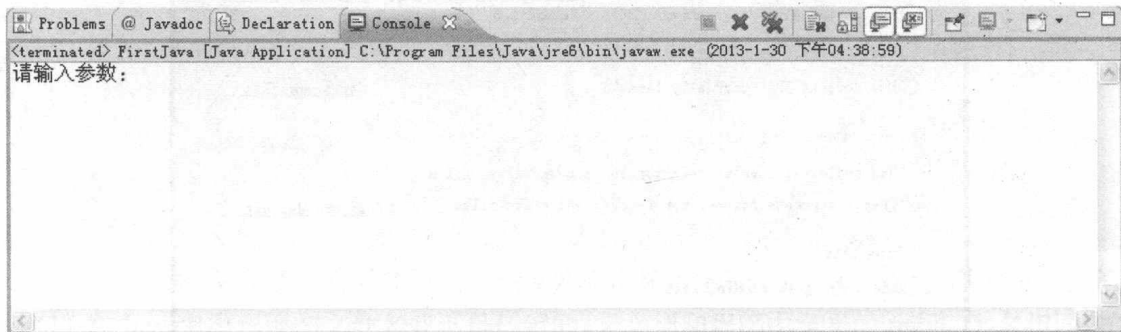


图 1.10 提示用户输入参数

(7) 配置运行参数并调试运行。执行 Run→Run Configurations 菜单命令,打开 Run Configurations 对话框。选择 Arguments 选项卡,在 Program arguments 文本框中输入命令

行参数“欢迎使用 Java 语言”，如图 1.11 所示。

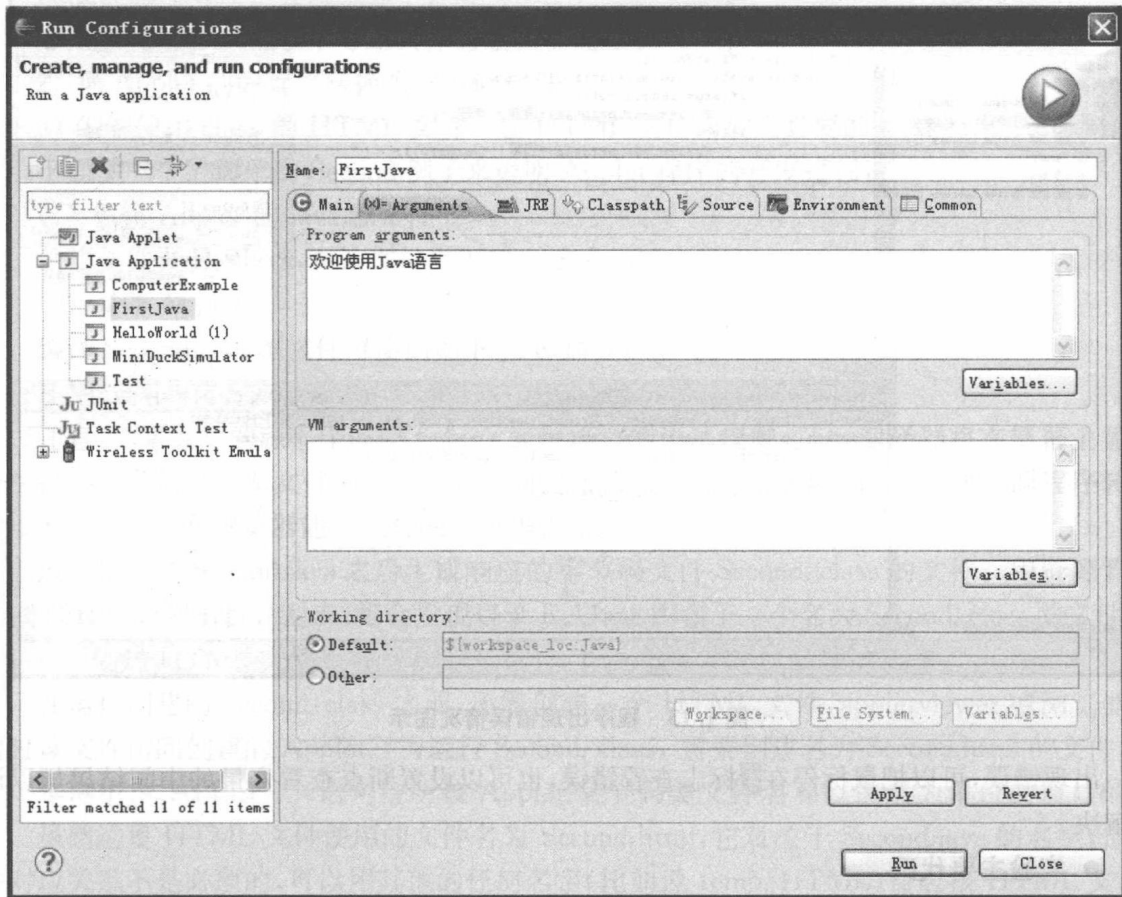


图 1.11 输入命令行参数

(8) 运行结果。单击 Run 按钮,运行结果如图 1.12 所示。

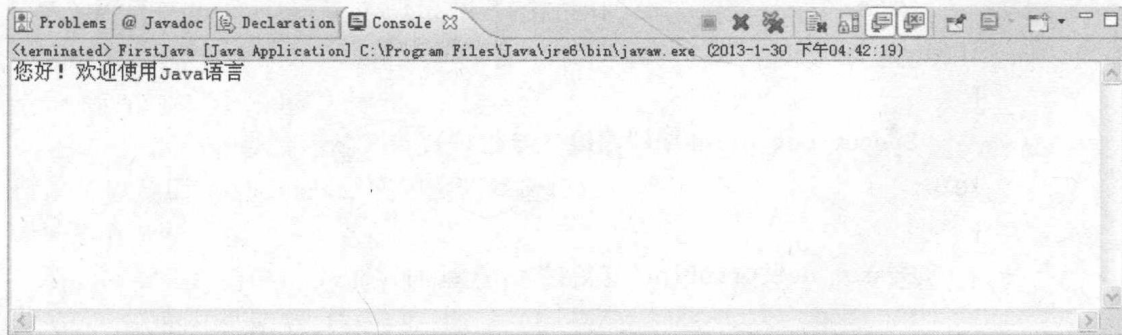


图 1.12 提供命令行参数的运行结果

● 实验分析

源代码编辑结束后,保存源代码,观察代码编辑窗口是否出现 $\times$ 标志。如果在某行代码前出现该符号,说明这行代码存在语法错误,同时在控制台窗口会出现错误提示,需进行修改。比如在输入“System.out.println(“请输入参数”)”语句时没有写“;”,就会出现如图 1.13 所示的错误提示。若没有出现该符号,说明源代码不存在语法错误,可以运行该程序。

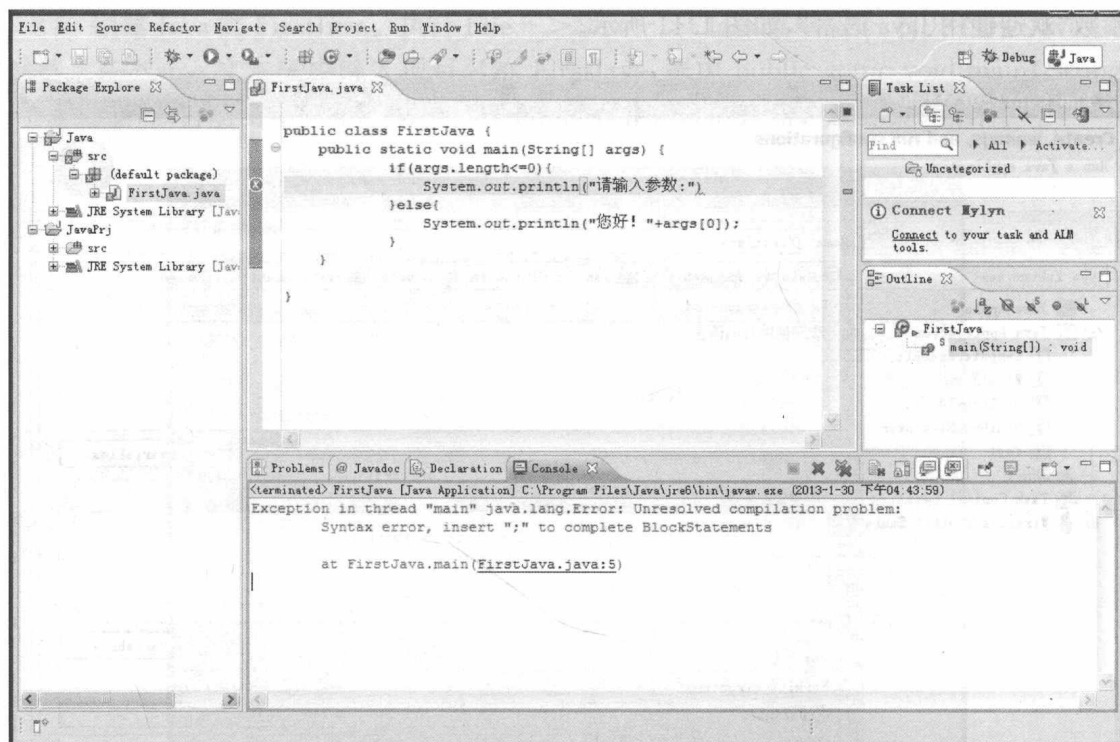


图 1.13 程序出现错误情况图示

出现错误,可以把鼠标停在标志查看错误,也可以设置断点查看变量的中间结果加以调试解决。

● 实验主要代码

```
public class FirstJava
{
    public static void main(String[] args)
    {
        if(args.length<=0)
        {
            System.out.println("请输入参数:");
        }else
        {
            System.out.println("您好!" + args[0]);
        }
    }
}
```

1.2.3 实验任务 3

● 实验任务

学习 Java 中 Java Applet 小程序的使用方法。

● 实验要点

Applet 程序开发主要步骤如下:

- 1) 选用 Eclipse、记事本等工具作为编辑器建立 Java Applet 源程序。
- 2) 把 Applet 的源程序转换为字节码文件。
- 3) 编制使用 class 的 HTML 文件。在 HTML 文件内放入必要的<APPLET>语句。

下面举一个最简单的 Second 例子来说明 Applet 程序的开发过程:

- (1) 编辑 Applet 的 java 源文件

创建文件夹 E:\Java,在该文件夹下建立 Second.java

- (2) 编译 Applet

编译 Second.java 源文件可使用如下 JDK 命令:

```
E:\Java\javac Second.java<Enter>
```

注意:如果编写的源程序违反了 Java 编程语言的语法规则,Java 编译器将在屏幕上显示语法错误提示信息。源文件中必须不含任何语法错误,Java 编译器才能成功地把源程序转换为 appletviewer 和浏览器能够执行的字节码程序。

成功地编译 Second.java 之后生成响应的字节码文件 Second.class 的文件。用资源管理器或 DIR 命令列出目录列表,将会发现目录 E:\Java 中多了一个名为 Second.class 的文件。

- (3) 创建 HTML 文件

在运行创建的 Second.class 之前,还需创建一个 HTML 文件,appletviewer 或浏览器将通过该文件访问创建的 Applet。为运行 Second.class,需要创建名为 Second.html 的文件。

本例中,<APPLET>语句指明该 Applet 字节码类文件名和以像素为单位的窗口的尺寸。虽然这里 HTML 文件使用的文件名为 Second.html,它对应于 Second.java 的名字,但这种对应关系不是必须的,可以用其他的任何名字(比如说 temp.HTML)命名该 HTML 文件。但是使文件名保持一种对应关系可给文件的管理带来方便。

- (4) 执行 Second.html

如果用 appletviewer 运行 Second.html,需输入如下的命令行:

```
E:\java\appletviewer Second.html<ENTER>
```

可以看出,该命令启动了 appletviewer 并指明了 HTML 文件,该 HTML 文件中包含对应于 Second 的<APPLET>语句。

至此,一个 Applet 程序的开发运行整个过程结束了(包括 java 源文件、编译的 class 文件、html 文件以及用 appletviewer 或用浏览器运行)。

● 实验分析

Applet 类是所有 Applet 应用的基类,所有的 Java 小应用程序都必须继承该类。

Applet 类的构造函数只有一种,即:public Applet()。

Applet 类中的四种基本方法用来控制其运行状态:init()、start()、stop()、destroy()。

- (1) init()方法

这个方法主要是为 Applet 的正常运行做一些初始化工作。当一个 Applet 被系统调用时,系统首先调用的就是该方法。通常可以在该方法中完成从网页向 Applet 传递参数,添加用户界面的基本组件等操作。

- (2) start()方法

系统在调用完 init()方法之后,将自动调用 start()方法。而且,每当用户离开包含该