

面向对象程序设计系列教材

Java

与面向对象程序
设计实验指导
与习题集

印 旻

高等教育出版社

面向对象程序设计系列教材

Java 与面向对象程序设计 实验指导与习题集

印 旻

JS84/2/

高等教育出版社

(京) 112 号

内容提要

本书是《Java 与面向对象程序设计教程》的配套上机实验指导和习题集,共安排了 15 个实验,收录了 798 道习题。书中的实验从零起点开始逐步引导读者入门,直至掌握 Java 与面向对象编程的基本原则和主要技术。书中的大量习题除了帮助读者反复强化训练基本概念和基本编程方法之外,还起到了分析常见错误和给出常用示范代码的作用。其中基本习题、综合习题和提高习题的比例大约是 3:1:1,读者可以通过基本习题掌握基本的 Java 编程方法,通过综合习题提高自己的综合编程能力,提高习题则为那些对 Java 和面向对象的编程有余力、有兴趣的读者提供了施展才能的空间。

本书实验安排恰当,例题、习题丰富,除了基本练习题外,许多题目都是实际应用开发的基础,如界面设计、多媒体动画和游戏设计、网络通信、数据库应用等。虽然本书是为《Java 与面向对象程序设计教程》一书配套编写的,但内容相对独立,完全可以配合其他介绍 Java 编程及面向对象技术的教材使用。本书可作为高等学校的师生讲授和学习 Java 与面向对象程序设计的实验教材,也适用于职业教育以及从事实际应用开发的技术人员作为练习掌握 Java 与面向对象编程的参考用书。

本书配有教学辅助课件,可供教师和学生使用。需要者可与清华大学计算中心王行言老师联系

邮政编码: 100084 电话: 010 62782934 E-mail: xywang@tsinghua.edu.cn

图书在版编目(CIP)数据

Java 与面向对象程序设计实验指导与习题集/印旻.

北京: 高等教育出版社, 1999

ISBN 7-04-007502-4

I .J… II .印… III .JAVA 语言-程序设计-自学参考资料
IV .TP312

中国版本图书馆 CIP 数据核字(1999)第 68061 号

Java 与面向对象程序设计实验指导与习题集

印 旻

出版发行 高等教育出版社

社 址 北京市东城区沙滩后街 55 号

邮政编码 100009

电 话 010-64054588

传 真 010-64014048

网 址 <http://www.hep.deu.cn>

经 销 新华书店北京发行所

印 刷 国防工业出版社印刷厂

开 本 787×1092 1/16

版 次 1999 年 11 月第 1 版

印 张 14

印 次 1999 年 11 月第 1 版

字 数 330 000

定 价 18.00 元

凡购买高等教育出版社的图书,如有缺页、倒页、脱页等

质量问题,请与当地图书销售部门联系调换

版权所有 侵权必究

前 言

本书是《Java 与面向对象程序设计教程》一书的配套上机实验指导和习题集，也是笔者在清华大学开设的一门深受欢迎的同名选修课的指定参考书。

全书收录大小习题 798 道，分为问答题、判断对错题、书写程序片断题、程序改错题、读程序写功能题、编程题六种形式，其中基本概念题目约占四分之一，其余为与具体编程有关的题目。问答题主要用于复习和巩固基本概念，覆盖了主要的 Java 编程技术和面向对象程序设计的基本概念，可以作为课后复习知识点的提纲。判断对错题主要针对一些容易混淆或需要反复强调的概念，也包括一些一般读者较为模糊的编程问题。书写程序片断题用于训练读者用 Java 编程的基本功，因为无论多么宏大的程序也离不开它的基石——一句句的语句和一个个的控制结构。程序改错题中的程序片断有长有短，不少来自于笔者在辅导学生上机和批改作业时发现的典型、常见的编程错误；读程序写出其功能或输出的题目中程序较为完整，后面章节中有的程序甚至相当长，读懂它们有一定的难度。之所以这样设计，一是笔者认为读程序同样是软件开发者需要掌握的重要技能，二是希望通过这些完整的有一定代码量的程序示范一些常用的编程原则、技巧或课本未涉及的补充内容。故而这些较长的程序一方面用于锻炼提高读者的读程序能力，另一方面也是提供给读者解决常见问题的范例代码。全书的编程题目有 226 道，这些题目的难度跨度较大，从只需三五行的简单代码到需要调试若干天的小型系统，有的是基本的编程练习，有的是综合提高题目，还有的则供学有余力的读者深入钻研。只有很少的编程题给出了程序答案，这样做原因有三个：一是篇幅所限；二是有志于学习程序设计的读者应该锻炼自己举一反三、触类旁通的能力，很多题目使用的是完全相同的编程技术，作为读者应该重复练习以求巩固所学，作为编者却需避免大量无意义的重复，因为解答常用问题的代码在前面几种题型中大都出现过，所以笔者仅仅挑出需使用未出现过的编程方法的题目或者综合性较强的题目加以解答，以求使全书的代码能够基本覆盖各项常见的 Java 编程技术和方法；最后，编程是一项创造性的活动，编程题的解答从来不是唯一的，笔者不希望读者被书中代码和思路所束缚，许多题目本身就设计为开放式，也非常欢迎读者就书中的题目和程序与笔者切磋探讨。

本书习题的编排次序与《Java 与面向对象程序设计教程》一书的章节目录完全相同，方便读者每学完一章后巩固练习，本书在《Java 与面向对象程序设计教程》一书内容的基础上，增加了一些该书未及收录的补充内容，如桶排序、快速排序、线程的同步等。因为基本覆盖了 Java 程序设计的主要技术，本书也可以配合其他的 Java 编程书籍使用，为方便这部分读者，本书中每一章之前都附有该章的主要知识要点，读者可以据此查找针对不同知识点的习题的具体位置。

在本书的最前面编写了 15 讲实验，其中第一讲实验介绍了 Java 编程工具的安装和使用方法，分为 JDK1.1 和 Visual J++ 6.0 两个版本，读者可以根据具体情况自行选用。这一讲和第二讲是全书的基础，因为计算机是注重实践的学科，纸上谈兵永远无法真正学会编程，本书本着循序渐进、由浅入深的原则，把前面的实验和习题都编写得非常详尽，就是要力促读

者及早上手，当你亲手编写并调试成功自己的第一个 Java 程序时，相信你一定能抛掉对编程的敬畏并对书中剩余的实验跃跃欲试。实验的编排与习题相吻合，实验内容注重训练基本的编程能力，每讲实验的时间基本控制在三到四小时之内。读者可以把它作为衡量自己学习是否合格的标准，能够顺利地全部完成 15 个实验的读者就已经达到了本书对应课程的基本要求，学有余力或学有兴趣的读者可以进一步尝试那些较难的编程题（组成学习小组合作开发是一个非常好的方式）。

总之，笔者的目的是力求把本书写成一本关于 Java 语言与面向对象编程的集入门手册、基本功训练、常见错误解析、程序范例和常见应用介绍等为一体的读物。但是由于个人水平所限，书中难免有疏漏和错误之处，欢迎各位专家同行和广大读者批评指正。

作者

1999 年 9 月

责任编辑 刘建元
封面设计 李卫青
责任印制 杨 明



上 篇
实 验

实验一 安装与熟悉 Java 开发工具

A JDK 开发工具

一、实验目的

1. 学习从网络下载 JDK 开发工具与帮助文档。
2. 学习安装 JDK 开发工具及其文档。
3. 掌握 Java Application 程序的开发过程并编写第一个 Java Application 程序。
4. 掌握 Java Applet 程序的开发过程并编写第一个 Java Applet 程序。
5. 学习编写简单的 HTML 文件配合 Java Applet 使用。
6. 学习使用 JDK 的帮助文档。

二、实验内容

1. 从网络下载 JDK 开发工具与帮助文档

(1) JDK(Java Developer Kit) 是 Sun 公司免费提供给全世界所有 Java 程序员的 Java 开发工具, 它的版本内容更新很快, 总是能够提供最新的 Java 技术。JDK 的全部内容可以从 Sun 公司的站点通过网络下载:

<http://www.sun.com/products/jdk/1.1/index.html>

也可以访问其他国内外的免费 FTP 站点或从 CD-ROM 光盘中下载。

(2) 下载之前首先要选择适合本地实验计算机平台的软件版本, JDK 提供若干平台下的 Java 开发工具, 以安装使用 Windows 95/98/NT 的计算机为例, 应选择“JDK1.1 Win32 Release”版本的 JDK 开发工具。

(3) 选择合适的 JDK 内容版本, JDK1.1 又有若干新分的版本, 如 JDK1.1.3、JDK1.1.4、JDK1.1.5 等, 我们这里以 JDK1.1.5 版本为例(读者可能可以下载到更高的版本, 从一般编程的角度它们之间差别不大)。

(4) 下载结束后将在本地计算机得到一个名为 jdk115-win32.exe 的开发工具安装文件, 在计算机的 C 盘建立一个名为 JDK1.1.5 的目录, 将 jdk115-win32.exe 保存在该目录下。

(5) 同理下载到 JDK 的帮助文档 jdk115-doc.zip 文件, 在 C 盘的 JDK1.1.5 目录下建立子目录 Document, 把 jdk115-doc.zip 文件保存在这个子目录下。

2. 安装 JDK 开发工具与帮助文档

(1) 在命令行运行 jdk115-win32.exe 文件, 或者在图形界面双击其图标运行该文件, 根据自动安装过程的提示将 JDK 安装在指定的目录下, 例如, 安装在 C:\JDK1.1.5\目录下。

(2) 设置环境变量 PATH。修改计算机的 C:\AUTOEXEC.BAT 文件, 在其中增加一句

```
SET PATH = %PATH%; C:\JDK1.1.5\BIN
```

这样，程序员可以在任何目录下运行 C:\JDK1.1.5\BIN 目录中的 Java 开发工具，例如 Java 编译器 javac.exe 和 Java 解释器 java.exe。

(3) 设置环境变量 CLASSPATH。修改计算机的 C:\AUTOEXEC.BAT 文件，增加一句

```
SET CLASSPATH = %CLASSPATH%; .; C:\JDK1.1.5\LIB; C:\JDK1.1.5\LIB\CLASSES.ZIP
```

CLASSPATH 环境变量将指定 Java 字节码文件.class 的缺省目录。

(4) 使用解压缩工具 Winzip 解压 JDK 帮助文档 jdk115-doc.zip 文件，将获得一组以 HTML 文件形式组织的帮助文件，把它们保存在本地的一个目录下，例如保存在 C:\JDK1.1.5\Document\目录下。

3. 编写并运行第一个 Java Application 程序

(1) 打开一个纯文本编辑器，例如 NotePad 写字板或者 Edit.exe。

(2) 键入如下程序：

//第一个 Java Application 程序：MyFirstJavaProgram.java

```
import java.io.*;
public class MyFirstJavaProgram
{
    public static void main(String args[])
    {
        System.out.println("This is my first Java program!");
    }
}
```

(3) 检查无误后（注意大小写准确）把文件保存起来，命名为 MyFirstJavaProgram.java，可以创建如下的目录保存这个文件：C:\JavaExperiment\exp1\Application\，调用 dir 命令观察该目录下的所有文件。

(4) 进入命令行方式(MS-DOS)，在保存有上述.java 文件的目录下运行 Java 编译器：

```
C:\JavaExperiment\exp1\Application>javac MyFirstJavaProgram.java
```

(5) 如果 JDK 安装正确、程序输入无误且保存的文件名正确，运行编译器没有任何输出；否则会输出错误信息，一般的错误都是由于拼写失误引起的。再运行 dir 命令查看是否已在相同目录下生成一个名为 MyFirstJavaProgram.class 的文件。

(6) 利用 Java 解释器运行这个 Java Application 程序，并查看运行结果：

```
C:\JavaExperiment\exp1\Application>java MyFirstJavaProgram
```

4. 编写并编译第一个 Java Applet 程序

(1) 打开一个纯文本编辑器，例如 NotePad 写字板或者 Edit.exe。

(2) 键入如下程序：

//第一个 Java Applet 程序：MyFirstJavaApplet.java

```
import java.applet.Applet;
import java.awt.Graphics;
public class MyFirstJavaApplet extends Applet
{
    public void paint(Graphics g)
```

```
{  
    g.drawString("This is my first Java Applet!",15,20);  
}  
}
```

(3) 检查无误后（注意大小写准确）把文件保存起来，命名为 `MyFirstJavaApplet.java`，可以创建如下的目录保存这个文件：`C:\JavaExperiment\exp1\Applet\`。

(4) 进入命令行方式(MS-DOS)，在保存有上述 `java` 文件的目录下运行 `Java` 编译器：

```
C:\JavaExperiment\exp1\Applet>javac MyFirstJavaApplet.java
```

(5) 如果程序输入无误且保存的文件名正确，运行编译器将没有任何输出；否则会输出错误信息，一般的错误都是由于拼写失误引起的。再运行 `dir` 命令查看是否已在相同目录下生成一个名为 `MyFirstJavaApplet.class` 的文件。

5. 编写配合 Applet 的 HTML 文件

(1) 打开一个纯文本编辑器，例如 `NotePad` 写字板或者 `Edit.exe`。

(2) 键入如下的 HTML 程序：

```
<html>  
<head><title>SimpleGraphicsInOut</title></head>  
<body>  
<hr>  
<applet    code= MyFirstJavaApplet  
           width=300  
           height=200>  
</applet>  
</body>  
</html>
```

(3) 检查无误后把文件命名为 `MyFirstJavaApplet.html`，与文件 `MyFirstJavaApplet.java` 保存在同一个目录下。

(4) 直接双击这个 HTML 文件的图标，或者打开 Web 浏览器（例如 IE4.0），在地址栏中键入这个 HTML 文件的全路径名 (`C:\JavaExperiment\exp1\Applet\MyFirstJavaApplet.html`)，查看 Applet 在浏览器中的运行结果。

(5) 利用模拟的 Applet 运行环境解释运行这个 `Java Applet` 程序并观察运行结果：

```
C:\AppletViewer MyFirstJavaApplet.html
```

6. 使用 JDK 帮助文档

(1) 找到 `C:\JDK1.1.5\Document\` 目录或其子目录（取决于解压缩时选定的目录）下的 `index.html` 文件，双击它或者在 Web 浏览器的地址栏中输入这个文件的全路径名，打开这个文件对应的网页。

(2) 点击 `README` 一项了解 `JDK` 安装与使用的有关知识。

(3) 返回 `index.html`，点击 `Java Platform Core API` 浏览 `Java` 的类库。

B Visual J++ 6.0 集成开发环境

一、实验目的

1. 学习安装 Visual J++ 6.0（以下也简称 VJ++）集成开发环境。
2. 掌握集成开发环境(IDE)的概念，了解并学习使用 Visual J++ 6.0 开发环境的基本构成和功能。
3. 了解“项目”的概念，掌握利用 VJ++开发并编译运行一个 Java Application 的过程。
4. 了解“可视化开发”的概念，掌握利用 VJ++的向导开发一个 Java Applet 的过程。
5. 初步了解 VJ++的网络功能和调试功能。

二、实验内容

1. 安装 Visual J++ 6.0

Visual J++ 6.0 集成开发环境可以从 CD-ROM 光盘上安装，也可以从网络上利用共享目录安装，整个安装过程是自动完成的，用户只要给出安装到本地机的目的地目录、选择安装选件（通常选择“典型安装”）即可。

2. 编写并运行第一个 Java Application 程序

Visual J++ 6.0 与其他开发工具一样，是以“项目”(Project)为单位来组织应用开发的，编程人员所写的每一个 Java 源代码程序文件都必须属于一个 Project，而每个 Project 里则可以包含不止一个 .java 文件和其他类型的文件。我们知道，开发一个完整的应用，仅仅有一个源程序是远远不够的，通常需要若干个源程序文件和辅助性文件，如 C/C++ 中的 .h 头文件等；同时为了保证开发的连续性，开发工具通常还会记录用户的桌面安排和其他种种设置；所有这些与一个应用开发任务相关的文件被放在一个文件夹（目录）中，合称为一个 Project。每个 Project 都有一个项目文件用于管理项目中的其他所有文件，在 VJ++ 中这个文件以 .sln 为后缀，只要在 VJ++ 开发环境中打开这个文件就可以进行所有关于这个项目的开发工作。

(1) 新建一个项目

双击 Visual J++ 6.0（以下也简称 VJ++）的图标或通过 Windows 95/98 “开始”菜单的“程序”选项选择“Microsoft Visual J++ 6.0”，打开 VJ++ 开发环境，出现如图 1.1 所示的“New Project”窗口。

对话框左边的三个图标分别代表三种不同的生成 Java Application 的模板：第一个图标代表生成一种在 Windows 平台上专用的 Java Application（其中使用了 Windows 基础类库 WFC）；第二个图标代表生成字符界面的 Java Application（Console 是字符界面的控制台，运行在字符界面的 Java Application 也必须是字符界面的程序）；第三个图标提供一个生成图形界面 Application 的向导，利用此向导可以生成包含菜单、编辑器、工具条和状态条的图形界面，并把程序打包成可以直接运行的 .exe 文件。

我们暂时不选用任何一种模板，而是选中图标左边区域的“Visual J++ Projects”一项，此时右边的图标变化成一个“Empty Project”，选中它如图 1.2 所示。然后在对话框下方的

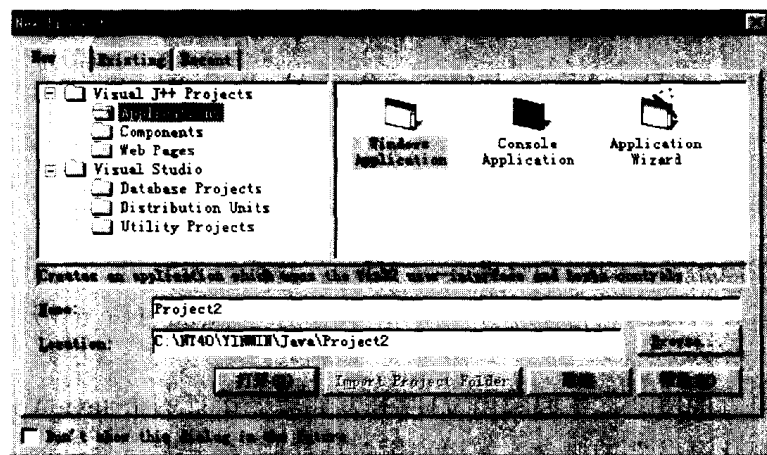


图 1.1 新建项目(New Project)对话框

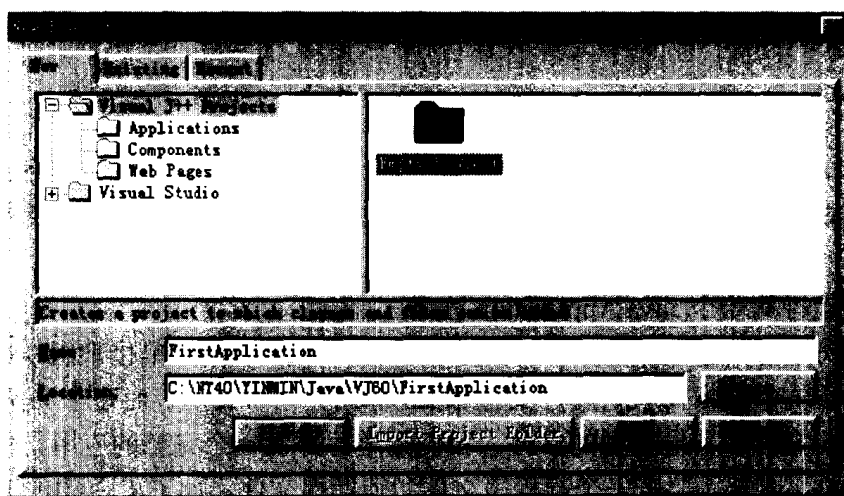


图 1.2 确定 Application 的类型、位置与名称

“Location”一栏输入新建项目所在的目录文件夹，在“Name”一栏中输入新建项目的名称，然后点击“打开”按钮关闭对话框。稍待片刻，系统将在指定 Location 目录中为这个项目生成一个指定名称的子目录，保存该项目中的所有文件。

(2) 加入并编辑 Java Application 源程序

选中位于屏幕右上方的窗口“Project Explorer -”，选中 FirstApplication 一项，如图 1.3 所示。用鼠标右键点击 FirstApplication 一项的图标，选择弹出菜单的 Add 菜单项的 Add Class

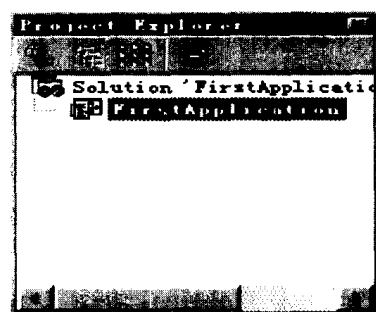


图 1.3 项目浏览器

子菜单项，得到如图 1.4 所示的对话框。

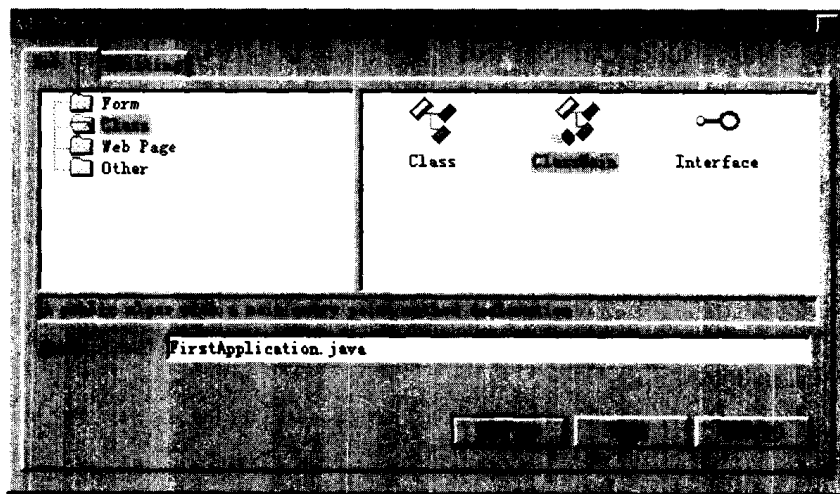


图 1.4 添加 Class 的 Add Item 对话框

由于我们要创建的是程序中唯一的类，即主类，所以选中右边第二个图标，然后在 Name 栏中输入主类的名字 `FirstApplication.java`，点击“打开”按钮，系统将创建一个名为 `FirstApplication.java` 的源代码文件，将其加入到 Project Explorer 窗口中并打开一个用于编辑这个源代码文件的窗口，其中包含系统为我们自动生成的几行代码（如图 1.5 所示）。在编辑窗口中加入一行代码如图 1.6 所示。

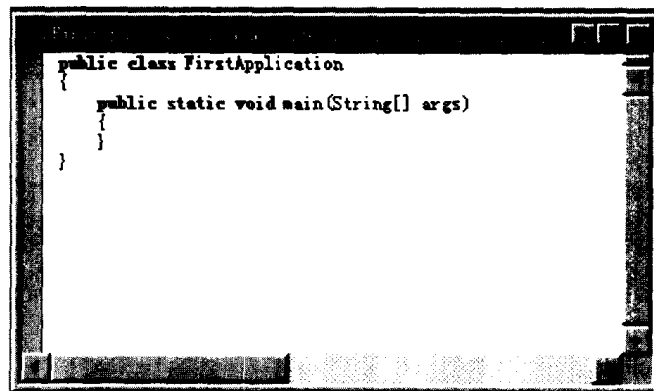


图 1.5 源代码编辑窗口

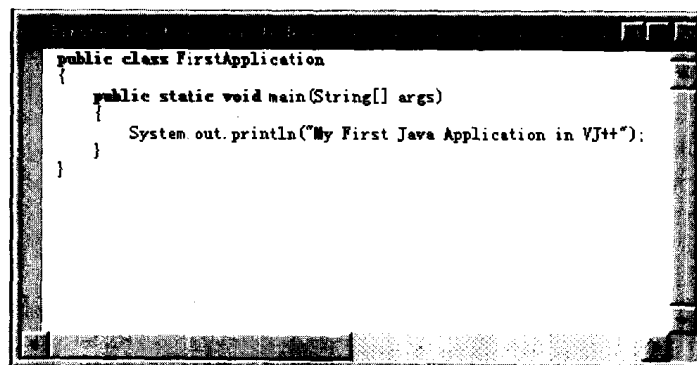


图 1.6 在编辑窗口中编辑源代码

然后选择屏幕左上角 File 菜单的“Save FirstApplication.java”选项保存源代码。

(3) 编译并运行 Java Application 程序

选择 Build 菜单的“Build Ctrl+Shift+B”选项编译该程序，屏幕最下方的状态条将显示“Solution update successfully”，表示编译成功。

选择 Debug 菜单的 Start 选项即可运行此程序，程序第一次运行时，系统需要了解这个项目的有关属性，将弹出一个如图 1.7 所示的 FirstApplication 项目的属性窗口，在 Launch 页面中选中“Launch as a console application”复选框，按 OK 按钮关闭这个窗口，程序就开始运行。

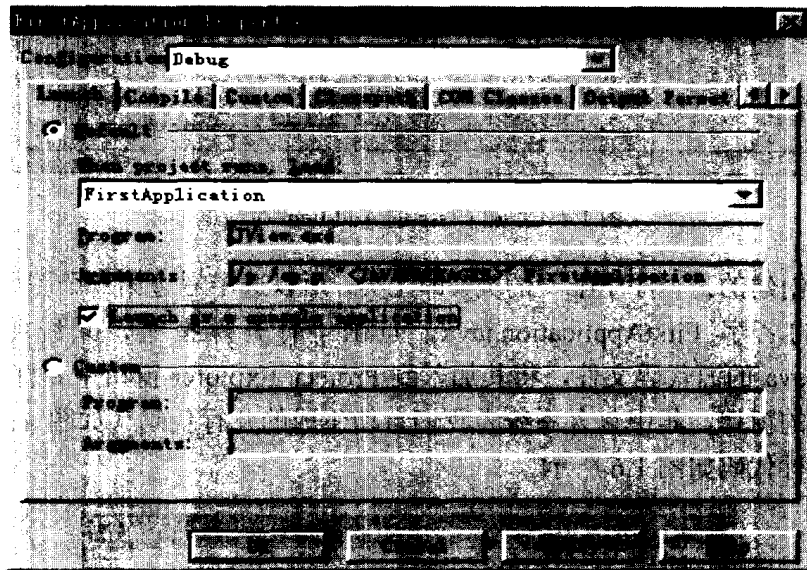


图 1.7 项目属性窗口

由于我们的 Application 程序是采用字符界面，所以系统将打开一个字符界面的窗口，在其中运行 FirstApplication.java 程序。这个字符界面的窗口将在程序运行完毕后自动关闭，读者可能来不及查看运行结果。为解决这个问题，可以把源代码修改为：

```
import java.io.*;
public class FirstApplication
{
    public static void main(String[] args)
    {
        System.out.println("My First Java Application in VJ++");
        try{
            System.in.read();
        }catch(IOException e){}
    }
}
```

这个程序在输出后将等待用户键入一个回车键后才关闭运行结果窗口。

3. 利用模板编写并生成第一个 Java Applet 程序

(1) 新建一个 Applet 项目

选择 File 菜单的“New Project...”选项,弹出 New Project 对话框,选择对话框左边区域中的“Web Pages”一行,则右边出现两个图标,选中第一个模板“Applet on HTML”。然后在 Location 一栏中输入新建项目的父目录,在 Name 一栏中输入新建项目(及其子目录)的名称 FirstApplet(如图 1.8 所示),点击 OK 按钮关闭 New Project 对话框,系统将在指定目录下创建项目。

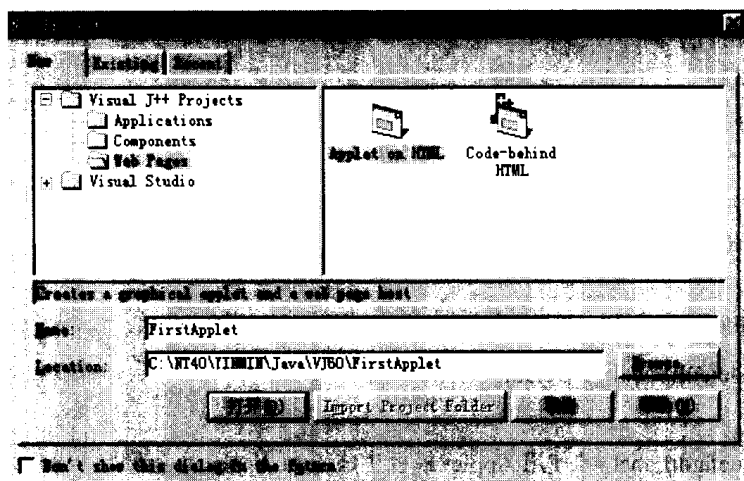


图 1.8 利用 Applet 模板生成新项目

(2) 编辑 Applet 源代码

选择屏幕左上方的“Project Explorer-FirstApplet”窗口,用鼠标双击其中的“FirstApplet”一行打开这个项目,可以看到系统为项目生成的两个源文件“Applet1.java”和“Page1.htm”(如图 1.9 所示),其中 Applet1.java 定义一个名为 Applet1 的 Applet 的子类,Page1.htm 是一个内嵌 Applet1.class 的 HTML 文件。选中文件所在行并点击鼠标右键,选择“rename”菜单,可以将文件名修改为用户定义的名称,例如把 Applet1.java 修改为 FirstApplet.java,把 Page1.htm 修改为 FirstApplet.htm。

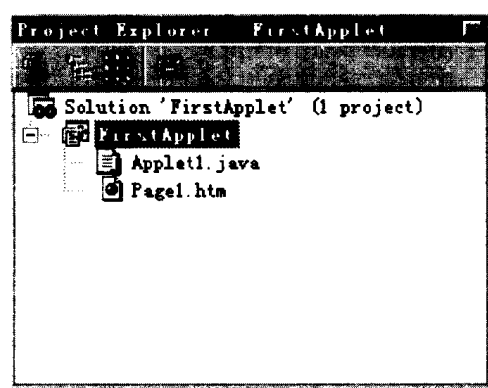


图 1.9 新建的 Applet 项目

修改了文件名称之后,还要相应地修改系统生成的代码,双击 Project Explorer 窗口中的 FirstApplet.java 一行打开其源代码编辑窗口。窗口中的代码很多,我们共做三处修改。

① 找到程序中如下的语句:

```
public class Applet1 extends Applet
```

修改为

```
public class FirstApplet extends Applet
```

- ② 找到程序中的 `init()` 方法，增加一个语句如下所示：

```
public void init()
{
    initForm();

    usePageParams();

    // TODO: Add any constructor code after initForm call.
    setSize(300,100);    //新增语句
}
```

- ③ 找到程序中 `usePageParams()` 方法中的如下语句：

```
final String defaultLabel = "Default label";
```

修改为

```
final String defaultLabel = "使用 Applet 模板生成 Java Applet 小程序";
```

编辑完成后保存此程序。

(3) 编译运行 Applet

选择 Build 菜单的 Build 菜单项，编译无误后选择 Debug 菜单的 Start 菜单项，使用 VJ++ 提供的一个类似于 AppletViewer 的 Applet 运行环境直接运行这个 Applet 程序。程序第一次运行时弹出如图 1.10 所示的 Project Properties 窗口。不修改系统的缺省设置，按 OK 按钮运行这个项目，得到如图 1.11 所示的运行结果。

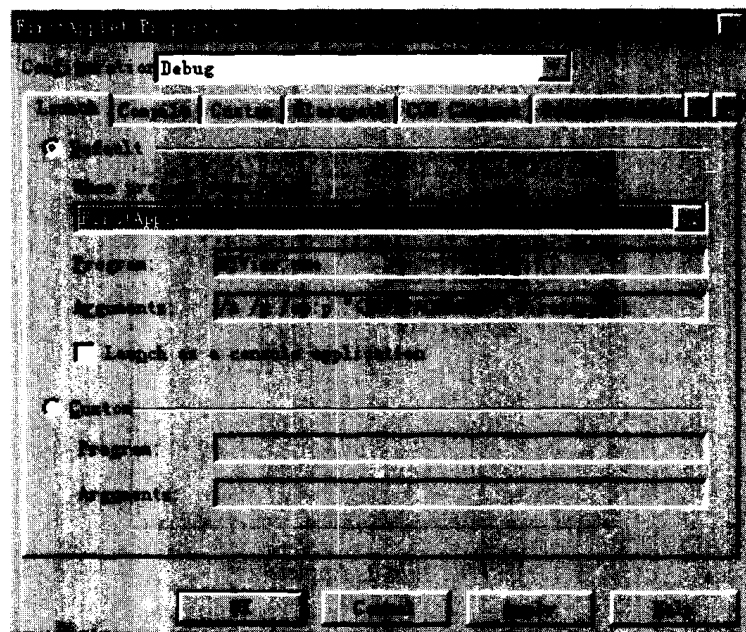


图 1.10 Applet 项目属性窗口

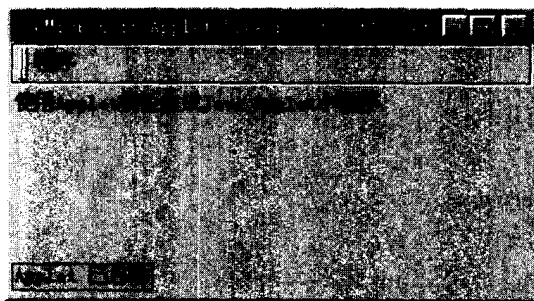


图 1.11 Applet 的直接运行结果

(4) 编辑 HTML 文件

如果需要 Applet 程序与 HTML 文件配合工作, VJ++ 同样提供了编辑 HTML 文件的环境, 选择 Project Explorer 窗口, 双击 FirstApplet.htm 一行, 弹出如图 1.12 所示的 HTML 编辑窗口。

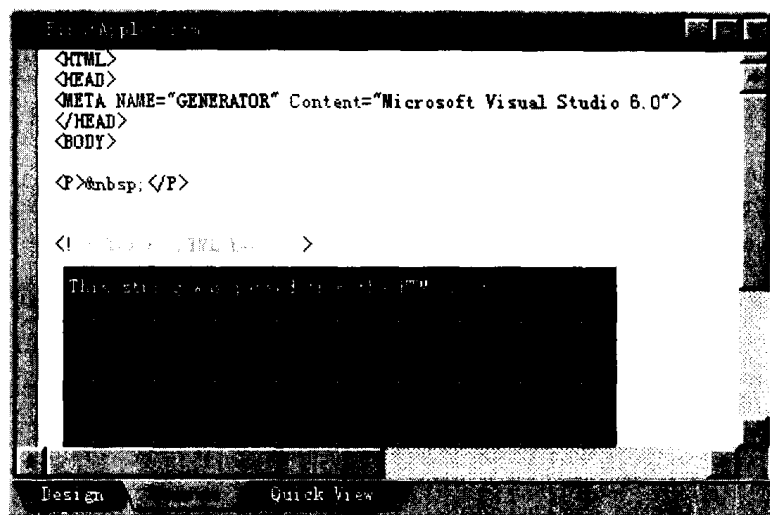


图 1.12 HTML 文件的编辑窗口

这个窗口有三页视图: 第一页 Design 中, 可以利用屏幕左边 Toolbox 窗口中的 HTML 工具可视化地加入一些图形界面组件, 如文本区域、按钮等 (注意: 这些组件是加入在 HTML 文件中, 不是 Applet 中); 第二页 Source 将显示 HTML 的具体代码, 其中蓝色的区域是 HTML 中所嵌入的 Applet 的显示效果; 第三页 Quick View 将显示整个 HTML 在浏览器中的显示效果。

选择第二页视图, 选中代表 Applet 的蓝色区域, 点击鼠标右键, 在弹出的菜单中选择 “Always View As Text” 菜单项, 可以看到如下的用于嵌入 Applet 的 HTML 语句:

```
<applet
  code=Applet1.class
  name=Applet1
  width=320
  height=200 VIEWASTEXT>
<param name=label value="This string was passed from the HTML host.">
<param name=background value="008080">
```