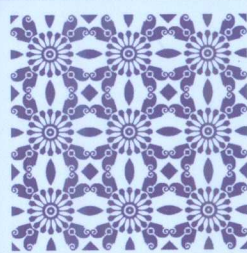


Java程序设计 习题精析与实验指导

施霞萍 王瑾德 史建成 马可幸 编著



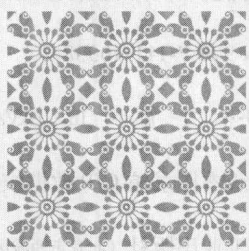
机械工业出版社
China Machine Press

高等院校计算机教材系列

Java程序设计 习题精析与实验指导

施霞萍 王瑾德 史建成 马可幸 编著

福建师范大学
图书馆
藏书印记



1043449



T1043449



机械工业出版社
China Machine Press

本书是为《Java 程序设计教程（第3版）》一书配置的教学辅导教材。书中为主教材的教学提供了重点复习、习题解析、实验指导、练习题以及综合练习等内容，内容丰富、形式多样、实用性强。书后的附录给出了所有实验内容、练习题以及综合练习的参考答案。

本书可作为学习 Java 程序设计语言的参考资料，也可作为相关课程的配套教材，是《上海市高等学校计算机等级考试（二级）——Java 程序设计》考试科目的指定参考书。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目（CIP）数据

Java 程序设计习题精析与实验指导 / 施霞萍等编著. —北京：机械工业出版社，2013.1
（高等院校计算机教材系列）

ISBN 978-7-111-40538-2

I . J… II . 施… III . JAVA 语言—程序设计—高等学校—教学参考资料 IV . TP312

中国版本图书馆 CIP 数据核字（2012）第 280543 号

机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码 100037）

责任编辑：余洁

北京市荣盛彩色印刷有限公司印刷

2013 年 1 月第 1 版第 1 次印刷

185mm×260mm • 14.25 印张

标准书号：ISBN 978-7-111-40538-2

定价：33.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991；88361066

购书热线：(010) 68326294；88379649；68995259

投稿热线：(010) 88379604

读者信箱：hzjsj@hzbook.com

前 言

Java 是目前国内外广泛应用的程序设计语言，它是面向对象技术成功应用在程序设计中的著名典范，其特有的性质使它随着互联网应用的发展而被广泛使用。

本书是为《Java 程序设计教程（第 3 版）》一书配置的教学辅导教材，书中为主教材的教学提供了完整的学习辅导材料。全书共有 11 章，前十章与主教材的章节对应，是对教材知识的学习指导，每章包括重点复习、习题解析、实验指导和练习题四部分。“重点复习”部分对知识重点和难点进行了概括和总结，使学习者能把握关键。“习题解析”部分以常见的选择题和程序填空的形式对相关知识点进行解析。“实验指导”部分对每一章的实验练习提供实验目的和要求，先由实验样例给出实验思路和方法，然后由实验内容进行实践和检验。实验样例有程序改错和编程两种方式，实战性强。“练习题”部分则提供了大量的习题供学习者检验学习效果。第 11 章是综合练习部分，一共提供了三套仿真练习题，题量和形式与上海市高等学校计算机等级考试最新版试卷形式完全一样，能为学习者提供相关知识和实践的检验。书后的附录给出了所有实验内容、练习题以及综合练习的参考答案。

本书的内容结合《Java 程序设计教程（第 3 版）》教材的知识体系由浅入深、循序渐进地从多个角度对相关知识点进行辨析，因此可作为学习 Java 程序设计语言的参考资料，也可作为相关课程的配套教材。本书所有的习题形式与上海市高等学校计算机等级考试（二级）同步，很多例题来源于曾经的真实试题，是《上海市高等学校计算机等级考试（二级）——Java 程序设计》考试科目的指定参考书。

建议读者在使用该书时，坚持独立思考和积极动手实践的原则，通过从对样例的模仿到独立设计调试的渐进过程，最终提高 Java 程序设计能力。

本书由施霞萍、王瑾德、史建成、马可幸编著。机械工业出版社华章分社的编辑及相关同志对本书的出版给予了充分的重视和周到的安排，使得本书得以在短时间内完成出版。本书在写作前的酝酿和整个写作过程中，得到了上海市教育委员会高教处和上海市教育考试院相关领导的指导和支持，并给出很多指示，为本书的定位和内容的选择提供了方向。为此对一切曾经鼓励、支持和帮助过我们的组织、领导和朋友表示真挚的感谢。

尽管本书作者都有 10 年以上的大学教龄并积累了许多程序设计方面的教学经验，但由于时间仓促和水平有限，书中难免有不妥之处，竭诚欢迎读者多提宝贵意见。电子邮箱：sxp@dhu.edu.cn。

施霞萍

2012 年 10 月

目 录

前言

第 1 章 Java 程序设计概述 1

1.1 重点复习 1

1.1.1 Java 程序设计语言的发展 1

1.1.2 Java 程序设计语言的特点 1

1.1.3 Java 相关程序介绍 1

1.1.4 Java 程序及开发过程 1

1.2 习题解析 2

1.3 实验指导 3

1.3.1 实验目的与要求 3

1.3.2 实验样例 4

1.3.3 实验内容 9

1.4 练习题 9

第 2 章 Java 程序设计的基本概念 10

2.1 重点复习 10

2.1.1 数据类型及标识符 10

2.1.2 常量的定义与使用 10

2.1.3 变量的定义与使用 10

2.1.4 运算符与表达式的使用 10

2.1.5 数据类型的转换 11

2.2 习题解析 11

2.3 实验指导 14

2.3.1 实验目的与要求 14

2.3.2 实验样例 14

2.3.3 实验内容 16

2.4 练习题 18

第 3 章 Java 的结构化程序设计 20

3.1 重点复习 20

3.1.1 顺序结构及基本语句 20

3.1.2 选择结构语句 20

3.1.3 循环结构语句 21

3.1.4 数组 21

3.1.5 方法 21

3.2 习题解析 22

3.3 实验指导 30

3.3.1 实验目的与要求 30

3.3.2 实验样例 30

3.3.3 实验内容 38

3.4 练习题 40

第 4 章 Java 的面向对象程序设计 44

4.1 重点复习 44

4.1.1 面向对象程序设计的基本思想 44

4.1.2 类的创建方法 44

4.1.3 类的成员以及声明方式 44

4.1.4 对象的创建和构造方法 45

4.1.5 对象的使用方法和销毁机制 45

4.1.6 类的封装与四种访问权限 45

4.1.7 类成员（静态成员）的声明和 使用 46

4.1.8 类的继承及应用 46

4.1.9 null、this、super 对象运算符 的意义和使用 47

4.1.10 最终类和抽象类及相关方法 的定义 47

4.1.11 多态的两种应用方式：重载 和覆盖 48

4.1.12 接口的声明和实现 48

4.1.13 Java 常用包及用户自定义包 48

4.2 习题解析	49	6.3.3 实验内容	120
4.3 实验指导	56	6.4 练习题	121
4.3.1 实验目的与要求	56	第7章 Java 的多线程程序设计	126
4.3.2 实验样例	57	7.1 重点复习	126
4.3.3 实验内容	59	7.1.1 线程的相关概念	126
4.4 练习题	61	7.1.2 创建线程的方法	126
第5章 Java 的图形用户界面	67	7.1.3 线程的状态及转换控制	126
5.1 重点复习	67	7.1.4 线程的优先等级与调度原则	126
5.1.1 图形用户界面	67	7.1.5 线程的同步控制机制	127
5.1.2 组件	67	7.2 习题解析	127
5.1.3 容器	67	7.3 实验指导	129
5.1.4 布局管理器	68	7.3.1 实验目的与要求	129
5.1.5 Java 的事件及事件处理机制	69	7.3.2 实验样例	129
5.1.6 Applet	70	7.3.3 实验内容	131
5.1.7 Swing 组件	71	7.4 练习题	131
5.1.8 二维图形设计	71	第8章 Java 的输入输出流	132
5.2 习题解析	72	8.1 重点复习	132
5.3 实验指导	77	8.1.1 数据流与输入输出	132
5.3.1 实验目的与要求	77	8.1.2 字节流及其常用方法	132
5.3.2 实验样例	78	8.1.3 字符流及其常用方法	133
5.3.3 实验内容	90	8.1.4 文件类及其常用方法	133
5.4 练习题	97	8.1.5 I/O 流与文件的操作	134
第6章 Java 的异常处理	107	8.1.6 文件对话框与文件的操作	134
6.1 重点复习	107	8.2 习题解析	134
6.1.1 异常类与错误类	107	8.3 实验指导	138
6.1.2 异常类结构及常见异常类型	107	8.3.1 实验目的与要求	138
6.1.3 异常处理机制	107	8.3.2 实验样例	139
6.1.4 throw 和 throws 语句	108	8.3.3 实验内容	144
6.1.5 自定义异常的一般步骤	108	8.4 练习题	147
6.2 习题解析	108	第9章 Java 的网络应用	150
6.3 实验指导	114	9.1 重点复习	150
6.3.1 实验目的与要求	114	9.1.1 网络的基本概念	150
6.3.2 实验样例	114	9.1.2 URL 的使用	150

9.1.3 Socket 的应用	151	10.1.2 动画实现	161
9.2 习题解析	152	10.1.3 声音播放	161
9.3 实验指导	154	10.2 习题解析	161
9.3.1 实验目的与要求	154	10.3 实验指导	163
9.3.2 实验样例	154	10.3.1 实验目的与要求	163
9.3.3 实验内容	156	10.3.2 实验样例	163
9.4 练习题	157	10.3.3 实验内容	165
第 10 章 Java 的多媒体应用	160	10.4 练习题	166
10.1 重点复习	160	第 11 章 综合练习	168
10.1.1 图像显示	160	附录 参考答案	184

第 1 章 Java 程序设计概述

1.1 重点复习

1.1.1 Java 程序设计语言的发展

Java 是美国 Sun Microsystems 公司于 1995 年 5 月正式发布的、采用面向对象技术的、能够跨平台使用的程序设计语言。

Java 2 分为标准版 (J2SE)、企业版 (J2EE) 和精简版 (J2ME) 三个版本, 分别被应用于不同的领域。随着软件版本的不断升级, 当前这三个版本分别被称为: Java SE、Java EE 和 Java ME。

1.1.2 Java 程序设计语言的特点

Java 语言具有很多优秀的特点, 使它拥有巨大的发展空间。Java 语言的主要特点如下: 易于学习、采用高效率的执行方式; 可以跨平台运行; 采用分布式应用架构; 具有较高的安全性和可靠性; 支持多线程技术以及具有丰富的 API (Application Program Interface) 类库。

1.1.3 Java 相关程序介绍

Java 语言编辑的源程序的执行必须以先经过编译器编译、再利用解释器解释的方式来运行。任何一种可以运行 (解释) Java 程序的软件都可以称为 Java 虚拟机。

由 Java 的开发工具所组成的帮助用户开发 Java 程序的环境称为 Java 的集成开发环境 (IDE)。其中最基本的是 Java 开发工具包 (Software Development Kit, SDK), 以及它的子软件 JRE。

Java 的 API 文档是 Java 程序开发的最好帮手, 可以在 Oracle 公司的网站 <http://download.oracle.com/javase/6/docs/api/> 上直接浏览查阅。

作为一个 Java 应用程序综合开发环境 (Integrated Development Environment, IDE), Eclipse 是一个功能完整且成熟的软件, 由 IBM 公司于 2001 年首次推出。大家可以在其官方网站 <http://www.eclipse.org> 免费下载该软件的打包文件, 释放到文件夹即可使用。但在其具体被使用前, 必须有 JRE 的支持。

1.1.4 Java 程序及开发过程

Java 程序都是由类 (class) 组成。Java 中类定义的语法形式为:

```
修饰符 class <类名> [extend <父类名>] {  
    type 类变量 1;  
    type 类变量 2;  
    ...  
}
```

成员变量

```

修饰符  type < 类方法名 1>( 参数列表 ){
           type 局部变量 ;
           方法体
}
修饰符  type < 类方法名 2>( 参数列表 ){
           type 局部变量 ;
           方法体
}
...
}

```

成员方法

Java 程序分为两大类型：Java Application（Java 应用程序）与应用在 WWW 上的 Java Applet（Java 小应用程序）。

Java Application 以 main() 方法作为程序的起始点，语句形式为：

```
public static void main(String args[])
```

Java Application 的开发一般须经过如下过程：

- 1) 编辑源程序。
- 2) 编译源程序。
- 3) 运行 Java 程序。

但在 Eclipse 环境下，第 2) 步往往在编辑源程序的保存过程中自动进行。

Java Applet 的开发过程与 Java Application 基本一致，但它必须搭配浏览器来运行，且它没有程序入口。

1.2 习题解析

1. Java 源程序经编译生成的字节码文件的扩展名为 _____，字节码文件由解释器执行。

- A. java B. exe C. class D. html

答案：C

解析：Java 编译器获取扩展名为“java”应用程序的源代码，把它编译成符合 Java 虚拟机规范的字节码文件，字节码文件的扩展名为“class”。

2. 在 Java 应用程序中，main() 方法的返回类型是 _____。

- A. int B. void C. boolean D. static

答案：B

解析：在 Java 应用程序中，main() 方法作为应用程序的起始点，只能用 void 返回。

3. main() 方法是 Java Application 程序执行的起始点，以下 _____ 的 main() 方法头部是合法的。

- A. public static void main()
 B. public static void main (String args [])
 C. public static int main(String [] arg)
 D. public void main(String arg[])

答案：B

解析：main 关键字后面的小括号中，是 main() 方法的参数列表，接收从外面向 main() 方法中传递的参数，其参数形式是一个字符数组，数组的名字为 args，这是 Java 编译器规定的。通常情况下，该语句还可以写成：public static void main(String [] args)。

4. 编译 Applet 源程序文件后，它所产生的字节码文件的扩展名为 _____。

A. class

B. java

C. html

D. exe

答案：A

解析：Applet 是一种被设计成可在浏览器中运行的小应用程序。Applet 与 Application 其实是类似的程序，只是程序的起点及运作原理不同。因此它的源程序经编译后所产生的字节码文件同样为 class 文件。

5. 下面选项 _____ 的 Java 源文件代码片段使用了不正确的定义次序。

A. package testpackage;

public class Test{//do something...}

B. import java.io.*;

class Person{// do something...}

public class Test{// do something...}

C. import java.io.*;

package testpackage;

public class Test{// do something...}

D. import java.io.*;

import java.awt.*;

public class Test{// do something...}

答案：C

解析：根据 Java 程序的源代码文件结构：package 语句必须放在文件开始，作用是把当前文件放入所指向的包中；其次为 import 语句和类定义语句。

6. 下面的说法错误的是 _____。

A. Java 语言是面向对象的、解释执行的网络编程语言

B. Java 语言具有可移植性，是与平台无关的编程语言

C. Java 语言编写的程序可以无须其运行环境便“一次编译，到处运行”

D. Java 语言可对内存垃圾自动收集

答案：C

解析：所谓“一次编译，到处运行”，指的是 Java 语言编写的程序经编译后得到的字节码文件可以跨平台运行，但其被编译的过程则是必须依赖其编译程序的运行环境的。

7. Java 语言具有许多特点，下列选项中，_____ 反映了 Java 程序并行机制的特点。

A. 安全性

B. 跨平台

C. 可移植

D. 多线程

答案：D

解析：所谓并行，指的是在同一时刻或者同一时间间隔内完成两种或两种以上相同或不同的工作，Java 支持的多线程技术充分反映了 Java 程序并行机制的特点。

1.3 实验指导

1.3.1 实验目的与要求

1. 掌握 Java 程序开发过程。

2. 熟悉 Eclipse 集成开发环境的使用。

1.3.2 实验样例

在 Eclipse 集成开发环境下，参照样例步骤建立和运行 Java 应用程序和小应用程序。

1. 重建工作区

首先在磁盘上（任何逻辑盘都可以，在此我们假设为 D 盘）创建一个文件夹 JavaExercise。在已经启动 Eclipse 的情况下，我们可以重建我们的工作区，操作方法是：单击“File”下拉菜单，在其中选择“Switch Workspace”其下的“Other...”项目，会弹出如图 1-1 所示对话框。

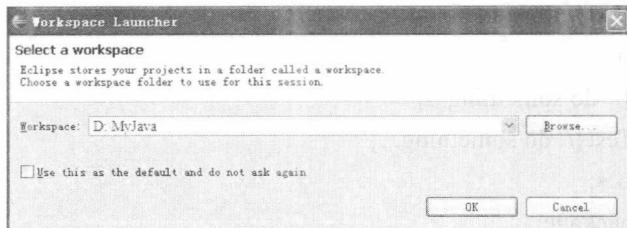


图 1-1 重建用户工作区（一）

该对话框用于帮助我们重建用户工作区，以存放我们练习中所有的 Java 程序，在“Workspace:”文本框中删除现有的内容，输入“D:\JavaExercise”，单击“OK”按钮便建立了我们自己的工作区，如图 1-2 所示。

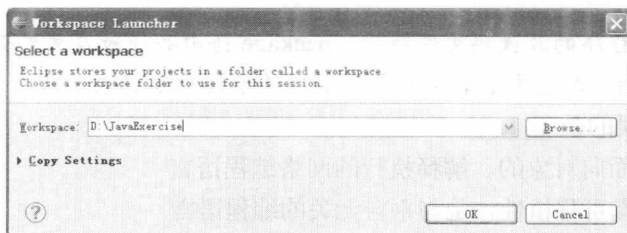


图 1-2 建立用户工作区（二）

Eclipse 在重建工作区的过程中，将自动关闭并且重启当前的系统，直至出现 Eclipse 的欢迎界面。

2. 创建项目

创建项目的步骤是：右键单击 Package Explorer 视图，在弹出菜单中选择“New”，在其子菜单项中选择“Java Project”，在提示项目名称时输入“ch1”（表示存放的是与第 1 章相关的程序），此时可以看到：程序文件的默认保存位置就是所建立的文件夹的地址；此时项目就创建在你自己的用户文件夹中（见图 1-3），然后单击“Finish”按钮。

3. 开发 Java 应用程序

现在我们将创建并运行一个输出两行字符串：“Hello friend:”以及“Welcome to Java!”的 Java 应用程序。使用 Java 的 Package Explorer 视图，右键单击“ch1”项目，选择 New -> Class，在随后出现的对话框（见图 1-4）中，键入“HelloApp”作为类名称。在“Which method stubs would you like to create?”下面，选中“public static void main(String[] args)”复选框，去掉“Inherited abstract methods”复选项，然后单击“Finish”按钮。

这样将在编辑器区域创建一个包含 HelloApp 类和空的主方法（main() 方法）的 .java 文件，

然后向该方法添加如下代码：

```
System.out.print("Hello friend:");  
System.out.println("Welcome to Java!");
```

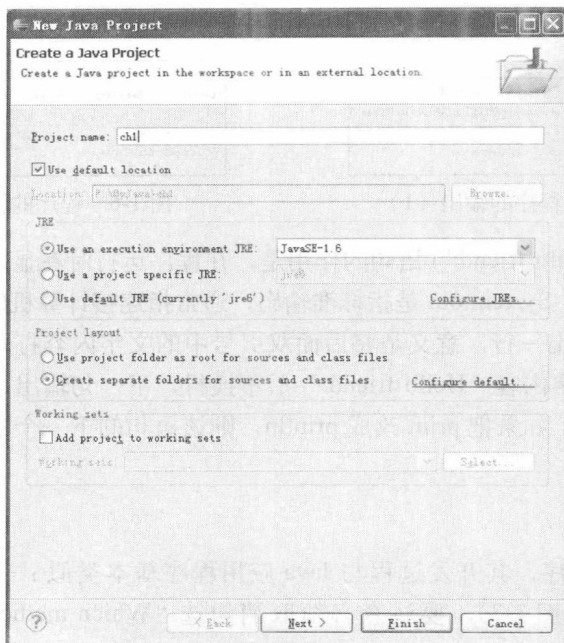


图 1-3 创建 Java 项目



图 1-4 在 Java 透视图图中创建新类

假设程序没有出错的部分，你就能够从 Eclipse 菜单栏选择“Run”来执行该程序（注意：这里不存在单独的编译步骤，因为编译是在你保存代码时进行的）。运行 Java 应用程序的方法是：单击“Run”下拉菜单，在其中选择“Run as”，其下的“Java Application”项目，一个新的选项卡式窗格“Console”将出现在下面的窗格（控制台）中，其中显示了程序的输出，如图 1-5 所示。

现在请将主方法中的语句改变成如下的形式：

```
System.out.println("Hello friend:");
System.out.println("Welcome to Java!");
```

再次运行程序，程序的运行结果如图 1-6 所示。

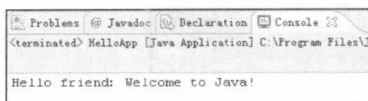


图 1-5 程序的输出 (1)

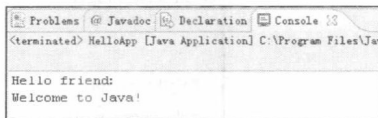


图 1-6 程序的输出 (2)

`System.out.print("Hello friend:");` 语句的作用是，在程序运行时在显示器上输出双引号内的文字“Hello friend:”。其中 `System.out` 是指标准输出，通常指连接计算机的设备，如打印机、显示器等。`print` 的含义是输出一行，意义是将后面双引号中的文字内容打印在标准输出设备——显示器上，并且在输出文字内容“Hello friend:”后不换行，下一句输出语句的输出内容将紧接在该语句输出内容的后面。如果把 `print` 改成 `println`，则该语句的下一个语句的输出将在换行以后进行。

4. 开发 Java Applet

对于 Java 小应用程序，其开发过程与 Java 应用程序基本类似：一开始直接在 `ch1` 中创建类，创建类的时候（见图 1-7），要注意必须取消勾选“Which method stubs would you like to create?”问题下面的所有复选框。在类名框中输入类的名字为 `JaApplet`，单击“Finish”按钮。



图 1-7 在 Java 透视图中创建 Applet 新类

在程序编辑窗口输入如下程序源码，如图 1-8 所示。

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

```
g.drawString("Hello friend: ",60,20);  
g.drawString("Welcome to Java!",50,40);  
}  
}
```

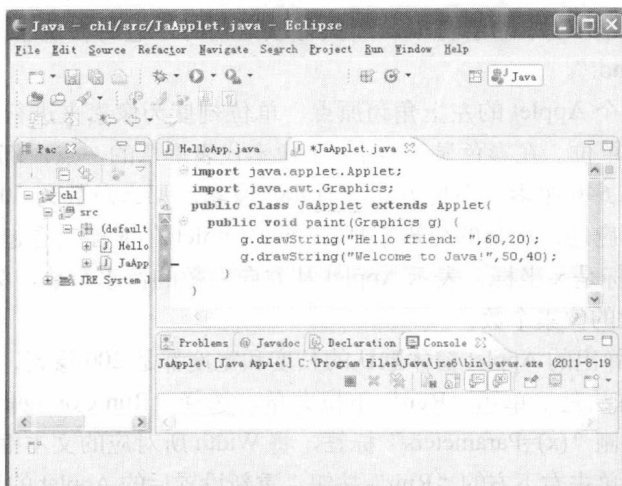


图 1-8 Applet 程序编辑窗口

输入完毕后保存文件，如果程序没有错误就运行程序。运行 Java Applet 程序的方法是：在菜单栏上单击“Run”项目，在弹出的下拉菜单中选择“Run As”菜单项，然后在紧接着出现的子菜单上选择“Java Applet”项目，如图 1-9 所示。

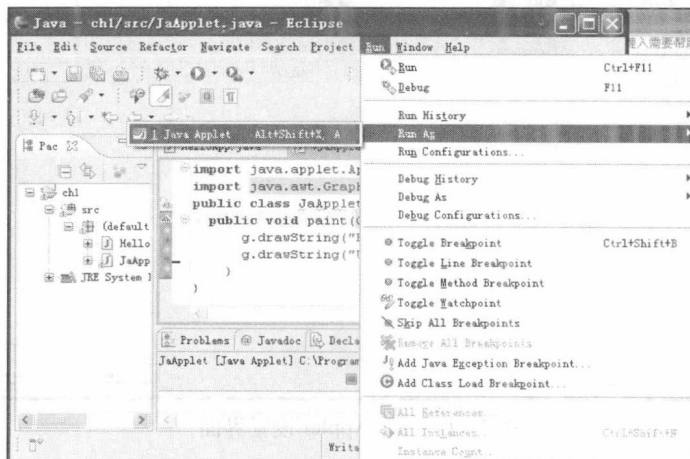


图 1-9 运行 Java Applet

Java Applet 程序运行结果如图 1-10 所示。

当我们关闭小应用程序查看器窗口后，程序运行完毕。

对于 Java Applet 来说，所有的 Java Applet 必须从 Applet 类中继承而来，在继承关系中，Applet 称为超类，JaApplet 称为子类，我们将在第 4 章为大家详细地介绍继承关系。

语句 `public void paint(Graphics g)` 引入了一个方法（关于方法的介绍将在第 3 章以及以后的章节中进行），当一个 Applet 执行时，将自动调用 `paint()` 方法，从而在屏幕上输出信息。`paint` 之后



图 1-10 Java Applet 运行结果

的左括号定义了这个方法的参数表，方法从此参数表中接收完成任务所需的信息，方法 `paint()` 的参数表说明它需要一个 `Graphics` 的对象（在此声明为 `g`）来完成相应的任务。

语句 `g.drawString("Hello friend:",60,20)` 指示计算机执行一个动作，即采用 `Graphics` 对象 `g` 的 `drawString()` 方法画出引号中的字符串。当执行该语句时，它将在屏幕坐标的（60,20）点开始显示信息“Hello friend:”。

坐标是以屏幕上这个 Applet 的左上角为原点、单位刻度为像素来进行计算的，像素是在屏幕上显示输出时的基本单位。在彩色显示器上，1 像素代表屏幕的一点。例如，许多个人计算机在屏幕的宽度方向上有 640 像素，高度方向上有 480 像素，那么总数为 307200（640×480）个可显示的图形元素。对应地，屏幕的分辨率越高 Java Applet 在屏幕上看起来就越小。在坐标点（60,20）中的第一个坐标是 *x* 坐标，表示 Applet 从左向右数的像素个数，第二个坐标是 *y* 坐标，表示 Applet 从上向下数的像素个数。

Java Applet 运行时弹出的 Applet 窗体默认的长度和宽度都是 200 像素，若有特殊需要则可以进行手动设置，设置方法是：单击“Run”下拉菜单，选择“Run Configurations...”选项，进入运行配置界面，单击右侧“(x)=Parameters”标签，将 Width 所对应的文本框中的数据 200 修改为 400（如图 1-11 所示），单击右下方的“Run”按钮，重新设置后的 Applet 的显示如图 1-12 所示。

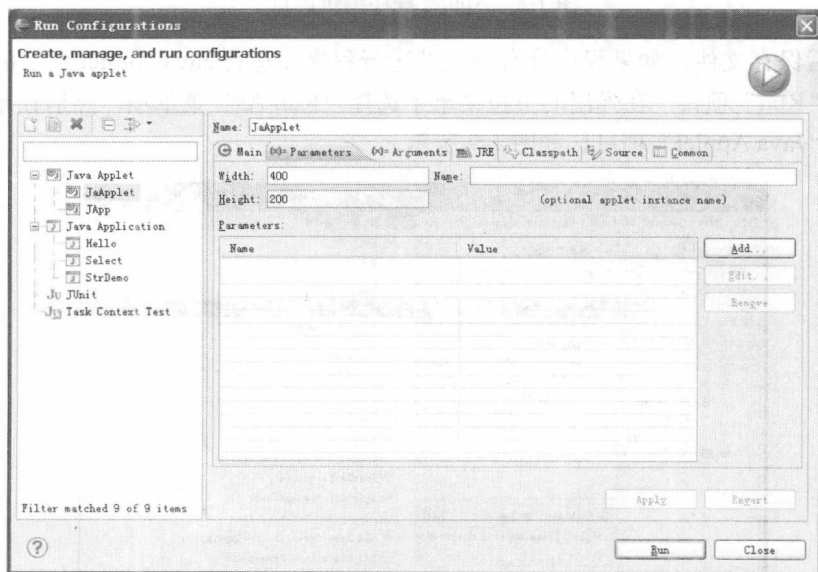


图 1-11 Java Applet 设置界面



图 1-12 重新设置后的 Java Applet 运行结果

1.3.3 实验内容

1. 试修改例 1.4.1 程序, 使其输出的字符串为 “I’d like to study Java !”, 并在 Eclipse 环境下编译与运行该程序。

2. 编辑 Applet 程序, 使运行后在 Applet 窗口输出如图 1-13 所示内容。

1.4 练习题

1. Java 源程序必须以 _____ 作为其扩展名。

- A. java
- B. exe
- C. class
- D. html

2. 以下 _____ 正确表述了 Java 语言面向对象特性。

- A. Java 程序设计语言提供了丰富的 API 文档
- B. Java 程序设计语言将类作为部分特性和行为事务的抽象
- C. Java 程序设计语言的运行与平台无关
- D. Java 程序设计语言具有封装、继承和多态的特性

3. 使用 Eclipse 开发与运行 Java 程序, 下列 _____ 不是其开发的主要步骤之一。

- A. 编辑源程序
- B. 调试源程序
- C. 编译源程序
- D. 运行程序

4. 下面关于 Applet 的说法正确的是 _____。

- A. Applet 程序不需要编译
- B. Applet 程序每次只能被一个用户下载并在客户端运行
- C. Applet 程序必须继承自 java.awt.Applet
- D. Applet 程序中需要 main() 方法

5. Eclipse 是 _____。

- A. 一种全新的程序设计语言
- B. 一种程序开发软件
- C. 一种由 Java 写成的, 并支持 Java Applet 的浏览器
- D. 一种游戏软件

6. 下面 _____ 不是 Java 的访问权限。

- A. public
- B. private
- C. protected
- D. static

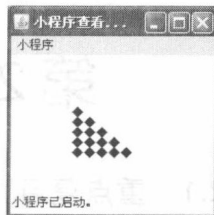


图 1-13 运行结果

第2章 Java 程序设计的基本概念

2.1 重点复习

2.1.1 数据类型及标识符

Java 的数据类型可分为两大类：基本数据类型和复合数据类型。

基本数据类型共有 8 种，分为 4 小类，分别是：布尔型（boolean）、字符型（char）、整型（byte、short、int、long）和实型（float、double）。

复合数据类型包括数组、类（class）和接口（interface）。

Java 的标识符是用于定义各种对象名称的字符串集合。标识符一般分为用户自定义标识符和系统标识符两种。系统标识符即关键字，而用户自定义标识符必须遵循如下的命名规则：

- 以字母、下划线“_”或“\$”作为开头字符，数字不能作为标识符的开头字符。
- 标识符中间不能使用空格。
- 不能使用 Java 的操作符。
- 不能使用 Java 的关键字。
- 严格区分大小写。

2.1.2 常量的定义与使用

根据 Java 的基本数据类型的分类方式，Java 中的常量同样分为整型常量、实型常量、字符型和字符串常量以及布尔型常量。其中，字符型常量中的转义字符用以表示类似回车符之类的难以用一般字符来表示的字符。

定义常量的形式如下：

```
final type <常量名>=<常量值> [, <常量名>[=<常量值>]]
```

2.1.3 变量的定义与使用

Java 规定：变量在使用之前，必须先定义（也称为声明）。

变量定义形式如下：

```
type <变量名> [= <变量初值>] [, <变量名> [= <变量初值>]];
```

变量一旦经过定义便可进行赋值，变量的赋值方式一般有两种形式：在定义的同时进行赋值；在程序体中进行赋值。

变量的作用域指可以访问该变量的程序代码的范围。按作用域来分，变量可分为局部变量、成员变量、方法参数和异常处理参数。

2.1.4 运算符与表达式的使用

运算符指明对操作数所进行的运算。