

高等学校计算机程序设计解析法系列教材

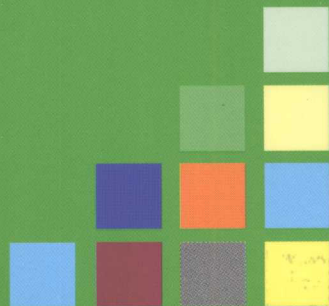
Java

程序设计习题解析 与实验教程

张璞 甘玲 李盘林 等 编著



清华大学出版社



高等学校计算机程序设计解析法系列教材

Java

程序设计习题解析 与实验教程

清华大学出版社
北京

内 容 简 介

本书是《解析 Java 程序设计》的配套习题解答与实验指导。全书共分 10 章,每章分两部分,第 1 部分是习题解析,给出了主教材中每一章的习题参考解答;第 2 部分是实验指导,实验内容与主教材的各章相对应,每个实验由实验目的、问题描述、实验思路、关键步骤等组成。

本书不仅可以结合《解析 Java 程序设计》使用,单独使用也会有很好的效果。本书内容丰富,实用性强,可作为高等院校计算机专业或非计算机专业、各类成人教育学院 Java 程序设计课程的教学辅导书,也可供 Java 编程爱好者参考使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Java 程序设计习题解析与实验教程/张璞等编著. —北京:清华大学出版社,2010.4
(高等学校计算机程序设计解析法系列教材)

ISBN 978-7-302-21510-3

I. ①J… II. ①张… III. ①Java 语言—程序设计—高等学校—教学参考资料
IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 033108 号

责任编辑:付弘宇

责任校对:梁 毅

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京市昌平环球印刷厂

装 订 者:北京国马印刷厂

经 销:全国新华书店

开 本:185×260 印 张:15.25 字 数:373 千字

版 次:2010 年 4 月第 1 版 印 次:2010 年 4 月第 1 次印刷

印 数:1~3000

定 价:22.00 元

产品编号:029427-01

高等学校计算机程序设计解析法系列教材

编审委员会

名誉主任：冯博琴

主 任：邱玉辉 陈流汀

副 主 任：王国胤 卢先和 郑 莉

委 员：（按音序排列）

安世全 陈 维 丁 岭 杜茂康

甘 玲 洪汝渝 刘 玲 刘达明

孟民果 邱 劲 石 岩 唐 雁

谢惠娟 应 宏 张 虹 张 璞

曾 一 周建丽 曾庆森 邹显春

丛 书 序

长久以来,一直有一个缺憾存在于计算机语言教育界:课堂讲授和与之配套的教材过多地阐述细节性的知识点,而对计算机语言所要达到的最终目的——提高编程能力本身注重不够,导致学生考试可以得高分,但解决实际问题的能力却不强。究其原因是多方面的,仅就教材而言,市面上可供读者选择的优秀著作如此之多,其中大部分出自业界的泰山北斗,我们也欣喜地从这些著作中感受到学术的变迁,欣赏到学者的风采。然而遗憾的是,这些著作的章节安排和行文叙述更适合研讨和参考,却不容易走进课堂;国内的程序设计教材注重语法讲解,条理清楚,便于总结性的教学和自学基础知识,但又缺少知识的发现过程和对问题来龙去脉的分析,更缺少解决实际问题的完整分析和解决方案,而这正是国外同类书籍最突出的闪光点,也是教学的真正意义所在。因此,我们希望能将国内外教材的特色结合起来,形成更加适合课堂教学的优秀教材。

出版这套“解析法”系列教材的目的就是为了弥补这一缺憾,对计算机语言传统的讲授方式进行改革。“解析法”的思路是从实际问题入手,剖析求解问题的关键点(进行知识的发现),然后结合问题讲解需要的知识点,最后给出问题的求解办法和实现过程,并举一反三。

读者可以从本套教材中看到解析法贯穿于始终,每章开始都提出一个问题,并对问题进行分析,给出解决问题的思路和方法,接下去根据分析引出相应的知识点,层层推进,最后给出问题解决的具体实现。这样更符合人们的思维方式以及程序设计课程的特点。更重要的是,解析的目的不仅在于讲授知识,更注重培养学习者独立发现问题、分析问题、解决问题的能力 and 进行知识迁移的能力,而这些能力又是每个软件开发人员应该具备的。

此外,本系列的每本教材中包含了具有实际意义的丰富例题、思考题和习题,具有较强的启发性,便于教师和学生的教与学。本系列教材对于高等院校计算机专业以及相关专业的程序设计教学工作有着重要作用。我们相信,本系列教材的出版会进一步推动我国教育教学改革特别是计算机基础教学的改革。

高等学校计算机程序设计解析法系列教材

编审委员会

2006年6月初稿

2007年11月修改

前 言

Java 语言是一种应用广泛的面向对象编程语言,它功能强大又简单易学,深受广大程序设计人员的喜爱。当前,许多高等院校都开设了和 Java 语言相关的程序设计课程。

本书编者已编写出版《解析 Java 程序设计》(清华大学出版社,ISBN9787302136217),为使选择该书作为教材的读者能更快、更方便、更准确地学习 Java 语言,作者特别编写了这本《Java 程序设计习题解析与实验教程》,作为教材的配套参考书。

本书对教材中各章的主要习题进行了解答,对编程题目给出了必要的分析和说明、源代码以及程序运行结果。此外,实验环节是 Java 程序设计课程的重要内容,也是提高教学质量的关键环节。学生一方面需要在课后通过做练习题来帮助理解相关基础知识,另一方面也需要通过上机实践,更好地掌握 Java 语言编程的方法,使自己的程序设计能力不断提高。为方便读者进行 Java 程序上机实验,本书每章均提供了 3~6 个实验供读者选做,每个实验由实验目的、问题描述、实验思路、关键步骤等组成。

需要说明的是,书中很多实验的关键步骤中,有一部分代码是作者有意识地做了省略,其目的是希望读者在进行上机实验时能在关键步骤的引导下,发挥主观能动性,自己编写代码将实验完成,而不仅仅是将教材上的代码加以简单的验证。这些实验试图使读者在掌握 Java 语言的基础上,对现实世界中的一些较简单的实际问题用 Java 语言进行描述,从而在培养读者基本实验技能的同时,增强理论联系实践的能力。

为了启发读者的思路,我们在一些习题解答和实验指导之后加入了相应的问题拓展部分,便于读者更深入地进行学习思考。

这里需要提醒读者,本书中很多习题和实验题目的答案和代码都不是唯一的,特别是对于程序代码,实现的方式更可以多种多样,希望读者在学习本书的过程中,能够举一反三,尝试编写出更简练,功能更完善,效率更高,可读性更好的代码。

本书所有参考代码都基于 J2SDK 1.5.0 平台,在 Eclipse 环境下调试通过。参考代码按照《解析 Java 程序设计》一书的章节顺序进行组织,除了习题中已指定名称的类以外,习题解答中的代码文件名称以 Test 开始,例如 Test2_2.java 代表第 2 章习题 2-2 的参考代码;实验指导部分的代码名称以 Exp 开始,例如 Exp7_1.java 代表第 7 章实验 7-1 的参考代码。

本书的源代码可以从清华大学出版社网站(www.tup.com.cn)下载。在实际运行参考代码时,需要在 JDK 1.5 下重新编译。如果程序要求图像、声音或者文本文件作为输入,则需要读者自行将相同类型的文件复制到相应目录下,方可正常运行。在运行 Applet 类的代码时,请读者自行编辑下面 HTML 代码中类的名称,与 Applet 类的名称相对应。

```
<html>  
<Applet code=MyApplet.class width=300 height=100>  
</Applet>  
</html>
```

本书由张璞主编,甘玲、李盘林、唐晓军、王利、段小林参编。其中,张璞编写了第2、4、7、10章的习题解析部分以及第4章的实验指导部分,甘玲编写了第1章,李盘林编写了第6、8章,唐晓军编写了第3、9章,王利编写了第5章,段小林编写了第2、7、10章的实验指导部分。全书由张璞统稿和校稿。同时,在此要感谢本书所列参考文献的作者。

由于作者水平有限,加之时间仓促,错误与疏漏之处在所难免,恳请批评指正。作者联系方式:zhangpu@cqupt.edu.cn。

本书的代码下载如果有任何问题,请联系:fuhy@tup.tsinghua.edu.cn。

编 者

2009年12月

目 录

第 1 章 面向对象技术概述与编程环境.....	1
1.1 习题解析	1
1.2 实验指导	2
第 2 章 Java 语言基础	7
2.1 习题解析	7
2.2 实验指导.....	20
第 3 章 对象与类	27
3.1 习题解析.....	27
3.2 实验指导.....	46
第 4 章 继承和多态	53
4.1 习题解析.....	53
4.2 实验指导.....	68
第 5 章 输入输出流和异常	75
5.1 习题解析.....	75
5.2 实验指导.....	94
第 6 章 图形用户界面设计.....	104
6.1 习题解析	104
6.2 实验指导	134
第 7 章 网络编程.....	141
7.1 习题解析	141
7.2 实验指导	161
第 8 章 Applet	176
8.1 习题解析	176
8.2 实验指导	190

第 9 章 JDBC 与数据库	196
9.1 习题解析	196
9.2 实验指导	208
第 10 章 常用数据结构的实现	215
10.1 习题解析	215
10.2 实验指导	223
参考文献	230

第 1 章 面向对象技术概述与编程环境

1.1 习题解析

习题 1-1 简述面向对象技术的基本特征。

【问题解答】

面向对象技术的基本特征主要有封装、继承和多态。

封装是指把对象的属性和服务结合成一个独立的系统单位,并尽可能隐蔽对象的内部细节,只是向外部提供接口。

继承是指特殊类拥有其一般类的全部属性与服务,称作特殊类对一般类的继承。继承简化了人们对现实世界的认识和描述,在定义子类时不必重复定义那些已在父类中定义过的属性和服务,只要指定它的父类并定义新的属性和服务即可。

多态是指同一个命名可具有不同的语义。面向对象方法中,常指在一般类中定义的服务被特殊类继承之后,可以表现出不同的行为。

习题 1-2 解释以下基本概念:对象、类、消息。

【问题解答】

对象(object)是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,由一组属性和对这组属性进行操作的一组方法组成。属性是用来描述对象静态特征(即状态)的一个数据项;方法是用来描述对象动态特征(即行为)的一个操作序列。

类(class)是具有相同属性和操作的对象集合的抽象。对象是类的实例。

消息(message)是对象之间相互请求或相互协作的途径,是一个对象要求某个对象执行某个操作的规格说明。

习题 1-3 面向对象程序设计 with 面向过程程序设计的区别。

【问题解答】

面向对象程序设计与面向过程程序设计两者相比较而言,主要有以下不同:面向对象程序设计以数据为中心描述系统,而面向过程程序设计是以功能为中心来描述系统。面向对象程序设计将数据和对数据的操作封装在一起,作为一个整体来处理,而面向过程程序设计采用函数(或过程)来描述对数据的操作,但又将函数与其操作的数据分离开来。面向对象程序设计提供了面向过程程序设计所不具备的特性,如继承和多态。

习题 1-4 阐述面向对象方法的基本思想。

【问题解答】

面向对象方法的基本思想是:从现实世界客观存在的事物出发来建立软件系统,强调直接以问题域(现实世界)中的事物为中心来思考问题、认识问题,并根据这些事物的本

质特征,把它们抽象地表示为系统中的对象,作为系统的基本构成单位。这可以使系统直接映射问题域,保持问题域中事物及其相互关系的本来面貌。

习题 1-5 简述 UML 语言的基本组成部分。

【问题解答】

UML 是为开发软件系统而进行规格化(specifying)、可视化(visualizing)、构造化(constructing)、文档化(documenting)的一种通用建模语言,其主要任务是在面向对象的软件开发中进行系统分析和设计的可视化建模。作为一种建模语言,UML 的定义包括 UML 语义和 UML 表示法两个部分。

UML 语义:描述基于 UML 的精确元模型定义。元模型为 UML 的所有元素在语法和语义上提供了简单、一致、通用的定义性说明,使开发者能在语义上保持一致,消除因人而异的最佳表达方法所造成的影响。此外,UML 还支持对元模型的扩展定义。

UML 表示法:定义 UML 符号的表示法,为开发者或开发工具使用这些图形符号和文本语法进行系统建模提供了标准。

1.2 实验指导

实验 1-1 JDK 的安装与配置。

【实验目的】

掌握安装 Java SDK 软件包的方法,掌握设置 Java 运行环境的方法,为后续 Java 程序的实验做准备。

【问题描述】

在网络上下载 JDK 软件,进行安装。设置环境变量,并验证 JDK 环境是否安装成功。

【关键步骤】

(1) 下载 JDK

为了建立 Java 运行环境,需要先下载 Sun 公司的免费 J2SE 安装程序。在浏览器中输入“<http://java.sun.com>”,进入 Sun 公司官方网站。在页面右边 Popular Downloads 栏中选择 Java SE 可进入下载页面。

(2) 安装 JDK

运行下载的软件包,在安装过程中需要设置安装路径及选择安装组件,默认的组件选择是全部安装。安装成功后,安装目录下的 bin 文件夹中包含编译器(javac.exe)、解释器(java.exe)、Applet 查看器(appletviewer.exe)等可执行文件,lib 文件夹中包含了所有的类库以便开发 Java 程序时使用,demo 文件夹中包含相应的程序实例。

(3) 设置环境变量

JDK 中的工具大多为命令行工具,需要从命令行运行它们。为了便于使用这些工具,通常需要设置相应环境变量。下面以 Windows XP 为例说明设置环境变量的过程。

选择桌面上“我的电脑”图标,再右击鼠标,出现“系统属性”窗口,在“高级”选项卡中单击“环境变量”按钮,将出现“环境变量”设置界面,如图 1-1 所示。



图 1-1 “环境变量”设置窗口

在“系统变量”框中单击“新建”按钮,出现“新建系统变量”对话框,在“变量名”一栏的文本框内输入“JavaHome”,在变量值一栏的文本框内输入“d:\java\jdk1.5”(即 JDK 安装的主目录,读者可根据自己指定的目录相应更改),如图 1-2 所示。

单击“确定”按钮,返回图 1-1。在“系统变量”框中选择 path,然后单击“编辑”按钮,出现“编辑系统变量”对话框,在“变量值”一栏的命令后添加“;%JavaHome%\bin”,如图 1-3 所示。

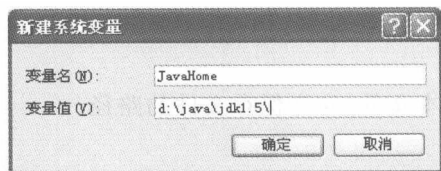


图 1-2 “新建系统变量”对话框



图 1-3 “编辑系统变量”对话框

在图 1-1 的“系统变量”框中单击“新建”按钮,弹出“新建系统变量”对话框,在“变量名”栏中输入“classpath”,在“变量值”一栏输入“.; %JavaHome%\lib”,其中“.”表示当前目录。

(4) 设置完成后,选择开始菜单的“开始”|“运行”,输入“cmd”命令,进入 DOS 窗口。在命令行提示符后输入“java”或“javac”并回车,如果出现其用法参数提示信息,则安装正确,如图 1-4 所示。

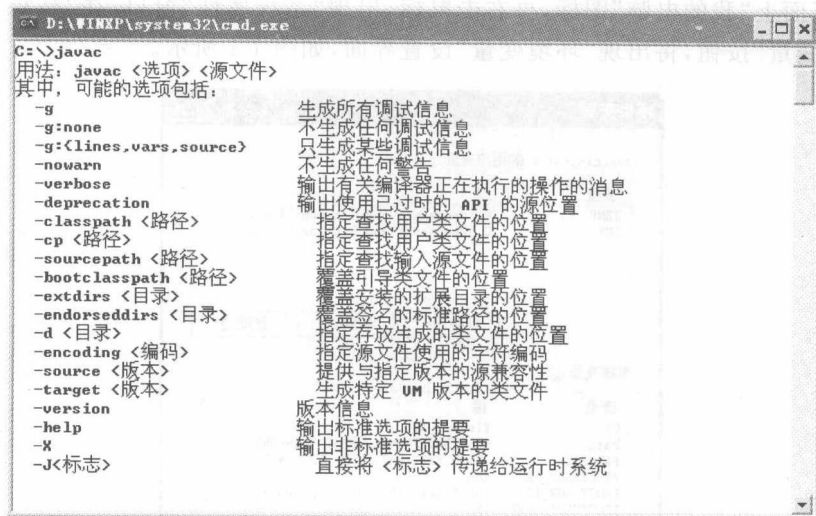


图 1-4 JDK 参数提示信息

实验 1-2 一个简单的应用程序的上机过程。

【实验目的】

掌握在 JDK 环境下编译及运行 Java 程序的基本步骤,为后续实验做准备。

【问题描述】

使用文本编辑器编写一个简单的 Java 程序,并使用 JDK 环境编译、运行它。

【关键步骤】

(1) 打开某个文本编辑器软件(如“记事本”程序),输入如下代码:

```
public class HelloJava {
    public static void main(String[] args) {
        System.out.println("Hello Java!");
    }
}
```

(2) 将文件命名为 HelloJava.java,保存为文本文件,注意保存文件的路径。

(3) 使用 javac 命令编译源程序。

打开命令提示符窗口,进入 Java 源程序所在路径,输入“javac HelloJava.java”并按 Enter 键,进行编译。

(4) 使用 java 命令运行程序。

使用 Java 解释器(java.exe)将编译得到的字节码文件 HelloJava.class 解释执行。在命令提示符窗口输入“java HelloJava”,按 Enter 键即开始解释执行,并可看到运行结果,如图 1-5 所示。需要注意的是,在解释执行代码时,不要用

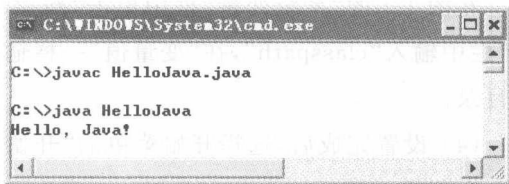


图 1-5 实验 1-2 程序的编译及运行结果

“java HelloJava.class”命令。

实验 1-3 常用 IDE 开发工具 Eclipse 的使用。

【实验目的】

掌握使用 Eclipse 开发工具来编写及运行 Java 程序。

【问题描述】

使用 Eclipse 开发工具编写一个简单的 Java 程序,并在集成环境中编译运行该 Java 程序。

【关键步骤】

(1) Eclipse 的下载与安装。

使用 Eclipse 前必须先安装好 JDK。Eclipse 是个免费的软件,可以从 Eclipse 的官方网站 <http://www.eclipse.org> 下载。

下载 Windows 平台的 Eclipse 软件后(以 Eclipse3.3.2 版本为例),得到一个压缩包文件 eclipse-SDK-3.3.2-win32.zip,将下载的压缩包解压到某个磁盘,即可完成 Eclipse 的安装。

(2) 双击 eclipse 文件夹下的 eclipse.exe 执行文件,启动 Eclipse。

(3) 启动 Eclipse 以后,选择菜单“文件”|“新建”|“项目”,出现“新建项目”窗口,从窗口的树形结构中选择“Java 项目”,如图 1-6 所示。

(4) 单击“下一步”按钮,将创建的项目命名为“HelloJava”,确保选中内容一栏的“在工作空间中创建新项目”,这样该项目就被创建在当前的工作空间中。若不需要定义该项目的构建设置,即改变该项目的输出路径、添加或编辑与该项目相关的包或库,则直接单击“完成”按钮,如图 1-7 所示。

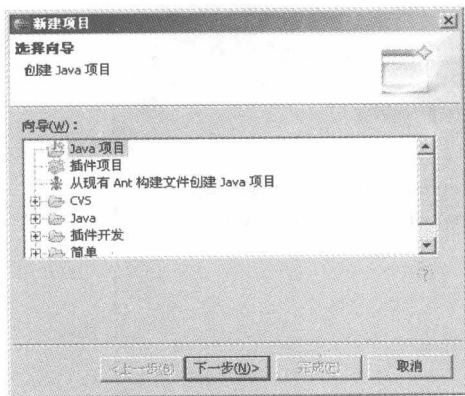


图 1-6 选择所要创建的项目类型

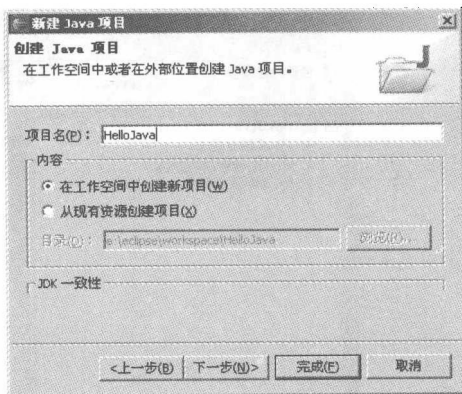


图 1-7 对 HelloJava 项目进行创建配置

(5) 在包资源管理器中选中 HelloJava 项目并右击鼠标,在弹出的菜单中选择“新建”|“类”,然后在“新建 Java 类”窗口中按照图 1-8 进行设置。

(6) Eclipse 完成类的创建后,工作台右边的窗体中会给出 HelloJava.java 的程序框架,在 main()方法中添加如下一行代码,如图 1-9 所示。

```
System.out.println("Hello Java!");
```

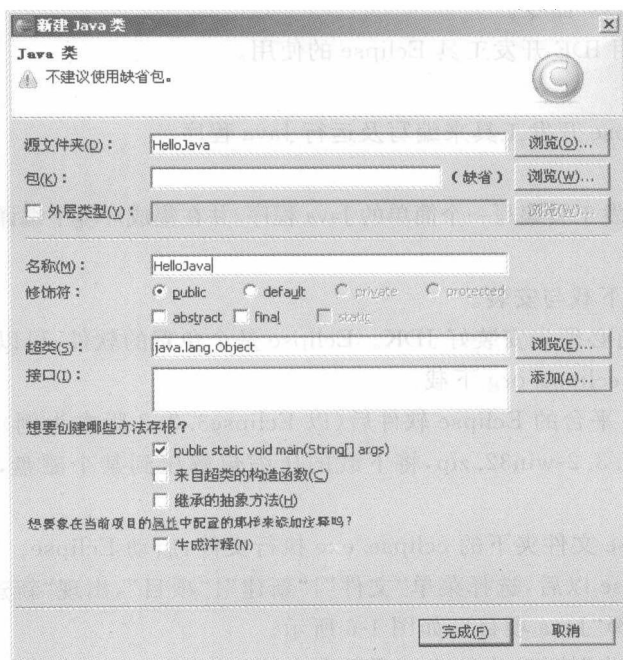


图 1-8 创建 HelloJava 类

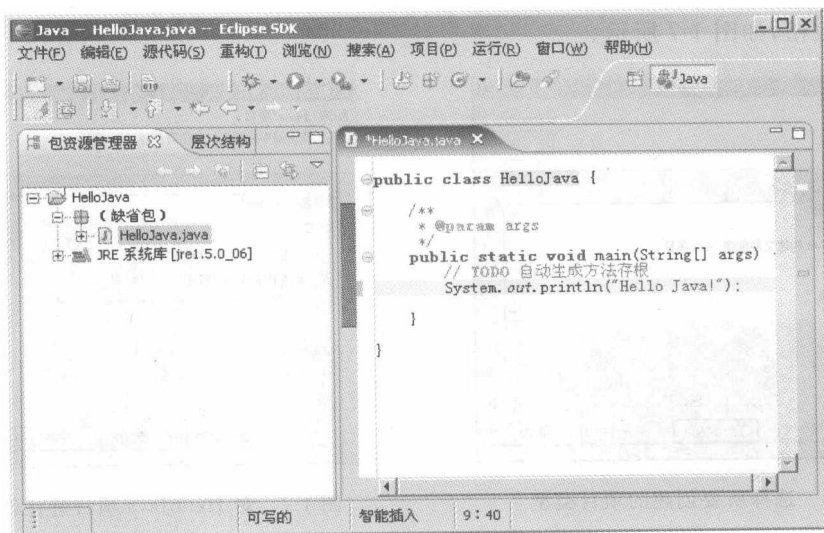


图 1-9 编辑 HelloJava 源程序

(7) 编辑好源程序之后,选择菜单“运行”|“运行方式”|“Java 应用程序”,即可得到程序运行结果。

第 2 章 Java 语言基础

2.1 习题解析

习题 2-1 编写一个 Java 应用程序,在屏幕上显示“我会编写 Java 程序啦!”。

【问题解答】

每个 Java 程序至少需要定义一个类。本题中,定义一个类,在类的 main 方法中调用系统预定义类 System 中的 out 成员的 println()方法,在屏幕上输出字符串。

【程序代码】

```
public class Test2_1{
    public static void main(String[] args) {
        System.out.println("我会编写 Java 程序啦!");
    }
}
```

【运行结果】

我会编写 Java 程序啦!

习题 2-2 对一个四位数 x ,编程计算 x 的每一位数字相加之和(忽略整数前的正负号)。例如, x 为 1234,则由 1234 分离出其千位 1、百位 2、十位 3、个位 4,然后计算 $1+2+3+4=10$,于是输出 10。

【问题解答】

要分离一个四位数的各位数字,主要考虑运用相关的算术运算符来完成功能。

【程序代码】

```
import java.io.*;
import java.util.*;
public class Test2_2{
    public static void main(String args[]){
        int n,a,b,c,d,m;
        System.out.println("请输入 n:");
        Scanner sc=new Scanner(System.in); // 构造对象来读取控制台窗口的输入
        n=sc.nextInt(); // 从键盘输入整数,并将其保存在 n 变量中
        a=n/1000; // 求千位
        b=n%1000/100; // 求百位
        c=n%1000%100/10; // 求十位
        d=n%10; // 求个位
    }
}
```

```

        m=a+b+c+d;
        System.out.println("m="+m);
    }
}

```

【运行结果】

请输入 n:

1234

m=10

说明：上述结果中“1234”的下划线表示它是用户的输入，下划线并不真正显示在运行窗口中。

习题 2-3 编写一个程序，产生三个随机整数，并把其中的最小数输出（提示：产生随机整数可以用 java.util 包中的 Random 类）。

【问题解答】

在 java.util 包中提供一个 Random 类，可以新建一个 Random 的对象来产生包括 int、float、double、long 等各种类型的随机数。本例中，设所产生的随机整数的值的范围在 0~100 之间，可以调用 Random 类的 nextInt(int n) 方法来实现，下面是该方法的说明：

```
int nextInt(int n)
```

该方法返回一个伪随机数，它是从此随机数生成器的序列中取出的、在 0(包括)和参数指定值(不包括)之间均匀分布的一个整型值。

【程序代码】

```

import java.util.*;

public class Test2_3 {
    public static void main(String args[]) {
        int a, x, y, z;
        Random r = new Random();
        x = r.nextInt(100);
        y = r.nextInt(100);
        z = r.nextInt(100);
        System.out.println("x=" + x + "   y=" + y + "   z=" + z);
        if (x > y)
            a = y;
        else
            a = x;
        if (a > z)
            System.out.println("最小数为: " + z);
        else
            System.out.println("最小数为: " + a);
    }
}

```