



高等学校精品规划教材

程序设计课程系列

主 编 陈语林

副主编 刘胤宏 王新英 杜 伟

Java 程序设计简明教程 实验实训与习题选解



中国水利水电出版社

www.waterpub.com.cn

21 世纪高等学校精品规划教材

Java 程序设计简明教程

实验实训与习题选解

主 编 陈语林

副主编 刘胤宏 王新英 杜 伟

内 容 提 要

本书是《Java 程序设计简明教程》(中国水利水电出版社出版,陈语林主编)一书的同步配套教材。根据教育部高等学校计算机科学与技术教学指导委员会非计算机专业计算机基础课程教学指导分委员会《关于进一步加强高等学校计算机基础教学的意见》,为方便广大师生的教与学,在本书的第一部分提供《Java 程序设计简明教程》第一、二层次的实验指导,共设计了 16 个教学实验;第二部分主要针对第三层次的实验教学,设计了三个比较大的综合实例,并对这些实例的分析和设计过程进行了详细的指导,供课程教学后的全日制综合实训选用;考虑既要有教学辅导意义,又要给学生留有适当的余地,第三部分对《Java 程序设计简明教程》的部分习题进行了解答;第四部分是“Java 语言程序设计”课程的模拟试题及解答;为满足部分学生参加全国计算机等级考试的需要,第五部分提供了全国计算机等级考试 Java 语言考试大纲,以及全真的笔试、机试试题及解答。

本书既可以作为高等学校“Java 语言程序设计”课程的辅导教材,又能给 Java 语言初学者提供一定的帮助。

图书在版编目(CIP)数据

Java 程序设计简明教程实验实训与习题选解 / 陈语林

主编. —北京:中国水利水电出版社, 2009

21 世纪高等学校精品规划教材

ISBN 978-7-5084-6669-9

I. J… II. 陈… 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 (2009) 第 124401 号

策划编辑: 雷顺加 责任编辑: 李 炎 加工编辑: 周益丹 封面设计: 李 佳

书 名	21 世纪高等学校精品规划教材 Java 程序设计简明教程实验实训与习题选解
作 者	主 编 陈语林 副主编 刘胤宏 王新英 杜 伟
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京泽宇印刷有限公司
规 格	184mm×260mm 16 开本 13.25 印张 329 千字
版 次	2009 年 7 月第 1 版 2009 年 7 月第 1 次印刷
印 数	0001—4000 册
定 价	22.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

前 言

为了使选择《Java 程序设计简明教程》作为教材的教师和学生能更快、更方便、更准确地学习 Java 语言,以及更熟练掌握 Java 的编程技巧和方法,特编写本书,作为教材的配套参考书使用。

虽然编写本书最初的出发点是作为《Java 程序设计简明教程》的配套教材,但实际上本书具有较好的独立性。除第三部分选择性地解答了教材的部分习题外,本书的其他部分均可以单独使用。因此,本书既可以作为高等学校“Java 语言程序设计”课程的辅导教材,又能给 Java 语言初学者提供一定的帮助。

学习程序设计语言最重要的落脚点就是要用这种语言来解决工作中的一些实际问题,学习 Java 语言也不例外。因此,学习“Java 语言程序设计”的一个重要环节就是要亲自动手做实验,通过实验培养自身分析问题、解决问题的能力,通过亲身实践理解 Java 语言的基本原理和编程方法。

教育部高等学校计算机科学与技术教学指导委员会非计算机专业计算机基础课程教学指导分委员会《关于进一步加强高等学校计算机基础教学的意见》中明确指出,要加强计算机课程的实践教学,注重能力的培养。意见中建议计算机实验课程分三个层次进行设计,即基础与验证型实验、设计与开发型实验、研究与创新型实验。本书正是按照这个思路,分层次地设计了验证型实验、设计开发型实验和创新型实验。

本书第一部分提供《Java 程序设计简明教程》第一、二层次的实验指导,共设计了 16 个教学实验;本书第二部分主要针对第三层次的实验教学,设计了三个比较大的综合实例,并对这些实例的分析和设计过程进行了详细的指导,供课程教学后的全日制综合实训选用;考虑既要有教学辅导意义,又要给学生留有适当的余地,本书的第三部分对《Java 程序设计简明教程》的部分习题进行了解答;第四部分是“Java 语言程序设计”课程的模拟试题及解答;为满足部分学生参加全国计算机等级考试的需要,第五部分提供了全国计算机等级考试 Java 语言考试大纲,以及全真的笔试、机试试题及解答。

本书由陈语林任主编并统稿,刘胤宏、王新英、杜伟任副主编。陈语林编写了第一部分的实验一至实验八、第二部分的实训项目三以及第四、五部分,刘胤宏编写了第二部分的实训项目一,杜伟编写了第二部分的实训项目二,王新英编写了第一部分的实验九至实验十六。参加本书编写工作的还有梁建武、刘建成、王鹰、刘军军、石山鹰、孔志周、何志斌、龙晓梅、田野、周媛媛、文拯、石玉英等。

在本书编写的过程中,得到了许多专家和同仁的热情帮助和大力支持,中国水利水电出版社的领导和编辑也付出了艰辛的劳动,谨此向他们表示最真挚的感谢!

由于计算机技术发展迅速且作者水平所限,书中错误和疏漏在所难免,敬请专家和广大读者批评指正。

加“*”号的章节供学有余力的学生选用。

编 者
2009 年 6 月

目 录

前言

第一部分 Java 程序设计实验指导	1
实验一 安装运行环境及简单 Java 程序的编制	2
实验二 Java 语言基础	8
实验三 程序控制结构一：分支结构	16
实验四 程序控制结构二：循环结构	20
实验五 数组的使用	23
实验六 类与对象	26
实验七 继承与多态	35
实验八 常用基础类库与工具类库	46
实验九 Java 异常处理	51
实验十 Applet 小应用程序	58
实验十一 AWT 常用组件的应用	62
实验十二 事件处理	65
实验十三 Swing 常用组件的应用	67
实验十四 多线程机制	70
实验十五 Java 输入/输出流	74
*实验十六 网络编程	77
第二部分 Java 程序设计综合实训	80
实训项目一 学生管理信息系统开发实例	81
实训项目二 “俄罗斯方块”游戏的设计 与实现	110
实训项目三 “连连看”游戏的设计与实现	150
第三部分 教材习题选解	171
第 1 章 Java 语言及其开发环境	171
第 2 章 Java 语言基础	171
第 3 章 程序控制结构	172

第 4 章 数组	174
第 5 章 Java 面向对象程序设计基础	176
第 6 章 Applet 程序的开发	178
第 7 章 创建图形用户界面	180
第 8 章 数据库编程	180
第 9 章 多媒体应用	181
第 10 章 流、多线程与网络编程	181
第 11 章 错误与异常处理	186
第 12 章 开发工具的应用	187
第四部分 模拟试题	188
模拟考试试题（笔试）	188
模拟考试试题（笔试）参考答案	192
模拟考试试题（上机）	194
模拟考试试题（上机）提示	195
第五部分 关于全国计算机等级考试	197
全国计算机等级考试（NCRE）考试大纲	197
全国计算机等级考试二级 Java 语言程序设计 （笔试）样题	199
全国计算机等级考试二级 Java 语言程序设计 （笔试）样题参考答案	203
全国计算机等级考试二级 Java 语言程序设计 （机试）样题	203
全国计算机等级考试二级 Java 语言程序设计 （机试）样题参考答案	207
参考文献	208

第一部分 Java 程序设计实验指导

一、实验环节教学目的

上机实验是程序设计语言的一个重要教学环节。根据课程的教学计划，应当在讲授课程内容的同时，每周安排一定时间进行上机实验。

通过系统的上机练习，力求能够达到以下四个目的：

(1) 加深学生对课堂讲授内容的理解，从计算机语言的基本概念、程序设计的基本方法、语法规则等方面加深理解，为程序设计、开发软件打下良好的基础。

(2) 在上机实验中，提高学生对 Java 语言各部分内容的综合使用能力，逐步掌握 Java 语言程序设计的规律与技巧。

(3) 在对 Java 程序的调试过程中，提高学生分析程序中出现的错误和排除这些错误的能力。

(4) 通过上机实践，使学生加深对计算机软件运行环境，以及对操作系统与计算机语言支持系统相互关系的了解。

二、对学生上机实验提出的要求

为提高实验教学效果，特对学生提出如下要求：

(1) 上机实验之前，学生应当为每次上机的内容作好充分准备。对每次上机需要完成的题目进行认真的分析，列出实验具体步骤，写出符合题目要求的程序清单，准备出调试程序使用的数据，以便提高上机实验的效率。

(2) 按照实验目的和实验内容，以及思考题的要求进行上机操作。录入程序，编译调试，反复修改，直至程序正常运行，得出正确的输出结果为止。

(3) 根据实验结果，写出实验报告。实验报告应当包括实验题目、程序清单、运行结果，以及通过上机取得了哪些经验。

三、实验教学内容与课时

本书第一部分所列出的上机实验题目，基本包括了教材中涉及的主要知识点。教师在组织教学实验时，可有针对性地选择其中一部分作为必做题，其余的实验题供学有余力的学生使用。

在设计教学方案时，建议安排 36~48 学时作为课程教学配套的实验。在此基础上，学生还可以课余自由抽出一定的时间作为补充。

为了使学生在上机实验中得到更大的提高，任课教师还可以结合具体情况补充一些专业性、针对性更强的实验题目。在学校实验条件允许的前提下，可把上机实验时间适当加长一些，这对提高学生程序设计能力的提高更加有利。

实验一 安装运行环境及简单 Java 程序的编制

一、实验目的

- 掌握下载和安装 JDK 开发工具的方法。
- 熟悉 JDK 开发工具的使用。
- 学习编写简单的 Java Application 和 Java Applet 程序。
- 了解 Java 程序的基本组成结构。
- 了解 Java 语言的概貌, 了解 Java 源代码、字节码文件, 掌握 Java 程序的编辑、编译和运行过程。

二、实验要求

从网络下载 Java 开发工具 JDK, 进行环境设置, 编写简单的 Java Application 和 Java Applet 程序, 编译并运行这两个程序。

三、知识要点

1. Java 运行环境的安装

JDK (Java Development Kit) 是一切 Java 应用程序的基础, 可以说, 所有的 Java 应用程序是构建在这个平台之上的。要想正确编译和运行 Java 程序, 必须先安装好 JDK。它是一组 API, 也可以说是一些 Java Class。考虑到大多数用户都使用 Microsoft 系统, 所以, 在这里主要介绍 Windows 2000/XP 系统下的安装。

(1) 在 <http://java.sun.com> 中根据自己的操作系统下载安装程序, 按安装说明进行安装。

(2) 设置环境变量。以 Windows 2000 为例, 右击“我的电脑”图标, 从弹出的快捷菜单中单击“属性”命令, 打开“系统属性”对话框, 选择“高级”选项卡, 单击“环境变量”按钮, 打开“环境变量”对话框, 在“系统变量”列表框中选中变量 path 进行编辑, 在后面加入“;C:\Program Files\Java\jdk1.7.0”。注意: 这里的目录 C:\Program Files\Java\jdk1.7.0 是安装 Java 的目录。相关设置的具体方法详见教材《Java 程序设计简明教程》。

(3) 打开命令提示符窗口, 输入 javac 命令, 按回车键, 如果出现命令的使用参数说明, 则表示设置正确, 如果出现非法命令的错误提示, 则应该重新设置。

2. Java 程序调试步骤

Java 程序的调试过程如图 1-1 所示。

(1) 源程序的编辑。

可用“记事本”等文本编辑器对 Java 源程序进行输入和修改。源程序文件的名字一定要与 Java 的主类名称保持一致。所谓主类是指 Java 应用程序中含有 main 方法的类, Java Applet 程序中继承 Applet 的类。

(2) 对源程序进行编译。

命令格式: javac 文件名.java

编译的目的是将 Java 源程序转化为字节码文件, 只有字节码文件才能被 Java 解释程序运

行。如果程序中有语法错误，则在命令行将显示出错信息，错误信息中会指示出错的行、错误类型，用户要仔细查看错误提示，根据错误提示修改源程序。每次改动程序后要重新编译，直到无错才执行程序。

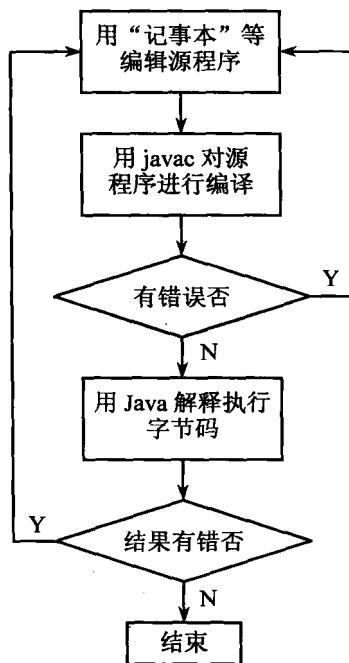


图 1-1 程序调试过程

(3) Java 应用程序的运行。

命令格式：java 文件名

执行程序是执行字节码文件（.class 类型的文件），但命令行在 java 命令后只要给出文件名即可。运行程序后将产生输出结果，用户可检查分析结果是否正确，如果不正确，说明程序逻辑思路存在问题，检查改正逻辑错误后重新编译运行。

(4) Applet 程序的运行。

在 Java Applet 程序编译通过后，运行时必须嵌入在一个 HTML 文件中，因此，需要制作一个 HTML 文件（不妨假设名称为 x.htm），文件内容中包括 Applet 标记。HTML 文件与字节码文件通常放在同一目录下。

查看 Applet 运行结果有两种方法：

方法 1：用浏览器访问 HTML 文件；

方法 2：用 appletviewer.exe 运行 HTML 文件。

例如：appletviewer x.htm

3. 常见错误处理

错误 1：javac 不是内部或外部命令，也不是可运行的程序或批处理文件，即提示信息为“javac Command not found”。

这种错误产生的原因是没有设置好环境变量 path。

处理办法：安装 Java 后，检查环境变量设置是否正确。

错误 2：在命令行执行 `javac HelloWorld` 出现提示信息 “HelloWorld is an invalid option or argument.”。

处理办法：Java 的源程序是一定要保存为 .java 文件的，而且编译时文件名要写全，即编译命令为 `javac HelloWorld.java`。

错误 3：执行 `javac HelloWorld.java` 时显示找不到文件 HelloWorld.java 文件。

处理办法：进入到 HelloWorld.java 所在的目录进行操作。查看当前目录下是否有 HelloWorld.java 文件，如果发现保存的文件为 HelloWorld.java.txt，则说明保存文件时未选择文件类型，Windows 自动给文件添加后缀。此时，应打开该文件，并另存为“所有文件”类型，如图 1-2 所示。

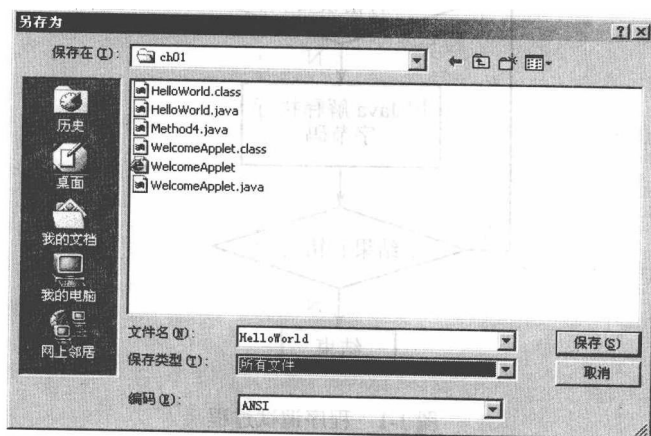


图 1-2 用“记事本”编辑 Java 源程序时，注意文件保存类型

错误 4：在命令行执行 `javac HelloWorld.java` 出现以下提示信息：

```
HelloWorld.java:1: Public class helloworld must be defined in a file called
HelloWorld.java.
public class helloworld{
^
```

出现这种错误的原因是类的名字与文件的名字不一致。准确地说，一个 Java 源程序中可以定义多个类，但是，具有 `public` 属性的类只能有一个，而且要与文件名一致；此外，`main` 方法一定要放在这个 `public` 的类之中，这样才能 `java`（运行）这个类。对于初学者，特别要注意的一点是 Java 语言里面是严格区分大小写的。

处理办法：按照以上指出的出错原因仔细修改。

错误 5：文件编译通过，但运行时显示找不到类。

处理办法：有可能文件名与类名不一致，而且类名未定义为 `public`。例如：程序文件名为 `hello.java`，编译通过，运行时输入 `java hello` 则出现 “NoClassDefFoundError” 的错误提示，运行输入 `java Hello` 可输出结果。还有可能是系统的 `classpath` 环境变量未将当前目录作为类的搜索路径。只要将代表当前目录的 “.” 放到 `classpath` 的设置值中即可。

错误 6：在命令行执行 `java HelloWorld` 出现如下提示信息：

```
Exception in thread main java.lang.NoSuchMethodError main
```

出现这种错误的原因是，在 main 方法的定义上，没有写对地方或写错。

处理办法：检查 main 方法的定义，是不是这样写的：

```
public static void main(String args[]) {...}
```

注意一个字都不要差，包括大小写。

4. Java 程序数据的输出显示

(1) 字符方式下数据的输出。

`System.out.println(数据);`

其中，数据可以是任何类型，利用“+”号可以将字符串与任意类型数据进行拼接。

(2) 图形方式下数据的绘制。

- `g.drawString(字符串,x,y)`

功能：在(x, y)位置绘制一个字符串。

- `g.setColor(Color c)`

功能：设置画笔的颜色。

- `g.drawLine(x1,y1,x2,y2)`

功能：从(x1, y1)到(x2, y2)绘制一条直线。

其中，g 代表 Graphics 对象（画笔），x、y 为输出坐标位置。

四、实验内容

1. 下载、安装

JDK 下载地址为 Java 官方站点：java.sun.com。

下载后，在 Windows 下直接运行 .exe 文件，安装到一个目录，这里用 C:\Program Files\Java\jdk1.7.0 作为例子。

安装完毕后，右击“我的电脑”图标，在弹出的快捷菜单中单击“属性”命令，打开“系统属性”对话框，单击“高级”选项卡，然后单击“环境变量”按钮，显示如图 1-3 所示的对话框。

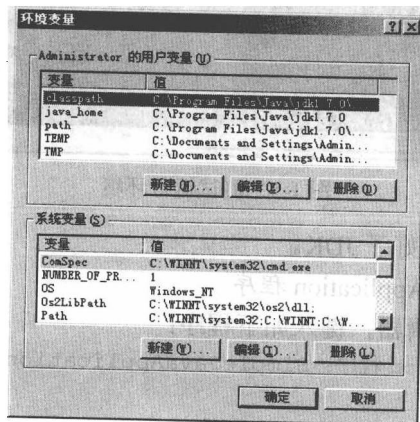


图 1-3 环境变量设置

在“系统变量”列表框中选定 Path，然后单击“编辑”按钮，在弹出的如图 1-4 所示的对话框中加入 C:\Program Files\Java\jdk1.7.0\bin（这个路径为 JDK 安装的实际路径），然后单击“确定”按钮退出，并重新启动计算机。

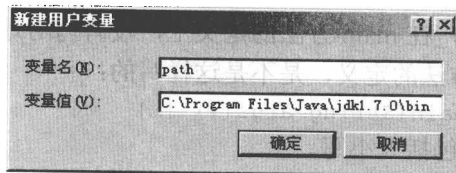


图 1-4 新建用户变量

2. 测试环境

(1) 用“记事本”编写一个简单的 java 程序：

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

这个例子的功能就是显示“Hello, Java World!”。

注意：该文件名称必须为 HelloWorld.java，大小写也区分。

(2) 编译：单击“开始”→“运行”命令，在“运行”对话框中键入命令 cmd，在命令提示符下执行：

```
javac HelloWorld.java
```

如果正常的话，将生成 HelloWorld.class 文件。

(3) 运行：在命令提示符下执行：

```
java HelloWorld
```

一定要注意不要输入：java HelloWorld.class，否则会出现下面的错误：

Exception in thread "main" java.lang.NoClassDefFoundError:HelloWorld/class

好了，运行 java HelloWorld 应该会出现“Hello, Java World!”了。运行结果如图 1-5 所示。

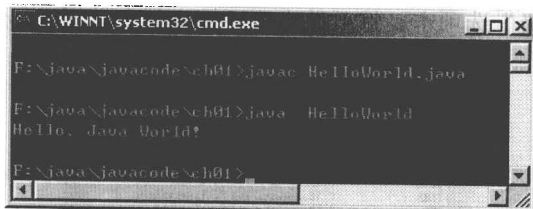


图 1-5 测试运行环境

到这一步，已经成功安装好了 JDK。

3. 编写并运行一个 Java Application 程序

(1) 打开一个纯文本编辑器，键入如下程序：

```
//简单的 Java Application 程序: SimpleJavaApplication.java
import java.io.*;
```

```
public class SimpleJavaApplication
{
```

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

```
        System.out.println("A Simple Java Application Program!");
```

```
}  
}
```

(2) 检查输入无误后把文件保存起来, 命名为 SimpleJavaApplication.java。进入命令提示符窗口, 在保存上述文件的目录下运行 Java 编译器。

(3) 如果 J2SE 安装正确, 程序输入无误, 保存的文件名正确, 则运行编译器无任何输出, 否则会输出错误信息, 一般都是由于拼写错误引起的, 在命令提示符下用 dir 命令查看是否在同目录下生成了一个名为 SimpleJavaApplication.class 的文件。

(4) 利用 Java 解释器运行这个程序, 并查看运行结果。

4. 编写并运行一个 Java Applet 程序

(1) 打开一个纯文本编辑器, 键入如下程序:

```
//简单的 Java Application 程序: SimpleJavaApplet.java  
import java.awt.Graphics;  
import java.applet.Applet;  
public class SimpleJavaApplet extends Applet{  
    public void paint(Graphics g){  
        g.drawString("A Simple Java Applet !", 15 , 20);  
    }  
}
```

(2) 检查无误后保存为 SimpleJavaApplet.java。

(3) 进入命令提示符窗口, 在保存的目录下运行 Java 编译器:

```
javac SimpleJavaApplet.java.
```

(4) 如果 J2SE 安装正确, 程序输入无误, 保存的文件名正确, 则运行编译器无任何输出, 否则会输出错误信息, 一般都是由于拼写错误引起的, 在命令提示符下用 dir 命令查看是否在同目录下生成了一个名为 SimpleJavaApplet.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, 键入如下程序:

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

(2) 检查无误后命名为 SimpleJavaApplet.html, 与文件 SimpleJavaApplet.class 保存到用一个目录下。

(3) 直接双击 HTML 文件的图标, 或打开 Web 浏览器, 在地址栏里输入全路径名查看运行结果。

(4) 利用模拟的 Applet 运行环境 Appletviewer SimpleJavaApplet.html, 解释运行这个 Java Applet 程序并观察运行结果。

五、实验思考与练习

- (1) 什么是 Java 虚拟机？它的作用是什么？
- (2) Java Application 程序与 Java Applet 程序的不同之处有哪些？
- (3) 请收集错误代码与提示信息，并分析产生错误的原因。
- (4) 编写 Java 程序，在屏幕上显示 “How are you?” 和 “I’m fine.Thank you!”。
- (5) 编写 Java 小程序，参考上述 Java Applet 程序，在屏幕中央输出两行语句 “Hello, China!” 和 “Hello,Asia!”。
- (6) 运行下面程序，分析程序运行结果。

```
import java.io.*;
public class Exer1_6
{
    public static void main(String args[])
    {
        char i=' ';
        System.out.println("Input an integer or character:");
        try
        {
            i=(char)System.in.read();           //从键盘中读入一个字符
        }
        catch (IOException e)
        { }
        System.out.println("The input is '"+i+"'");
    }
}
```

(7) 第 6 题中，如果将 try 语句和 catch 语句去掉，保留 try 大括号中的语句，程序能正确运行吗？如不能正确运行，错误信息是什么？

(8) 如果将第 6 题中 “import java.io.*;” 语句去掉，程序能正确运行吗？如不能正确运行，错误信息是什么？

实验二 Java 语言基础

一、实验目的

- 掌握标识符的定义规则。
- 掌握表达式的组成。
- 掌握 Java 基本数据类型及其使用方法。
- 理解定义变量的作用，掌握定义变量的方法。
- 掌握各种运算符的使用及其优先级控制。
- 通过以上内容，掌握 Java 语言的编程规则。

二、实验要求

- 编写一个声明 Java 不同数据类型变量的程序。
- 编写一个使用运算符、表达式、变量的程序。
- 编写表达式语句、复合语句的程序。

三、实验内容

1. 声明不同数据类型的变量

(1) 编写声明不同数据类型变量的程序，源代码如下：

```
public class LX2_1 {  
    public static void main(String args[]) {  
        byte b=0x55;  
        short s=0x55ff;  
        int i=1000000;  
        long l=0xfffffL;  
        char c='a';  
        float f=0.23F;  
        double d=0.7E-3;  
        boolean B=true;  
        String S="这是字符串数据类型";  
        System.out.println("字节型变量 b = "+b);  
        System.out.println("短整型变量 s = "+s);  
        System.out.println("整型变量 i = "+i);  
        System.out.println("长整型变量 l = "+l);  
        System.out.println("字符型变量 c = "+c);  
        System.out.println("浮点型变量 f = "+f);  
        System.out.println("双精度变量 d = "+d);  
        System.out.println("布尔型变量 B = "+B);  
        System.out.println("字符串类对象 S = "+S);  
    }  
}
```

(2) 编译并运行该程序，结果如图 1-6 所示。

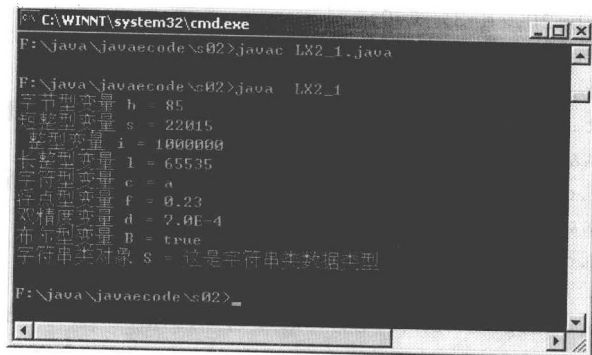


图 1-6 不同数据类型的变量运算结果

2. 了解变量的使用范围

(1) 建立 LX2_2.java 文件, 通过本程序了解变量的使用范围, 源代码如下:

```
public class LX2_2 {
    static int i=10;
    public static void main(String args[]) {
        {
            int k=10;
            System.out.println("i="+i);
            System.out.println("k="+k);
        }
        System.out.println("i="+i);
        System.out.println("k="+k);
        //编译时将出错, 已超出 k 的使用范围
    }
}
```

(2) 编译 LX2_2.java。

此时会出现错误提示, 如图 1-7 所示。因为变量 k 在方法中的语句块中声明, 在方法块之外它是不存在的, 所以编译时会出错。

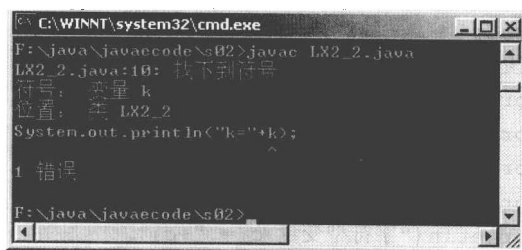


图 1-7 注意变量的使用范围

(3) 思考: LX2_2.java 程序说明了什么问题?

3. 使用关系运算符和逻辑运算符

(1) 建立使用关系运算符和逻辑运算符的程序文件, 源代码如下:

```
class LX2_3 {
    public static void main(String args[]) {
        int a=25, b=20, e=3, f=0;
        boolean d=a<b;
        System.out.println("a=25,b=20,e=3,f=0");
        System.out.println("因为关系表达式 a<b 为假, 所以其逻辑值为 : "+d);
        if (e!=0 && a/e>5)
            System.out.println("因为 e 非 0 且 a/e 为 8 大于 5, 所以输出 a/e= "+a/e);
        if (f!=0 && a/f>5)
            System.out.println("a/f = "+a/f);
        else
            System.out.println("因为 f 值为 0, 所以输出 f = "+f);
    }
}
```

(2) 编译并运行该程序, 请写出运行结果。

4. 使用表达式语句与复合语句

(1) 建立包含表达式语句程序, 源代码如下:

```
class LX2_4{
    public static void main(String[] args) {
        int k, i=3, j=4;
        k=20*8/4+i+j*i;
        System.out.println("表达式 (20*8/4+i+j*i)="+k);
    }
}
```

(2) 建立包含复合语句程序, 源代码如下:

```
class LX2_5{
    public static void main(String args[]) {
        int k, i=3, j=4;
        k=i+j;
        System.out.println("在复合块外的输出 k="+k);
        {
            float f;
            f=j+4.5F;
            i++;
            System.out.println("在复合块内的输出 f="+f);
            System.out.println("在复合块内的输出 k="+k);
        }
        System.out.println("在复合块外的输出 i="+i);
    }
}
```

(3) 编译并运行上述两个源程序, 并写出运行结果。

5. 编写字符界面的 Java Application

接收用户输入的一个字符, 并把该字符打印在屏幕上。

(1) 键入并运行如下 Java Application 程序:

//字符界面下的基本输入/输出程序: SimpleCharInOut.java

```
import java.io.*;
public class SimpleCharInOut{
    public static void main(String args[]){
        char ch=' '; //定义字符型变量
        System.out.print("Enter a character please:");
        //在屏幕上显示提示信息
        try{
            ch=(char)System.in.read(); //接收用户键盘输入的字符并保存
        }catch(IOException e){}; //在变量 ch 中
        System.out.println("You've entered character:"+ch);
        //输出 ch 字符
    }
}
```

(2) 回答问题: 运行该程序时屏幕显示什么? 如果不敲击键盘输入字符, 程序会如何反应? 如果敲击键盘输入字符但不按回车键, 程序会如何反应? 为什么? 如何结束此程序的运行? 如果从键盘输入了一个字符并按回车键, 程序输出什么? 如果从键盘输入了多个字符并按回车键, 程序又输出什么? 为什么?

6. 编写图形界面的 Java Applet

其中包括一个按钮、一个文本域和一个标签。当用户在文本域中输入若干字符并单击按钮, 将用户输入的字符显示在标签上。

(1) 键入并运行如下的 Java Applet 程序:

```
//图形界面的基本输入/输出程序:
//文件名为 SimpleGraphicsInOut.java
import javax.swing.*;
import java.awt.*;
import java.applet.Applet;
import java.awt.event.*;

public class SimpleGraphicsInOut extends Applet implements ActionListener{
    JLabel prompt;          //定义显示提示信息的标签对象
    JTextField input;       //定义接受用户键盘输入的文本区域对象
    JLabel output;         //定义回显用户输入的标签对象
    public void init()     //定义浏览器打开 Applet 后的初始操作
    {
        prompt=new JLabel("Enter a character please:");    //创建提示标签
        input= new JTextField(3);                          //创建输入区域
        output=new JLabel("          ");                  //创建输入标签
        add(prompt);    //将提示标签加入图形界面
        add(input);     //将输入区域加入图形界面
        input.addActionListener(this);
        add(output);    //将输出区域加入图形界面
    }
    public void actionPerformed(ActionEvent e)//定义对用户输入的响应
    {
        //在输出标签回显输入
        output.setText("You've entered character:"+input.getText());
    }
}
```

(2) 编写嵌入这个 Applet 程序的 HTML 文件 SimpleGraphicsInOut.html:

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