

高职高专计算机实用规划教材

—— 案例驱动与项目实践



本书特色：

- 以应用能力培养为基础，以提高职业素质为目的
- 将职业能力培养与课程相结合，精心设计工作任务模块
- 以工作任务为核心，贯穿章节内容，编排贴近实际开发环境

基于工作过程的 Java程序设计

魏 勇 编著

■ Java

■ 案例驱动

■ 项目实践



提供代码、电子教案下载
<http://www.tup.com.cn>



清华大学出版社

高职高专计算机实用规划教材——案例驱动与项目实践

基于工作过程的 Java 程序设计

魏 勇 编著

清华大学出版社

北 京

内 容 简 介

本书打破传统的以学科为中心的体系,而是以工作任务为中心,从Java最简单的程序开始,介绍Java的基本程序结构。

封装、继承、多态是面向对象程序设计的3个重要特征。本书从人“类”入手,介绍Java中的类、对象、属性和方法等概念。通过人类在生物图中的关系,帮助读者理解继承等概念。

本书把初始化一个Human实例作为一个核心工作任务,通过该工作任务在各章节的进一步展开,来讲解Java的异常处理、图形用户界面、SWT技术及流技术等内容。

线程是Java实现并行处理的重要技术,在线程一章介绍了Java线程的基本概念,讲述如何在Java中编写线程程序以及多线程的程序设计等。

本书还安排了实验与实训内容,便于读者在学完基本知识的前提下,提高实际编程和调试能力。

本书适合作为高职高专计算机专业教材,也可供程序设计与开发人员参考。

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

基于工作过程的Java程序设计/魏勇编著. —北京:清华大学出版社,2010.1

(高职高专计算机实用规划教材——案例驱动与项目实践)

ISBN 978-7-302-21535-6

I. 基… II. 魏… III. JAVA语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2008)第220148号

责任编辑:黄飞

装帧设计:杨玉兰

责任校对:王晖

责任印制:王秀菊

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:北京市清华园胶印厂

装 订 者:北京国马印刷厂

经 销:全国新华书店

开 本:185×260 印 张:18 字 数:389千字

版 次:2010年1月第1版 印 次:2010年1月第1次印刷

印 数:1~4000

定 价:27.00元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:033726-01

前 言

Java 是一种跨平台、适合于分布式计算环境的面向对象的编程语言。它有许多值得称道的优点，如简单、面向对象、分布式、解释性、可靠、安全、结构中立性、可移植性、高性能、多线程和动态性等。Java 摒弃了 C++ 中各种弊大于利的功能和许多很少用到的功能。Java 可以运行于各种操作系统环境，特别适合网络应用程序的开发。Java 程序语言是软件技术专业的一门很重要的基础课程。

国内外现存同类教材在内容组织上已比较成熟。但普遍注重以学科为中心，编排形式以介绍 Java 程序设计知识为主，辅以例子加以验证。由于 Java 系统庞大，引入的如面向对象等概念较多，使得初学者难以入门。

为了贯彻“以能力本位教育为基础，以提高职业素质为目的”的指导思想，我们把教学目标定位在职业技能培养与训练贯穿于高职教学的全过程，重点培养学生分析和解决岗位实际问题的能力。

本书欲打破传统的以学科为中心的教材体系，内容编排贴近实际开发环境，时刻遵循能力培养规律，并结合课程实际，设计了与该课程能力目标要求紧密结合的工作任务模块，采用由简单到复杂的递进方式进行教学设计。将理论讲解与实践紧密结合，在完成任务的过程中，使学生掌握分析问题和解决问题的策略，体验到知识的应用价值。

为贯彻以工作过程为导向，本书基于工作任务来确定课程的学习任务，突出职业素质培养。本书每部分的实际内容都从提出一个具体的实际工作任务开始，分别由详细设计、编码实现、源代码、测试与运行、技术分析和问题与思考几个步骤来完成。每个步骤需要达到的目的如下。

1. 详细设计

这部分提出实现本任务的基本程序框架和主要算法等。

2. 编码实现

如何用 Java 语句实现以上设计。对重点语句进行分析和说明。

3. 源代码

这个步骤给出实现程序的完整源程序。读者可以逐步锻炼如何在前两个步骤的基础上，写出源程序，从而达到最终完成设计和编写源程序的目的。

4. 测试与运行

对以上程序进行测试，有时用几组数据直接运行以上程序进行测试，有时需要编写测试程序，并对结果进行基本的分析。



5. 技术分析

以上步骤基本上是围绕提出的工作任务而展开的,对引出的知识,需要系统地整理。如果按学科体系组织教学内容,这个步骤放在最前面,然后再通过一些例子验证。基于工作过程与以往教材不同的是,本书每个具体内容都先让读者知道如何做,再去梳理做的过程中所涉及的知识。

6. 问题与思考

这个步骤对学习过程中遗存的一些问题展开讨论,一来为以后的知识作一些铺垫,二来对所学内容起到举一反三的作用。

以 Java 的基本程序结构章节为例,按学科知识体系可由顺序、分支和循环 3 种方式构成。尤其是分支和循环,如果没有其他语言基础,初学者在理解时会感到有一定的难度。本书一改过去以学科知识为中心,而采用基于工作过程的编排风格。以第 2 章内容为例,为了按工作任务组织教学内容,把 Java 基本程序结构的工作任务确定为查找素数,并从分支和循环两个方面去实现这一目标。具体来讲,if-else 分支语句隐含在实现查找最大数的工作任务中,switch 语句是通过如何判断某年某月的工作任务来介绍的。循环语句也是通过两个子任务讲授的,即“判断一个正整数 n 是否为素数”来引入 Java 的一般循环结构,任务“查找区间内的素数”引入循环嵌套。按工作任务安排基本程序结构章节的内容如下:

查找素数

分支语句

再谈找两数中较大数

从 3 个数中找出最大数

判断某年某月的天数

循环语句

判断一个正整数 n 是否为素数

查找区间内的素数

作为本书的重点章节,第 4 章介绍面向对象的程序思想。封装、继承、多态是面向对象程序设计的 3 个特点。本书始终用 Java 如何编写“人”类这个容易理解的概念入手,展开介绍 Java 中的类、对象、属性和方法等。“人”类即是一个类,通过“人”类与每个人之间的关系建立 Java 类与对象的概念。把“人”类放在生物关系图中,帮助理解继承等概念。本书反复用编写的“人”类 Human 作为例子,使初学者学习 Java 编程更容易并融会贯通。

本书另一个重要特色是把初始化一个 Human 实例作为核心工作任务,该工作任务贯穿在大部分重要章节中。Java 中如何声明异常、抛出异常和捕获异常均体现在初始化一个 Human 对象时,来自于一个错误的性别字符。在图形用户界面一章,通过图形用户界面接收数据来完成初始化 Human 实例的工作任务。本书也利用 SWT 构成的图形界面接收 Human 对象的初始化数据。Java 流这一章是从一个文件流中获取 Human 对象的初始化数据。把初



始化一个 Human 实例作为本书的一个核心工作任务，不仅让读者在学习相关知识时能够举一反三，而且深入浅出地为读者展示着看似深奥的 Java 编程技术。

AWT(Abstract Windows Toolkit)是最原始的 Java GUI 工具包，它是一个非常简单的具有有限 GUI 组件、布局管理器和事件的工具包。Java Swing 是试图解决 AWT 缺点的一个尝试。然而自 IBM 公司提供了跨平台 GUI 开发包 SWT 以来，越来越受到广大程序员的青睐，已经有不少程序员用它开发出美观、高效、实用的桌面应用程序。SWT 在性能、外观上以及易用性上，比起 Swing 来确实要好得多。SWT 的构建者从 AWT 和 Swing 实现中学习了很多经验，成功地构建成一个集 AWT 和 Swing 二者优点于一体的系统。为适应高职教育的特点，实现教学内容与企业应用的零距离，本书率先在国内同类教材中引入 SWT/JFace 技术。在 Java 图形用户界面一章，通过 AWT 和 Swing 介绍了 Java 图像界面与事件处理机制。作为教材，在 SWT 技术一章没有花太大的篇幅涉及更多的细节，但可以起到抛砖引玉的作用。

本书中实例和例子程序都已调试通过，因而读者在上机实践时，不会出现不必要的困惑。

本书最后还安排了项目实践内容，便于读者在学完基本知识的前提下，提高实际编程和调试能力。

本书在编写过程中，得到了清华大学出版社和深圳信息职业技术学院的大力支持，深圳信息职业技术学院还专门设立教材建设项目予以支持，在此表示衷心的感谢！由于时间和水平所限，书中难免有不妥之处，欢迎各界专家和读者朋友批评指正，也欢迎读者来电交流，本人的联系方式是 E-mail: weiuser@hotmail.com，电话：0755-25859280。

编 者

目 录

第 1 章 简单 Java 程序	1	4.6 Java 中的抽象类与接口	91
1.1 屏幕上显示一句话的 Java 程序	1	4.6.1 抽象类计算几何形状的面积	91
1.2 基本数据类型及运算	6	4.6.2 接口计算几何形状面积	94
1.3 把 1、2、3 累加到变量	9	第 5 章 Java 异常处理	99
1.4 运算符	15	5.1 捕获异常	99
1.4.1 找 6 的所有因子	15	5.2 自定义异常	104
1.4.2 求 6!	21	第 6 章 Java 图形用户界面	110
1.4.3 找两数中较大数	24	6.1 通过图形界面输入数据 初始化 Human 对象	110
第 2 章 查找素数	28	6.2 Java Applets	121
2.1 分支语句	28	6.2.1 在网页中显示一句话的 程序	121
2.1.1 再谈找两数中较大数	28	6.2.2 Applets 应用	129
2.1.2 从 3 个数中找出最大数	31	6.3 匿名类简化图形事件处理程序	135
2.1.3 判断某年某月的天数	33	6.4 围绕圆心转的圆	141
2.2 循环语句	36	6.5 创建和使用菜单	151
2.2.1 判断一个正整数 n 是否 为素数	36	6.6 应用 Swing 创建用户界面	158
2.2.2 查找区间内的素数	41	6.7 图形打印	162
第 3 章 数组与字符串	45	第 7 章 SWT 技术	168
3.1 在 10 个整数中找最大数	45	7.1 用 SWT 技术初始化 Human 对象	168
3.2 建立并输出一个矩阵	49	7.2 在左、右两个 SWT 列表框中 交换数据	176
3.3 Vector 实现堆栈的压入和弹出操作	53	7.3 SWT 实现选项卡	187
3.4 字符串操作	57	7.4 一个 JFace 程序	192
第 4 章 Java 面向对象程序设计	62	7.5 JFace 实现表格	196
4.1 编写“人”类	62	7.6 JFace 实现树	205
4.2 把类打包	67	第 8 章 Java 的流	212
4.3 为每个“人”生成唯一编号	73	8.1 从键盘上输入字符	212
4.4 在“人”类基础上编写 “教师”类 Teacher	78	8.2 文件流	218
4.5 教师编码的生成	84		



8.2.1 从一个文件中读入数据 初始化 Human 对象	218	10.3 计算后缀表达式	260
8.2.2 把对象按流进行读、写	228	10.3.1 实训内容及要求	260
第 9 章 Java 线程程序	232	10.3.2 实训过程	260
9.1 并行程序设计	232	10.4 Java 读取 XML 文件	261
9.2 动画实现	237	10.4.1 实训内容及要求	261
9.3 分别对堆栈进行压入和出栈的 并行程序	243	10.4.2 实训过程	261
9.4 线程的同步处理	250	10.5 利用 JMF 编写摄像头拍照程序	265
第 10 章 实验与实训	256	10.5.1 实训内容及要求	265
10.1 洗牌程序	256	10.5.2 实训过程	266
10.1.1 实训内容及要求	256	10.6 动画实现	270
10.1.2 实训过程	256	10.6.1 实训内容及要求	270
10.2 中缀表达式转化成后缀表达式	258	10.6.2 实训过程	270
10.2.1 实训内容及要求	258	附录 A Eclipse 开发环境的搭建	274
10.2.2 实训过程	259	附录 B Linux 下构建 JDK	277
		参考文献	278

第 1 章 简单 Java 程序

Java 是由 Sun Microsystems 公司于 1995 年 5 月推出的一种面向对象的计算机编程语言。

Java 平台由 Java 虚拟机(Java Virtual Machine)和 Java 应用编程接口(Application Programming Interface, API)构成。Java 的 API 提供了一个独立于操作系统的标准接口。Java 程序只需编译一次,就可以在各种系统中运行。Java 分为 3 个体系 J2SE(Java2 Standard Edition)、J2EE(Java 2 Platform,Enterprise Edition)和 J2ME(Java 2 Micro Edition)。

1.1 屏幕上显示一句话的 Java 程序

知识要点

- ☐ Java 程序编写与运行
- ☐ 类定义
- ☐ main 方法
- ☐ 配置开发环境

Java 是一种面向对象的编程语言,程序编写从类开始,类包含了许多属性和方法。即使最简单的 Java 语言程序,也需编写至少包含有 main()方法的类。

典型实例 编写程序,从屏幕输出一句话: Hello, world!。

1. 详细设计

本程序由类 HelloWorld 实现,它只有一个方法 main(),方法中仅一条输出文字的语句。

```
class HelloWorld {  
main(String arg[]) {  
    输出文字"Hello, World!";  
}  
}
```

2. 输出文字编码实现

语句:

```
System.out.println("Hello, World!");
```

分析 System.out.println 是 Java 输出语句,输出后换行。输出的文字需用双引号括起来。System.out.print 是另一条输出语句,与上一条的不同之处是它输出后不换行。



3. 源代码

```
/* 文件名: HelloWorld.java
 * Copyright (C): 2009
 * 功能: 在屏幕输出文字"Hello, World!"
 */
public class HelloWorld {
    public static void main(String arg[]) {
        System.out.println("Hello, World!");
    }
}
```

4. 测试与运行

1) JDK 的安装

Java 是跨平台的, 无论运行在什么操作系统下, 都需要有 Java 虚拟机。

本书使用的操作系统是 Windows 系列, JDK 用的是 j2sdk-1_4_2-windows-i586.exe 软件。直接运行 j2sdk-1_4_2-windows-i586.exe, 按照提示安装好 JDK。安装 JDK 后在计算机上会显示 java.exe、appletviewer.exe、javac.exe、jar.exe 等程序。其中 javac 命令用于编译源代码, java 文件为.class 类文件, java 命令用于执行编译好的 Java 应用程序(.class 或.jar 文件)。

Sun 公司提供的开发软件包含几个常用术语:

- J2SE 指 Java 2 Standard Edition, 即 Java2 标准版。
- J2EE 指 Java 2 Enterprise Edition, Java2 企业版包括 JSP/Servlet、EJB、JNDI、JTA 等。
- J2SDK 就是 JDK, 是 Java 2 Standard Development Kit 的简写, 是 Java2 的标准开发工具集。
- J2RE 指 JRE, 是 Java 2 Runtime Environment 的简写, 是 Java2 运行时的环境。
- JVM 是 Java Virtual Machine 的简写, 也称 Java 虚拟机。

J2SDK 内包含 J2RE, 一般下载了 J2SDK, 就有编译调试 Java 程序和执行 Java 应用程序(Java Application)及 Java 小程序(Java Applet)的功能。

假设 J2SDK 安装于 c:\j2sdk1.4.2, 则在 c:\j2sdk1.4.2\bin 下一定有 javac.exe 和 java.exe 两个可执行程序。如果把源程序也放在该目录, 就可以进行编译和运行了。如果源程序不放在该目录下, 就需要设置 Path 和 Classpath 等环境。

2) 编译和运行一个 Java 程序

源程序是一个 HelloWorld 类, 必须存入名为 HelloWorld.java 的文件中。编译源程序的命令是 javac.exe, 如果编译成功, 会产生一个和源程序同名的可执行的.class 文件。执行编译好的程序命令是 java.exe。运行结果如图 1-1 所示。

运行一个程序看似很简单, 但初学者在实际操作过程中却会遇到很多意想不到的问题。例如, 在输入源程序时, 不小心有错误, 见下面的程序:

```
public class HelloWorld {
    public static void main(String arg[]) {
```



```

    System.out.println("Hello, World!")
}
}

```

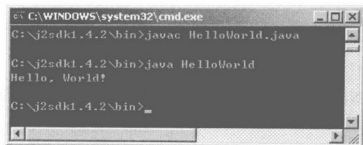


图 1-1 HelloWorld 运行结果

Java 要求每条语句都以";"结尾,main()方法中尽管只有一条语句,这个程序却忽视了这一点,所以在编译时会出现错误,如图 1-2 所示。

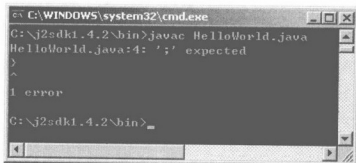


图 1-2 编译时出错

Java 的编译工具 Javac 发现了这个错误并给出提示,显然不可能产生可执行文件 HelloWorld.class,这个时候需要用户重新用编辑工具打开文件 HelloWorld.java 修改错误,重新编译,反复以上过程直至没有语法错误,并生成可执行文件 HelloWorld.class 为止。

5. 技术分析

本节的应用程序“Hello World!”由 3 个主要部分构成:源代码注释、HelloWorld 类定义和 main()方法。

1) 源代码注释

注释会被编译器忽略,但对清楚地表达源程序意图非常有用,如本实例中源程序中的头 4 行。

和 C 语言一样,至少有两种注释类型被 Java 编程语言支持。

/* text */: 编译器忽略从/*到*/之间的所有内容。

// text: 编译器忽略从//到行尾的所有内容。

2) 类定义

面向对象的程序思想是 Java 语言的核心。正因为如此,Java 语言编写程序的过程很符合对客观事物的认识过程。所谓对象就是客观存在的事物,如张三、李四等,以及具体的某台计算机、某本书等。这些对象总是可以按照一定的特征划分为不同的类,如张三、李四属于人“类”,再如计算机类、书类等。Java 中为描述客观存在的一个个对象,就必须



首先从定义类开始。

正如本节实例中定义类 HelloWorld 一样，Java 中类定义的最基本形式是：

```
class Name {  
    :  
}
```

关键字 `class` 为名为 `Name` 的类进行类定义，Java 的类名一般都以大写字母开始。类一般包含属性和方法。

例 1-1 定义人“类” Human。

这里对人“类”取名 `Human`。每个人都有编号、姓名、出生年月等属性，对每个人都需有编码的过程，定义 `Human` 类时，至少要有属性 `code`、`name`、`age`、`birth`，以及进行编码处理的方法 `setCode()` 和获取编码的方法 `getCode()`。

定义类 `Human` 的代码如下：

```
class Human{  
    code;  
    name;  
    age;  
    birth;  
    setCode(codevalue){  
        this.code = codevalue;  
    }  
    String getCode(){  
        return code;  
    }  
}
```

无论是属性还是方法都要用一个数据类型来表示。这里为使读者迅速建立对象、类等概念，简化了 Java 的一些语法细节，接下来的章节会给出 `Human` 的完整定义。

3) main 方法

在 Java 编程语言中，每个应用程序都必须包含 `main` 方法，其格式如下：

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

`public`、`static`、`void` 的含义将在以后的章节中介绍。

该程序中的 `main()` 方法只有一条语句 `System.out.println("Hello World!")`，使用 Java 的 `System` 类把消息 `Hello World!` 发送到标准输出。

`main()` 方法是应用程序的入口点，它可以接收命令行参数。括号 `()` 中的 `String args[]` 表示命令行的第一个参数、第二个参数等，分别保存在 `String` (字符串) 变量 `args[0]`、`args[1]` 等中。或者说 `args` 是保存命令行参数的字符串数组。下面的例子演示 Java 如何接收命令行参数。

例 1-2 编写程序，从屏幕输出一句话 Hello World!，Hello 和 World 从命令行输入。

编写源程序如下：

```
public class Hello World {
```



```
public static void main(String args[]) {
    System.out.println(args[0] + " " + args[1] + "!");
}
}
```

编译运行结果如图 1-3 所示。

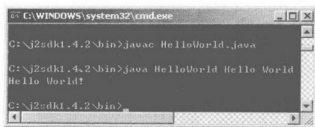


图 1-3 从命令行输入字符串的 HelloWorld 程序

注意到在运行 Hello World 的同时，传送了命令行参数 Hello 和 World。

4) 配置开发环境

本节实例中，需把源程序复制到 JDK 的安装目录 C:\j2sdk1.4.2\bin 中去进行编译和运行。在实际开发过程中，程序员可能在自己定义的目录中进行调试，如希望在目录 D:\myworkspace 中开发程序，这时就需要设置 Java 的开发环境。

首先是设置命令环境 path。编译、运行等命令都在安装目录 C:\j2sdk1.4.2\bin 中，所以需要把该目录设置在 path 变量中，直接用 echo %path% 可以看到当前 path 变量的值，如图 1-4 所示。

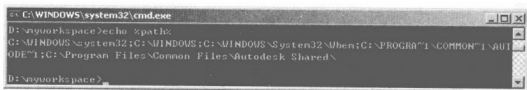


图 1-4 设置 Java 命令环境

用命令 path = %path%; C:\j2sdk1.4.2\bin 添置一个路径，如图 1-5 所示。

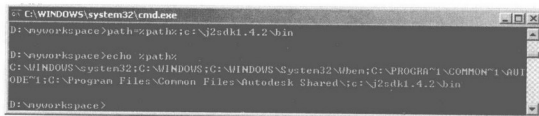


图 1-5 增添命令路径

图 1-5 所示表明，C:\j2sdk1.4.2\bin 成功添入命令环境。

再就是设置类环境 classpath。因为程序中会引用 JDK 中其他类，如 System 等，所以必须告诉编译或运行程序这些类所在的位置。一般情况下，tool.jar 和 dt.jar 是需要放在 classpath 类环境中的，如图 1-6 所示。

注意：classpath 最前面有“.”表示当前目录。

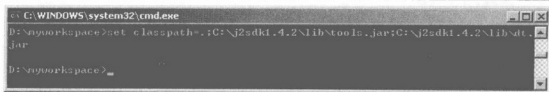


图 1-6 设置类环境 classpath

接下来就可以在 C:\j2sdk1.4.2\bin 目录下编译和运行 Java 程序了, 如图 1-7 所示。

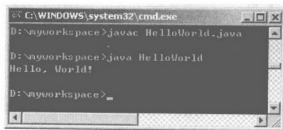


图 1-7 在自设的工作目录中开发 Java 程序

6. 问题与思考

例 1-2 的语句 `System.out.println(args[0] + " " + args[1] + "!");` 中, “+” 起到什么作用?

1.2 基本数据类型及运算

知识要点

- ☐ 基本数据类型
- ☐ Java 对象数据类型

数据类型是对内存位置的抽象表达。Java 简单数据类型是不能再简化的、内置的数据类型, 由编程语言定义, 表示真实的数字、字符和整数。Java 提供了几类简单数据类型表示数字和字符, 通常划分为以下几种类别: 浮点数(实数)、整数、字符串和布尔数, 这些类别中又包含了多种简单类型。

典型实例 整数、浮点数、布尔数、字符串及其运算。

1. 详细设计

本程序由类 `DataCalculate` 实现, 程序处理过程均在方法 `main()` 中完成。该程序分别输出 Java 的整数类型数据、浮点型数据、字符型数据、布尔型数据。

```
class DataCalculate{
    main(String args[]){
        整数运算;
        浮点数运算;
        输出布尔数;
        输出字符;
    }
}
```



2. 编码实现

有以下语句:

```
System.out.println(3+5);
System.out.println(3.5+5.4);
System.out.println(3>5);
System.out.println('J');
```

分析: 一般来说 Java 中只有类型相同才可以相互运算, 3、5 是整数; 3.5、5.4 是浮点数; 3>5 是布尔数; 'J'表示一个字符。

3. 源代码

```
/* 文件名: DataCalculate.java
 * Copyright (C): 2009
 * 功能: 整数、浮点数、布尔数、字符数及其运算
 */
public class DataCalculate{
    public static void main(String args[]){
        System.out.println(3+5);
        System.out.println(3.5+5.4);
        System.out.println(3>5);
        System.out.println('J');
    }
}
```

4. 测试与运行

运行结果如图 1-8 所示。

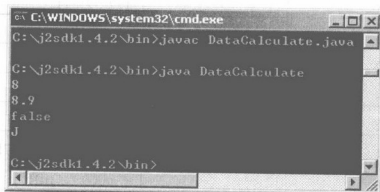


图 1-8 DataCalculate 运行结果

5. 技术分析

类型是用来划分某个事物不同属性的组成部分。

数据类型就是指数据的属性归类。数据可以有多种形式, 比如数字类型的数据 4500(指某人的工资)、32(指某人的年龄), 字符类型的数据 10289(指某人的编码)、张三(代表某人的名字)等。很明显, 这些不同类型的数据无论是其表现形式还是其所能描述的范围都是有自己的限制的, 因此为了对它们进行有效的区分, 才提出了数据类型的概念。

Java 数据类型从大的方面可分为两部分: 基本数据类型和对象数据类型。



1) 基本数据类型

基本数据类型是比较简单地表达一些基本信息的数据类型, Java 有 8 种基本数据类型, 这 8 种基本类型又可归为 4 类。

(1) 整数类型

byte 字节型: 计算机中一个字节一般来说占 8 位。

short 短整型: 占 2 个字节, 也就是 16 位。

int 整型: 占 4 个字节, 也就是 32 位。

long 长整型: 占 8 个字节, 也就是 64 位。

(2) 浮点类型

float 单精度实型: 小数点后面保留 7 位有效数字, 占 32 位。

double 双精度实型: 小数点后面保留 15 位有效数字, 占 64 位。

(3) 字符类型

char 字符型: 占 2 个字节, 也就是 16 位。Java 的 char 类型采用 Unicode 编码。

(4) 布尔类型

boolean 布尔型: 占一个字节, 8 位。Boolean 类型是用来表示逻辑(布尔)型数据的数据类型。Boolean 类型的取值只有 true 和 false。其中, true 表示“真”, false 表示“假”。

各种基本数据类型的取值范围见表 1-1。

表 1-1 各种基本数据类型及表示范围

类 型	位 数	字 节 数	取值范围
整数类型	byte	1	$[-2^7, 2^7-1]$
	short	2	$[-2^{15}, 2^{15}-1]$
	int	4	$[-2^{31}, 2^{31}-1]$
	long	8	$[-2^{63}, 2^{63}-1]$
浮点类型	float	4	$[1.4\text{E}-45, 3.4028\text{E}+38]$
	double	8	$[4.9\text{E}-324, 1.7977\text{E}+308]$
字符类型	char	2	$[\text{"\u0000"}, \text{"\uffff"}]$
布尔类型	boolean	1	true, false

2) 对象数据类型

从广义上讲, Java 中每一个类都可以看做是一种数据类型, 这里统称为对象数据类型。如 Java 中处理字符串用类 String 表示。在例 1-2 中, main()方法括号中的 String args[] 表示 args 是一个字符串数组, 命令行的第一个参数、第二个参数……, 分别保存在 String(字符串)变量 args[0], args[1], …中, 它们都是 String 类型的数据。字符串“Hello, World!”也是 String 类型的数据。

与 8 种基本数据类型相对应, Java 中有 8 种相应的对象数据类型, 分别是 Byte、Short、Integer、Long、Float、Double、Character 和 Boolean。除以上这些对象数据类型外, 以后还



会学习更多的 Java 的类。

6. 问题与思考

- (1) 实例中如果输出的不是 $3>5$ ，而是 $3<5$ ，将是什么结果？
- (2) 把表 1-2 中对应的十进制数转换成二进制数、八进制数和十六进制数。

表 1-2 写出对应十进制数的二进制数、八进制数和十六进制数

十 进 制	二 进 制	八 进 制	十六进制
5			
7			
11			
25			

1.3 把 1、2、3 累加到变量

知识要点

- ☐ 常量和变量
- ☐ 赋值和初始化
- ☐ 数据类型转换

常量就是在程序运行期间不能被修改的量，变量简单地就是在程序运行期间可以修改的量。

赋值运算符是一个等号“=”。它在 Java 中的运算与在其他计算机语言中的运算一样，其通用格式为：

```
var = expression;
```

这里，变量 var 的类型必须与表达式 expression 的类型一致。

赋值运算符不同于数学中的等号，如语句 `sum = sum + 2;` 表示 sum 加上 2 后再赋给变量 sum，如果原来 sum 的值是 1，现在 sum 的值就成了 3。

典型实例 1、2、3 累加到变量。

1. 详细设计

本程序由类 Sum 实现，程序处理过程均在方法 main() 中完成。程序利用 Java 的赋值语句实现累加，代码框架如下：

```
class Sum{
    main(String args[]){
        定义变量 sum 并赋初值;
        sum ← sum + 1;
```