

软件工程师培养丛书

Java编程技术开发

武汉厚溥教育科技有限公司 编著



“理论→总结→上机→习题”四阶段教学模式

- ★ 理论结合实践，注重动手能力培养
- ★ 任务驱动讲解，有效激发学习兴趣
- ★ 典型项目案例，扎实培养专业素质
- ★ 教学做一体化，极大提高教学效率



清华大学出版社

软件工程师培养丛书

Java 编程技术开发

武汉厚溥教育科技有限公司 编著



清华大学出版社

北 京

内 容 简 介

本书按照高等院校、高职高专计算机课程基本要求,以案例驱动的形式来组织内容,突出计算机课程的实践性特点。本书共包含 12 个章节:Java 语言简介及基础、Java 语法基础、类和对象、继承与多态、包及抽象类和接口、Java 常用基础类、集合框架和泛型、异常和错误调试、Java IO 系统、JDBC(一)、JDBC(二)、JDBC(三)。

本书内容安排合理,层次清楚,通俗易懂,实例丰富,突出理论和实践的结合,可作为各类高等院校、高职高专及培训机构的教材,也可供广大程序设计人员参考。

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

Java 编程技术开发 / 武汉厚溥教育科技有限公司 编著. —北京:清华大学出版社, 2014
(软件工程师培养丛书)

ISBN 978-7-302-36154-1

I. ①J… II. ①武… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 072647 号

责任编辑:刘金喜

封面设计:崔东方

版式设计:妙思品位

责任校对:成凤进

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 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

课 件 下 载: <http://www.tup.com.cn>, 010-62794504

印 装 者:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:23.75

字 数:548 千字

版 次:2014 年 6 月第 1 版

印 次:2014 年 6 月第 1 次印刷

印 数:1~3000

定 价:39.80 元

产品编号:059441-01

编 委 会

主 任:

翁高飞

副主任:

王 鹏	余晓刚	刘 伟	曹 静
方风波	邹治伟	李建利	管胜波

委 员:

罗 炜	谢日星	吴金秀	胡智方
夏超群	陈 琴	夏 晶	彭 莉
徐 霞	明素华	王 敏	严 滔

前 言

Java 是一种可以撰写跨平台应用程序的面向对象的程序设计语言,是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台(即 Java SE, Java EE, Java ME)的总称。Java 技术具有卓越的通用性、高效性、平台移植性和安全性,广泛应用于个人 PC、数据中心、游戏控制台、科学超级计算机、移动电话和互联网,同时拥有全球最大的开发者专业社群。在全球云计算和移动互联网的产业环境下,Java 更具备了显著优势和广阔前景。

本书是“软件工程师培养丛书”中的一本专业课教材,该丛书是由武汉厚溥教育科技有限公司开发,以培养符合企业需求的软件工程师应用开发、实施为目标的 IT 职业教育丛书。在开发该丛书之前,我们对 IT 行业的岗位序列做了充分的调研,包括研究从业人员技术方向、项目经验和职业素质等方面的需求,通过对面向的学生特点、行业需求的现状以及实施等方面的详细分析,结合“厚溥”对软件人才培养模式的认知,按照软件专业总体定位要求,进行软件专业产品课程体系设计。该丛书集应用软件知识和多领域的实践项目于一体,着重培养学生的熟练度和规范性、集成和项目能力,从而达到预定的培养目标。

本书共包含 12 个章节:Java 语言简介及基础、Java 语法基础、类和对象、继承与多态、包及抽象类和接口、Java 常用基础类、集合框架和泛型、异常和错误测试、Java IO 系统、JDBC(一)、JDBC(二)、JDBC(三)。

我们对本书的编写体系做了精心的设计,按照“理论学习—知识总结—上机操作—课后习题”这一思路进行编排。“理论学习”部分描述了通过本案例要达到的



学习目的与涉及的相关知识点,使学习目标更加明确;“知识总结”部分概括了案例所涉及的知识点,使知识点完整系统地呈现;“上机操作”部分对案例进行了详尽分析,通过完整的步骤帮助读者快速掌握该案例的操作方法;“课后习题”部分帮助读者理解章节的知识点。本书在内容编写方面,力求细致全面;在文字叙述方面,注意言简意赅、重点突出;在案例选取方面,强调案例的针对性和实用性。

本书凝聚了编者多年来的教学经验和成果,可作为各类高等院校、高职高专及培训机构的教材,也可供广大程序设计人员参考。

本书PPT教学课件和案例源文件可通过<http://www.tupwk.com.cn/downpage>下载。

本书由武汉厚溥教育科技有限公司编著,由翁高飞、刘伟等多名企业实战项目经理编写。本书编者长期从事项目开发和教学实施,并且对当前高校的教学情况非常熟悉,在编写过程中充分考虑到不同学生的特点和需求,加强了项目实战方面的教学。本书编写过程中,得到了武汉厚溥教育科技有限公司各级领导的大力支持,在此对他们表示衷心的感谢。

参与本书编写的人员还有:武汉商学院曹静老师、荆州职业技术学院方风波老师、武汉工程职业技术学院邹治伟和夏超群老师、湖北三峡职业技术学院李建利老师、武汉软件工程职业学院谢日星老师、黄冈职业技术学院陈琴和夏晶老师、湖北国土资源职业学院胡智方老师等。

限于编写时间和编者的水平,书中难免存在不足之处,希望广大读者批评指正。

服务邮箱: wkservice@vip.163.com。

编 者

2014年2月

目 录

第1章 Java 语言简介及基础.....1

1.1 Java 语言简介.....2

1.1.1 Java 语言发展历史.....3

1.1.2 Java 平台简介.....3

1.2 Java 语言基础.....4

1.2.1 标识符.....4

1.2.2 基本数据类型.....4

1.2.3 条件语句 if-else.....5

1.2.4 嵌套 if 语句.....5

1.2.5 多重 if 语句.....5

1.2.6 分支结构之 switch.....7

【小结】.....9

【自测题】.....9

【上机部分】.....11

【课后作业】.....14

第2章 Java 语法基础.....15

2.1 循环结构.....16

2.1.1 while 循环.....16

2.1.2 do-while 循环.....17

2.1.3 for 循环.....18

2.1.4 嵌套循环.....19

2.1.5 break 语句.....20

2.1.6 continue 语句.....21

2.2 数组.....22

2.3 方法.....23

【小结】.....26

【自测题】.....26

【上机部分】.....28

【课后作业】.....32

第3章 类和对象.....35

3.1 面向对象编程.....36

3.2 类的定义.....36

3.2.1 类的声明.....37

3.2.2 类的属性.....37

3.2.3 类的行为.....38

3.3 对象的创建.....39

3.3.1 使用关键字 new 创建对象.....39

3.3.2 通过对象访问属性和行为.....40

3.3.3 属性的默认值.....43

3.4 构造方法.....44

3.4.1 无参数的构造方法.....44

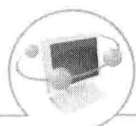
3.4.2 带参数的构造方法.....46



3.5 方法重载.....	49	5.1.2 使用包.....	88
3.6 this 关键字.....	50	5.2 抽象类.....	89
3.7 static 关键字.....	52	5.3 接口.....	93
【小结】.....	52	5.3.1 接口的概念.....	94
【自测题】.....	52	5.3.2 接口的用法.....	94
【上机部分】.....	54	5.3.3 接口与抽象类.....	97
【课后作业】.....	58	5.4 访问修饰符.....	99
第4章 继承与多态.....	59	5.4.1 类(接口)的访问修饰符.....	99
4.1 继承.....	60	5.4.2 方法及属性的访问修饰符.....	100
4.1.1 什么是继承.....	60	【小结】.....	104
4.1.2 为什么要继承.....	61	【自测题】.....	105
4.1.3 使用继承.....	62	【上机部分】.....	107
4.2 方法覆盖.....	64	【课后作业】.....	112
4.3 多态.....	65	【指导学习 I：面向接口编程】.....	112
4.4 super 关键字.....	66	第6章 Java 常用基础类.....	113
4.5 Object 类.....	71	6.1 概述.....	114
4.6 final 关键字.....	71	6.2 包装类.....	115
4.6.1 final 修饰类.....	72	6.3 Integer 整数类.....	116
4.6.2 final 修饰方法.....	72	6.3.1 属性.....	116
4.6.3 final 修饰属性和变量.....	73	6.3.2 构造方法.....	116
4.6.4 final 修饰对象引用.....	74	6.3.3 方法介绍.....	117
【小结】.....	75	6.4 Character 字符类.....	118
【自测题】.....	75	6.4.1 属性.....	118
【上机部分】.....	75	6.4.2 构造方法.....	118
【课后作业】.....	83	6.4.3 方法.....	118
第5章 包、抽象类和接口.....	85	6.5 String 字符串类.....	120
5.1 包.....	86	6.5.1 构造方法.....	121
5.1.1 创建包.....	86		



6.5.2 方法.....	121	7.5 HashMap 和 LinkedHashMap.....	160
6.5.3 字符串的不变性.....	125	7.6 Collections.....	163
6.6 StringBuffer 字符串缓冲类.....	126	7.7 Arrays.....	166
6.6.1 构造方法.....	126	7.8 泛型.....	167
6.6.2 方法.....	126	7.8.1 为什么要使用泛型程序设计.....	168
6.7 Random 随机类.....	131	7.8.2 泛型类的定义.....	168
6.7.1 构造方法.....	131	7.8.3 泛型方法.....	169
6.7.2 方法.....	131	7.8.4 类型变量的限定.....	170
6.7.3 关于 Random 类的说明.....	132	7.8.5 常见问题.....	171
6.8 Date 时间类.....	133	7.8.6 通配符类型.....	172
6.8.1 构造方法.....	133	【小结】.....	173
6.8.2 方法.....	133	【自测题】.....	173
6.9 Calendar 日历类.....	135	【上机部分】.....	174
6.9.1 字段.....	135	【课后作业】.....	186
6.9.2 方法.....	136	第 8 章 异常和错误调试.....	187
6.10 SimpleDateFormat 日历 格式化.....	138	8.1 异常.....	188
6.11 Math 算术运算类.....	139	8.2 异常的处理.....	190
【小结】.....	140	8.3 try-catch 结构.....	191
【自测题】.....	140	8.4 使用 finally 回收资源.....	193
【上机部分】.....	141	8.5 多重 catch 块.....	196
【课后作业】.....	146	8.6 嵌套 try-catch.....	198
第 7 章 集合框架和泛型.....	147	8.7 throw、throws 及自定义异常.....	200
7.1 集合框架.....	148	8.8 异常使用原则.....	204
7.2 ArrayList.....	151	【小结】.....	204
7.3 Iterator.....	155	【自测题】.....	204
7.4 LinkedList.....	157	【上机部分】.....	205
		【课后作业】.....	212



第9章 Java IO 系统.....	213
9.1 Java 的 IO 系统.....	214
9.2 File 类.....	216
9.3 FileInputStream 和 FileOutputStream.....	221
9.4 BufferedReader 和 BufferedWriter.....	226
9.5 Java 字符集.....	231
9.6 PrintWriter.....	234
【小结】	235
【自测题】	235
【上机部分】	236
【课后作业】	245
【指导学习 II: 三层架构】	246
第10章 JDBC(一).....	249
10.1 ODBC.....	250
10.2 JDBC.....	250
10.2.1 JDBC API.....	252
10.2.2 使用直连操作数据库.....	254
10.3 完成注册功能.....	261
【小结】	264
【自测题】	264
【上机部分】	265
【课后作业】	268
第11章 JDBC(二).....	269
11.1 带参数的 PreparedStatement	270
11.2 事务处理.....	274
11.3 批处理.....	276
11.4 调用存储过程.....	277
11.5 使用 properties 文件.....	279
【小结】	282
【自测题】	282
【上机部分】	283
【课后作业】	288
第12章 JDBC(三).....	289
12.1 数据库部分.....	290
12.2 逻辑实现.....	292
12.2.1 分析.....	292
12.2.2 分离数据库 连接信息 DBHelper.....	293
12.2.3 提供基础的 操作的父类 BaseDAO	294
12.2.4 用以封装数据的 Mail 实体类.....	296
12.2.5 继承了 BaseDAO 的 MailDAO 类.....	297
12.2.6 继承了 BaseDAO 的 UsersDAO 类.....	298
12.2.7 提供数据给显示层的 MailService 类.....	299
12.3 功能测试.....	301
12.3.1 测试发送邮件功能.....	301
12.3.2 测试显示邮件功能.....	302
【自测题】	303



【上机部分】	304	阅读III ODBC 数据源的创建及使用 ...	350
阅读 I Runtime Exception		阅读IV JDBC 性能大比拼	354
和 Checked Exception	331	测试 1 Statement	359
阅读 II 设计模式之 Decorator 模式	337	测试 2 PreparedStatement	361
设计模式	338	测试 3 PreparedStatement,	
Java IO 之装饰器模式	338	开启事务	362
新的输出流	342	测试 4 PreparedStatement,	
实现自己的装饰器	344	使用批处理(开启事务)	363

第1章

Java语言简介及基础



课程目标

- ▶ Java 简介
- ▶ Java 历史及版本
- ▶ Java 语法基础及分支结构



简介

在 Java 基础课程中,我们学到了如何编写 Java 程序,理解了分支结构、循环结构,并能通过编写方法及使用数组等高级知识,完成较复杂程序的编写。具体如下:

- Eclipse 的基本用法
- Java 的数据类型:基本数据类型(8 种)、复杂数据类型(3 种)
- 赋值运算符、算术运算符、关系运算符、逻辑运算符、条件运算符
- 分支结构: if 语句、switch 语句
- 循环结构: do-while 循环、while 循环、for 循环
- 数组:一维数组与二维数组的初始化与使用
- 为什么要用方法,方法的定义与返回值,方法的参数

这部分内容是整个编程体系的基础,为我们学习后续知识打下深厚的基础。如果说第一阶段的内容是搭建好了火箭的话,那么接下来的工作,就是要为它填充燃料了。而事实上,Java 是一种纯面向对象的语言,一个 Java 程序内的一切其实都是对象。因此,用 Java 进行程序设计,必须将自己的思想转入到一个面向对象的世界,以面向对象的思维方式来思考问题。如果说面向过程符合电脑顺序工作的“思维”方式,那么面向对象则更符合人脑的思维方式,因此面向对象程序设计并不难学习和掌握。那么什么叫面向对象呢?它有什么好处?如何利用面向对象的思想来编写程序呢?这就是本章甚至本书要解决的问题,从本章开始,我们将逐步接触到面向对象编程,学习最基本的面向对象语法。其中有几个重要的概念:什么叫类,什么叫对象,什么叫构造方法。

在本书中,我们将会学到 Java 的面向对象知识,内容涉及 Java 的面向对象、类的属性和方法、封装继承与多态、抽象类与接口、异常、Java 的常用包和类,以及数据库连接 JDBC 等内容。

首先我们回顾一下 Java 基础中的主要内容。

1.1 Java 语言简介

Java,是一种可以编写跨平台应用程序的面向对象的程序设计语言,由 Sun Microsystems(太阳微电子)公司的詹姆斯·高斯林(James Gosling)等人于 20 世纪 90 年代初开发。它最初被命名为 Oak,作为一种小家用电器的编程语言,用于解决诸如电视机、电话、闹钟、烤面包机等家用电器的控制和通信问题。由于这些智能化家电的市场需求没有预期的高,Sun 放弃了该项计划。就在 Oak 几近失败之时,随着 Internet 的发展,Sun 看到了 Oak 在计算机网络上的广阔应用前景,于是改造了 Oak,在 1995 年 5 月以“Java”的名称正式发布了。

目前,Java 已得到了工业界的认可,广泛地使用在桌面、企业应用、移动设备以及互



联网应用程序中,许多大公司包括 IBM, Apple, DEC, Adobe, HP, Oracle, Toshiba 等都购买了 Java 许可证,将自己的解决方案建立在 Java 技术之上。Java 已经成为最为流行的企业级应用开发工具,Java 语言已经成为最流行的网络编程语言。

1.1.1 Java 语言发展历史

1995 年 5 月 23 日,Java 语言诞生。

1996 年 1 月,第一个 JDK——JDK 1.0 诞生。

1996 年 4 月,10 个最主要的操作系统供应商申明将在其产品中嵌入 Java 技术。

1997 年 2 月 18 日,JDK 1.1 发布。

1998 年 12 月 8 日,Java 2 企业平台 J2EE 发布。

1999 年 6 月,SUN 公司发布 Java 的三个版本:标准版、企业版和微型版(J2SE、J2EE、J2ME)。

2000 年 5 月 8 日,JDK 1.3 发布。

2000 年 5 月 29 日,JDK 1.4 发布。

2002 年 2 月 26 日,J2SE 1.4 发布,自此 Java 的计算能力有了大幅提升。

2004 年 9 月 30 日 J2SE 1.5 发布,是 Java 语言发展史上的又一里程碑事件。为了显示这个版本的重要性,J2SE 1.5 更名为 J2SE 5.0。

2006 年 12 月,SUN 公司发布 JDK 6.0。

2011 年 7 月,JDK 7.0 发布。

1.1.2 Java 平台简介

1999 年 Sun 根据应用的领域不同把 Java 技术划归为三个平台,以满足不同类型应用程序的开发,当时分别称为 J2SE、J2EE 和 J2ME,现在改名为 Java SE、Java EE 和 Java ME。

(1) Java SE: Java SE 是 Java 平台标准版的简称(Java Platform, Standard Edition),用于开发和部署桌面、服务器以及嵌入设备和实时环境中的 Java 应用程序。Java SE 包括用于开发 Java Web 服务的类库,同时,Java SE 为 Java EE 提供了基础。

(2) Java EE: Java EE 是 Java 平台企业版的简称(Java Platform, Enterprise Edition),用于开发便于组装、健壮、可扩展、安全的服务器端 Java 应用。Java EE 建立于 Java SE 之上,具有 Web 服务、组件模型以及通信 API 等特性,这些为面向服务的架构(SOA)以及开发 Web 2.0 应用提供了支持。Java EE 基于 Java SE,此外新加了企业应用所需的类库。

(3) Java ME: Java ME 是一种以广泛消费性产品为目标的高度优化的 Java 运行时环境,包括寻呼机、移动电话、可视电话、数字机顶盒和汽车导航系统。Java ME 为小型设备带来了 Java 语言的跨平台功能,允许移动无线设备共享应用程序。有了 Java ME, Sun 已经使 Java 平台能够将桌面应用、无线设备、企业应用融合起来。



1.2 Java 语言基础

还记得怎样在安装和配置好 JVM 后使用 Eclipse 编写 Java 程序吧？

示例 1.1:

```
/*
 * 这是注释部分，多行文档注释
 * 第一个 Java 程序
 * @author hopeful
 */
public class Hello {
    public static void main(String[] args) {
        //这是一个 main 方法
        System.out.println("Hello World!");
    }
}
```

1.2.1 标识符

用来标识类名、变量名、方法名、类型名、数组名、文件名的有效字符序列称为标识符，简单地说，标识符就是一个名字。

Java 语言规定标识符由字母、下划线、美元符号和数字组成，并且第一个字符不能是数字。下列都是合法的标识符。

```
Girl_$, www_12$, $23boy
```

标识符中的字母是区分大小写的，Boy 和 boy 是不同的标识符。

1.2.2 基本数据类型

Java 是典型的强类型语言，即必须声明每个变量的类型。Java 定义了 8 个基本数据类型：字节型(byte)、短整型(short)、整型(int)、长整型(long)、字符型(char)、浮点型(float)、双精度型(double)、布尔型(Boolean)、这些类型可分为 4 组。

- 整数：该组包括字节型(byte)、短整型(short)、整型(int)、长整型(long)、它们代表有符号整数。
- 浮点型数：该组包括浮点型(float)、双精度型(double)，它们代表有小数精度要求的数字。
- 字符：这个组包括字符型(char)，它代表字符集的符号，例如字母和数字。
- 布尔型：这个组包括布尔型(Boolean)，它是一种特殊的类型，表示真/假值。



1.2.3 条件语句 if-else

在 Java 语言中，最简单的分支结构就是 if 语句。if 语句将根据条件真或假，判断是否执行 if 的从属语句。相对的 if-else 的使用很频繁，if-else 结构根据一个布尔值的真假来选择做不同的事情，即“如果……否则……”。其实 if-else 只不过是条件运算符的完整写法而已。它的流程图如图 1-1 所示。

例如：如果 number 的值比 5 大就欢迎，否则就不欢迎！

示例 1.2:

```
/*
 * 测试: number 的值决定输出
 * @author hopeful
 */
public class Test{
    public static void main(String []args){
        int number;
        number = 6;
        if(number > 5){
            System.out.println("Hello !");
        }else{
            System.out.println ("Bay-Bay!");
        }
    }
}
```

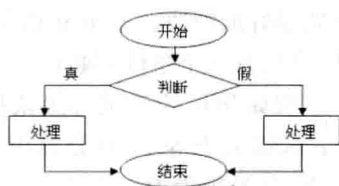


图 1-1 if 语句流程图

1.2.4 嵌套 if 语句

嵌套 if 语句可以在条件内，针对真或假的情况，再指定条件进行判断，从而拥有执行更多分支的功能。嵌套 if 语句示意图如图 1-2 所示。

示意图 1-2 只指定了外层判断为真时内层嵌套 if 的情况，其实，在图中①至③的地方，均可替换为条件判断语句，从而实现更深层次的条件嵌套。

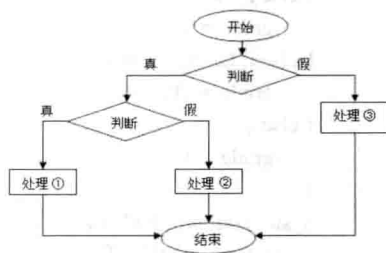
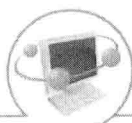


图 1-2 嵌套 if-else 语句流程图

1.2.5 多重 if 语句

在某些情况下，我们需要对一系列对等的条件进行判断，从而决定采用什么样的解决



办法。例如，对于学员的成绩来说，不同的分数对应不同的等级。对于这一类问题来说，使用之前讲述的 if 语句很难解决，或写法不简洁，可读性较差，但使用多重 if 语句将能很好地解决。多重 if 语句的示意图大致如图 1-3 所示（假设只有三个条件判断）。

假如学员的考试等级依据考试成绩来判断，规则如下：成绩大于等于 90 分等级为 A，成绩大于等于 80 分小于 90 分等级为 B，成绩大于等于 70 分小于 80 分等级为 C，成绩大于等于 60 和小于 70 等级为 D，成绩小于 60 分等级为 E。下面的程序首先要求学员输入成绩，然后输出对应的等级。

示例 1.3:

```
/*
 * 根据分数计算等级
 * @author: hopeful
 */
import java.util.Scanner;

public class GradeTest {
    public static void main(String[] args) {
        int score; //分数
        char grade; //等级
        Scanner scanner = new Scanner(System.in);
        System.out.println("请输入分数: ");
        score = scanner.nextInt();
        if (score >= 90) {
            grade = 'A';
        } else if (score >= 80) { //①
            grade = 'B';
        } else if (score >= 70) { //②
            grade = 'C';
        } else if (score >= 60) {
            grade = 'D';
        } else {
            grade = 'E';
        }
        System.out.println("分数为: " + score);
        System.out.println("等级为: " + grade);
    }
}
```

运行结果如图 1-4 所示。

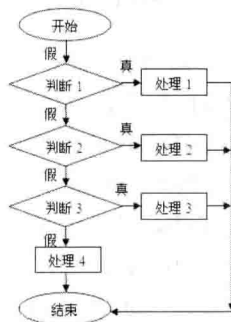


图 1-3 多重 if 语句流程图