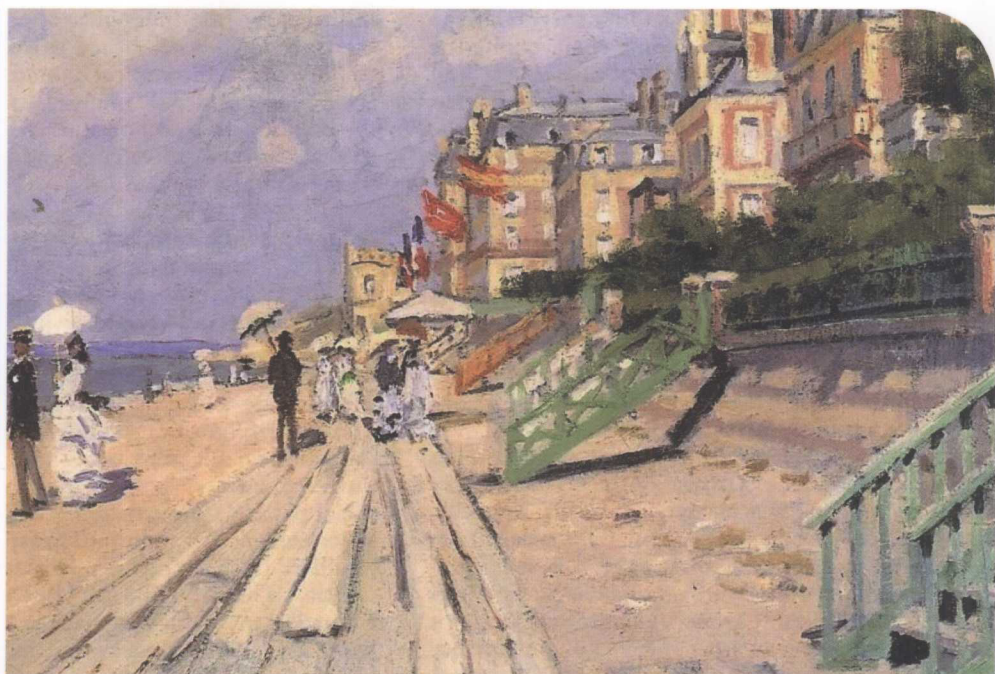




21世纪高等学校计算机
专业实用规划教材

Java 程序设计

◎ 湛卫军 王浩娟 编著

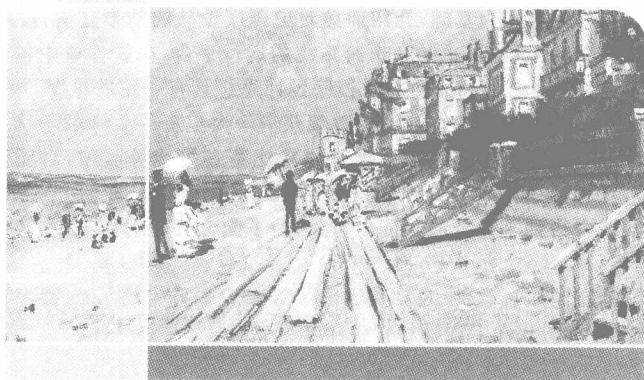


清华大学出版社





21世纪高等学校计算机
专业实用规划教材



Java 程序设计

◎ 湛卫军 王浩娟 编著

清华大学出版社
北京

内 容 简 介

本书是清华大学全校性选修课程“Java 程序设计”的教材,主要介绍 Java 语言的基本概念和相关知识,包括 Java 语言的语法、面向对象编程、异常处理、输入与输出、图形用户界面、多线程编程、网络编程、对象集合和 Android 编程等内容,并以此为基础,介绍面向对象程序设计的基本方法、技术和理论。

众所周知,Java 既是一种编程语言,也是一个跨系统的运行平台,在软件业中得到了广泛应用,已成为众多程序员的首选语言。但 Java 语言和面向对象程序设计思想具有一定的难度,理解起来有一定的困难,学生不易掌握。而本书作者具有丰富的教学经验,曾荣获清华大学青年教师教学优秀奖、霍英东教育基金会高等院校青年教师奖、北京市高校青年教师教学基本功比赛一等奖等奖项,对教学规律、课程内容和学生特点有着深入的理解。因此,在写作本书时,能够以生动活泼、通俗易懂的方式来讲述复杂的原理和概念,这一点已经在教学实践中得到了充分的验证。另外,为了配合课程的内容,在章末附有相应的习题,以便读者及时复习相关内容。

本书适合作为高等院校计算机、软件、自动化和电子等专业本科生的 Java 程序设计教材,也适合正在学习 Java 语言的广大科技人员、软件工程师和青少年学生参考。

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 程序设计/湛卫军等编著. —北京:清华大学出版社,2016

21 世纪高等学校计算机专业实用规划教材

ISBN 978-7-302-43217-3

I. ①J… II. ①湛… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2016)第 041729 号

责任编辑:魏江江 王冰飞

封面设计:刘 键

责任校对:梁 毅

责任印制:杨 艳

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>, 010-62795954

印 装 者:河北新华第一印刷有限责任公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:16.5

字 数:411 千字

版 次:2016 年 7 月第 1 版

印 次:2016 年 7 月第 1 次印刷

印 数:1~2000

定 价:34.00 元

产品编号:067968-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机专业课程领域,以专业基础课为主、专业课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 反映计算机学科的最新发展,总结近年来计算机专业教学的最新成果。内容先进,充分吸收国外先进成果和理念。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,融合先进的教学思想、方法和手段,体现科学性、先进性和系统性,强调对学生实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。专业基础课和专业课教材配套,同一门课程有针对不同层次、面向不同应用的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选。在制定教材规划时要依靠各课程专家在调查研究本课程教

材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机专业实用规划教材

联系人: 魏江江 weijj@tup.tsinghua.edu.cn

前言

“Java 程序设计”是清华大学计算机系开设的一门全校性选修课，主要面向全校对 Java 编程感兴趣的本科生。课程主要介绍 Java 语言的基本概念和相关知识，并以此为基础，介绍面向对象程序设计的基本方法、技术和理论。为了配合课程的讲授，便于学生的学习，我们编写了本教材。

众所周知，Java 既是一种编程语言，也是一个跨系统的运行平台，在软件业中得到了广泛应用，已成为众多程序员的首选语言。事实上，在 TIOBE 指数历年的排名榜中，Java 语言始终名列前茅，并在大部分时间内排名第一。尤其是随着 Android 手机的爆炸式增长，Java 语言又出现了新的增长点，因为 Android 系统上的应用软件都是用 Java 语言开发的。因此，Java 语言的普及和推广具有广泛的市场需求。然而，由于 Java 是一种面向对象的编程语言，在很多内容上具有一定的难度，如抽象与封装、方法重载与重写、数据的存储、继承、多态、动态绑定、抽象类与接口、对象集合等，这些内容单调枯燥、晦涩难懂，学生在学习时会面临一些困难。而本书作者具有丰富的教学经验，曾经荣获清华大学青年教师教学优秀奖、霍英东教育基金会高等院校青年教师奖、宝钢优秀教师奖、北京市高校青年教师教学基本功比赛一等奖、清华大学清韵烛光第一届“我最喜爱的教师”评选活动“十佳教师”等奖项和荣誉，对教学规律、课程内容和学生特点有着深入的理解。因此，在写作本书时，能够以生动活泼、通俗易懂的方式来讲述复杂的原理和概念，这一点已经在教学实践中得到了充分的验证。

本书共分 10 章。第 1 章是 Java 语言概述，介绍基本的 Java 程序结构；第 2 章是 Java 语言基础，介绍 Java 语言的基本语法知识；第 3 章介绍 Java 面向对象编程的核心原理，包括类、对象、访问控制、重载、继承和多态等；第 4 章介绍异常处理、输入/输出和文件的访问；第 5 章介绍 Java 语言的图形用户界面编程；第 6 章介绍 Java 线程，包括进程与线程的基本原理、线程间的数据共享、线程间的互斥与同步、线程调度与优先级等；第 7 章介绍网络编程，主要是基于 TCP 和 UDP 的 Socket 编程；第 8 章介绍 Java 语言的对象集合，包括 List、Set 和 Map 等内容；第 9 章介绍 Android 编程，并通过一个案例介绍了一个典型的 Android 应用程序的开发过程；第 10 章介绍面向对象软件开发，包

括团队建设、软件过程、软件需求、面向对象分析与设计等内容。

在本课程的筹备过程中,搜集了大量的相关材料,包括国内外大学的课程讲义、Java 编程书籍、互联网上的素材等,这些材料使我们受益匪浅。

在本书的写作过程中,得到了许多人的关心和帮助,清华大学计算机系的郑莉老师最早提出了本书的构想,清华大学出版社的魏江江老师积极联系本书的出版事宜,笔者的家人在背后默默支持我们的工作,在此一并表示感谢。

湛卫军

2016 年 1 月于清华园

目录

第 1 章 Java 语言概述	1
1.1 Java 语言简介	1
1.1.1 Java 语言的历史	1
1.1.2 Java 语言的特点	4
1.1.3 Java 语言的应用领域	5
1.2 Java 程序结构	7
1.2.1 Java 程序的类型	7
1.2.2 一个简单的 Java 程序	7
1.2.3 Java 运行环境	9
1.2.4 Java 开发环境	11
1.3 练习题	13
第 2 章 Java 语言基础	14
2.1 数据类型	14
2.1.1 数据类型的基本概念	14
2.1.2 整数类型	15
2.1.3 实数类型	17
2.1.4 字符类型	18
2.1.5 布尔类型	18
2.2 常量	18
2.2.1 整型常量	18
2.2.2 实型常量	19
2.2.3 字符常量	19
2.3 变量	20
2.3.1 变量的基本概念	20
2.3.2 变量的命名	20
2.3.3 变量的定义	21

2.3.4	变量的初始化	21
2.4	运算符与表达式	21
2.4.1	算术运算符和算术表达式	22
2.4.2	赋值运算符和赋值表达式	23
2.4.3	类型转换	24
2.5	选择结构	26
2.5.1	关系运算符和关系表达式	26
2.5.2	逻辑运算符和逻辑表达式	27
2.5.3	if 语句	28
2.5.4	switch 语句	31
2.6	循环控制语句	32
2.6.1	for 语句	33
2.6.2	while 语句	33
2.6.3	do...while 语句	34
2.6.4	break 和 continue 语句	35
2.7	数组	36
2.7.1	一维数组的定义和使用	36
2.7.2	二维数组的定义和使用	39
2.8	练习题	40
第 3 章	Java 面向对象编程	42
3.1	面向对象概述	42
3.1.1	软件系统的复杂性	42
3.1.2	结构化方法与面向对象方法	43
3.1.3	对象模型	44
3.2	类和对象	46
3.2.1	对象	46
3.2.2	类	47
3.2.3	类的定义	48
3.2.4	类的使用	53
3.2.5	成员方法	54
3.2.6	引用类型	57
3.2.7	静态类型	60
3.3	访问控制与方法重载	62
3.3.1	访问控制	62
3.3.2	函数重载	65
3.3.3	存储管理	68
3.3.4	this 关键词	69
3.4	继承与多态	71

3.4.1	继承关系	72
3.4.2	子类对象的存储	76
3.4.3	构造函数	78
3.4.4	类的继承举例	80
3.4.5	多态	82
3.4.6	函数调用绑定	86
3.4.7	抽象类和抽象函数	89
3.4.8	接口	90
3.5	练习题	92
第4章	异常处理与输入/输出	95
4.1	异常处理	95
4.1.1	为何引入异常处理	95
4.1.2	什么是异常	97
4.1.3	try...catch	98
4.1.4	throw	101
4.2	输入/输出	103
4.2.1	输入/输出概述	103
4.2.2	字节流与字符流	103
4.2.3	输入数据	104
4.3	文件	108
4.3.1	文件的基本概念	108
4.3.2	访问文件属性	110
4.3.3	读文本文件	111
4.3.4	写文本文件	114
4.4	练习题	114
第5章	图形用户界面	117
5.1	概述	117
5.2	Graphics 绘图	118
5.2.1	绘图环境	118
5.2.2	颜色	119
5.2.3	字体	119
5.2.4	Graphics 类	120
5.3	Swing 组件	125
5.3.1	Swing 概述	125
5.3.2	容器	126
5.3.3	基本控件	128
5.3.4	布局管理器	130

5.3.5 事件处理	134
5.4 Swing 举例	138
5.5 练习题	147
第 6 章 线程	148
6.1 进程与线程	148
6.1.1 进程	148
6.1.2 线程	152
6.2 Java 线程	153
6.2.1 Thread 类	153
6.2.2 Runnable 接口	155
6.3 线程间的数据共享	156
6.4 线程间的互斥与同步	157
6.4.1 线程间的互斥	157
6.4.2 线程间的同步	161
6.5 线程调度与优先级	164
6.6 练习题	166
第 7 章 网络编程	168
7.1 网络基础知识	168
7.1.1 网络概述	168
7.1.2 网络模型	169
7.1.3 网络协议的类型	170
7.1.4 IP 地址	172
7.2 基于 URL 的网络编程	174
7.3 基于 TCP 的 Socket 编程	177
7.3.1 Socket 概述	177
7.3.2 客户/服务器模型	177
7.3.3 Socket 编程	179
7.4 基于 UDP 的 Socket 编程	185
7.4.1 UDP 的相关类	185
7.4.2 UDP 报文的发送和接收	186
7.5 练习题	188
第 8 章 对象集合	189
8.1 Java 集合框架	189
8.1.1 对象数组	189
8.1.2 集合	190
8.1.3 Iterator 接口	191

8.1.4	Iterable 接口	192
8.2	List	193
8.2.1	List 接口	193
8.2.2	ArrayList 与 LinkedList	194
8.2.3	泛型	196
8.2.4	ArrayList 与 LinkedList 的性能比较	198
8.3	Set	200
8.3.1	Set 接口	200
8.3.2	HashSet	201
8.3.3	HashTable	203
8.3.4	TreeSet	204
8.4	Map	205
8.4.1	Map 接口	205
8.4.2	HashMap	206
8.4.3	TreeMap	207
8.5	练习题	208
第 9 章	Android 编程	210
9.1	移动应用开发	210
9.2	Android 系统	211
9.2.1	运行环境	211
9.2.2	Android 系统的发展历史	212
9.2.3	Android 系统的体系结构	212
9.3	创建开发环境	213
9.4	一个 Android 程序	216
9.4.1	创建一个空白项目	217
9.4.2	用户界面设计	219
9.4.3	BMI 用户界面设计	221
9.4.4	strings.xml	223
9.4.5	BMI 程序主体	224
9.5	练习题	227
第 10 章	面向对象软件开发	228
10.1	团队	228
10.1.1	团队开发	228
10.1.2	团队的组建	229
10.2	软件过程	230
10.2.1	什么是软件过程	230
10.2.2	软件过程模型	230

10.3	软件需求	232
10.3.1	什么是软件需求	232
10.3.2	需求获取	232
10.3.3	用例建模方法	233
10.4	面向对象分析与设计	235
10.4.1	概述	235
10.4.2	UML	236
10.4.3	类图	237
10.4.4	面向对象分析	243
10.4.5	面向对象分析举例	245
10.5	练习题	248
参考文献		249

Java 语言概述

第 1 章

1.1 Java 语言简介

一天,一位语言学家和一位计算机科学家聊天。

语言学家说:我精通十门语言——英语、法语、西班牙语、俄语、阿拉伯语、汉语、德语、日语、葡萄牙语和印地语。

计算机科学家说:这算什么!我也精通十门语言——汉语、英语、Fortran、Basic、C 语言、C++、Java、C#、Perl、Python。

当然,这只是一个玩笑。计算机科学家精通的多是编程语言,而不是人类的自然语言。

这个玩笑所表示的另一层含义是计算机编程语言种类繁多。事实上,自从 1954 年世界上第一种高级语言 Fortran 问世以来,人类已经发明了数以千计的编程语言,但只有很少一部分得到了广泛的应用。

TIOBE(The Importance Of Being Earnest)公司(<http://www.tiobe.com>)是一家专注于软件质量评估和监测的公司,著名的 TIOBE 指数就是由该公司推出并维护的。这个指数将编程语言的排名以列表的形式呈现出来,并且每个月更新一次,用来表示各种编程语言当前的流行程度。在该指数历年的排行榜中,Java 语言始终名列前茅,在绝大部分时间内都排名第一。2013 年它曾经短暂离开了榜首的位置,被 C 语言所取代,屈居第二。但时隔一年半之后,它又重新回到了榜单首位。由此可见,Java 语言是一门非常重要的编程语言,在软件业有着广泛的应用。

1.1.1 Java 语言的历史

Java 语言的产生,实际上是“有心栽花花不开,无心插柳柳成荫”。20 世纪 90 年代,美国 Sun 公司正处于蒸蒸日上的发展阶段,该公司主要研发工作站和服务器,在互联网泡沫时期获得了迅猛发展,成立仅十余年,就一跃成为财富 500 强公司。但好景不长,互联网泡沫破碎后,公司业务迅速下滑,股价

暴跌,最终于 2009 年被 Oracle 公司收购。1991 年,尚处于扩张期的 Sun 公司试图开发消费电子领域中的商机,因而成立了一个项目组,名为 Green,成员包括 James Gosling、Mike Sheridan 和 Patrick Naughton 等。该项目组的基本思路是向消费电子厂商出售现代软件技术。换言之,它不打算从事硬件方面的研发,硬件还是由各厂商自己来做,项目组只进行软件方面的研发,即基于现代软件技术,开发高水平的软件系统产品,运行在各厂商的硬件平台上,然后销售给终端用户。这是 Green 项目组成立时所制定的经营策略和赢利模式。

所谓消费电子,就是围绕着终端消费者的应用而设计的各种电子类产品,如手机、音频播放器、视频播放器、电视机顶盒、平板电脑等。消费类电子产品具有如下特点。

- 功耗限制。对于手机、平板电脑等随身携带的设备,一般都由电池供电,因此功耗是一个很大的问题。
- 资源有限。此类设备的硬件配置,如处理器、存储器和 I/O 设备等,一般都稍低。诚然,随着硬件技术的发展,目前处理器和存储器等都已经比较高端了,但 I/O 设备还是不及台式机方便。
- 用户界面友好。此类设备最好采用图形用户界面,并且易学易用。因为终端用户主要是普通大众,并不一定是计算机高手,所以复杂的操作方式并不适合。
- 设备种类繁多,功能各不相同。

在这种情形下,对于初次进入消费电子领域、毫无任何相关经验的 Green 项目组,该如何开展工作呢?该选择什么事情作为切入点呢?据说 19 世纪美国西部淘金热期间,真正靠淘金发财的人没有几个,而那些卖淘金工具的人却发财了。Green 项目组采用了类似的思路,不是去做具体的应用软件,而是做通用的工具软件。具体来说,对于消费电子厂商生产出来的硬件产品,它最需要的软件有两种:操作系统和编程语言。操作系统是必须要有的,需要它管理底层的所有硬件,并向上提供应用程序编程接口(Application Programming Interface,API)。没有操作系统的硬件,什么也做不了。此外,编程语言也很重要,因为任何一个应用软件都需要使用某种编程语言来实现。总之,对于任何一种硬件来说,操作系统和编程语言是它最需要的、必不可少的工具软件。因此,Green 项目组就把工作的切入点放在这两种软件的开发上。这是很聪明的做法,因为一旦成功,那么无论是哪一家硬件厂商,无论开发什么样的应用软件,都离不开这两种工具软件。这样,无论厂商做的硬件和应用开发人员做的应用软件是否成功(即淘金者发财),工具软件都能成功(即卖淘金工具的人发财)。事实上,已经有人在普通台式机上做类似的事情并获得成功,即著名的微软公司,现在 Green 项目组想把它移植到消费电子领域。

随后项目组开始工作,将想法变为现实。操作系统问题不大,因为 Sun 公司也是做操作系统的,Solaris 操作系统就是它的产品。那编程语言呢?项目组开始的想法是利用现有的面向对象语言 C++,然后加以改进,开玩笑地称为 C++++--语言。意思是:C++是从 C 语言派生出来的,是在 C 语言的基础上增加了面向对象的内容,于是从 C 变成了 C++;他们想对 C++ 进行一些改进,先增加一些功能,再删减一些功能,也就是在 C++ 的基础上,先++,再--,因此称为 C++++--语言。但后来发现,这种方式还是不能解决问题,不契合消费电子类设备的特点。例如,C++ 占用内存较大,而且内存管理比较麻烦,对程序员的要求较高,在动态申请内存空间后必须要记得释放,否则就会出现内存泄漏的问题。再如,C++ 的可移植性也是一个问题。由于电子设备种类繁多,所以经常需要把应用软件从一个平台移

植到另一个平台。如果是使用 C++ 编写的程序,那么移植时需要做的事情就是源代码不变,然后新的平台上,用新的编译器重新编译。但 Green 项目组的人觉得,重新编译也是需要额外做的事情,还是不太方便,他们希望的是不用做任何事情,就能直接移植过去。所以后来就放弃了对 C++ 进行改进的想法,而是打算从头开发一种新的编程语言,即 Oak 语言。Oak 是橡树的意思,这个名字的由来,是因为在项目组办公室的窗外有一棵橡树,抬头可见,所以就起了这样一个名字。

在任务明确后,Green 项目组就启动了实质性的开发工作,项目开发持续了一年半时间,然后要进行项目验收,要出一个阶段性的成果。既然要验收,就需要一个载体,有一个看得见、摸得着的东西给别人演示。为此,他们就做了一个掌上电脑,又称个人数字助理(Personal Digital Assistant, PDA),然后把所有的工作都集成在里面,包括一个操作系统(Green OS)、一种编程语言(Oak)和一个图形用户界面。在短短一年半时间内做了这么多事情,非常不容易,技术上也比较先进。后来 Sun 公司觉得这个项目比较有前途,就想把它商业化,于是在项目组的基础上成立了一家子公司 FirstPerson,开始进行市场推广,想把这项技术销售出去。它投标了时代华纳公司的一个机顶盒项目。从技术上来说,其系统非常适合这个机顶盒项目,因为机顶盒就是一台小型电脑,以电视屏幕为显示器,以遥控器为键盘,用户可以通过遥控器与系统交互,如视频点播、回放、录像等,这种交互式体验比单纯被动地看电视要好得多。在具体实现时,机顶盒只是一个硬件,没有软件,所以需要操作系统、图形用户界面和应用软件的支持,而这些正是 FirstPerson 公司的强项。很不幸,这次投标没有成功。因此,商业化尝试没有成功,没有申请到项目,也就没有资金支持,最后 FirstPerson 公司被撤销了。

到 1994 年,互联网发展迅猛,Green 项目组负责人 James Gosling 决定把应用转向互联网。如前所述,他们原本的想法是在有线电视网络上做一些事情。有线电视网络的情形是各个家庭的电视机连着机顶盒,然后机顶盒连接到有线电视网络。实际上,互联网也是类似的——计算机和计算机通过互联网连接在一起。事实上,未来的发展趋势可能是三网(电信网、广播电视网和互联网)合一。因此,Gosling 想把原来在有线电视网络上做的工作转移到互联网上。具体如何操作呢?他们的初步想法是做浏览器。因为当时虽然已经有一些浏览器(当时最有名的浏览器是 Netscape),但当时的网页都是静态网页,就像一张画纸,其内容是不能变化的。所以他们想做一个浏览器,可以承载动态内容,也可以和用户交互。而这就需要编程,就需要编程语言,这是他们工作的切入点。后来项目组把 Oak 语言更名为 Java 语言,因为 Oak 只是刚开始随意取的名字,后来查了一下,发现 Oak 已经被人注册了。取名 Java 是因为他们平常最喜欢喝的一种咖啡产自爪哇(Java),爪哇是印尼的一个岛屿。他们用 Java 语言编写了一个浏览器 HotJava,能够浏览动态网页,也能够执行 Applet 程序。但后来这个浏览器并没有流行开来,因为当时比较流行的浏览器都通过插件的方式集成了这项功能。1996 年,Sun 公司成立了一家子公司 Javasoft 来发展 Java。

从 Java 语言的发展历史看,1995 年,Sun 公司发布了 Java 语言的第一个公开版本 Java 1.0。它号称“只需写一次,便可四处运行”。换言之,它具有极好的兼容性,不存在移植问题,也不需要重新编译。当时的主流浏览器很快都集成了相应插件,可以运行带有 Java 小应用程序的网页,因此 Java 语言很快就流行起来。1998 年,Sun 公司发布了 Java 2,它具有多重配置,可用于不同类型的平台。例如,J2EE 为企业应用,J2ME 为移动应用,J2SE 为标

准版本。

2006 年 11 月, Sun 公司开始着手将 Java 语言开源化, 大部分内容都是免费的, 而且源代码公开。到 2007 年 5 月, Java 的所有源代码都已对外公开。

时至今日, Java 已经成为一门极其重要的编程语言, 在软件业得到了广泛的应用。尤其是在基于客户/服务器的 Web 应用开发中, Java 是最受欢迎的编程语言之一。根据维基百科的数据, 全球大约有 900 万 Java 程序员, 并形成了一个庞大的 Java 开发社区; 每年有 9.3 亿份 Java 运行环境被下载; 全世界有 30 亿部手机运行 Java。这些数据都说明了 Java 语言的流行程度。

1.1.2 Java 语言的特点

Java 之所以获得如此广泛的应用, 与它的特点是密切相关的。

首先, Java 是一种面向对象的编程语言。对于一种编程语言来说, 一个重要的评价指标是看它解决复杂问题、编写复杂系统的能力。对于结构化编程语言, 如 C 语言, 编程时主要是通过模块分解的方法, 把大问题分解为小问题。而对于面向对象编程语言, 如 Java 语言, 则采用另一种分解方法编程, 即把软件系统分解为一个个的类和对象, 然后使用抽象和封装机制, 将对象的本质行为和它的具体实现分开(即将类的对外接口和内部实现分开)。此外, 还有模块化和层次结构, 实现不同层次的抽象。这些特点使面向对象编程语言非常适合编写大规模、复杂的软件系统。

其次, Java 具有可靠性和安全性的特点。Java 的初衷是用于消费电子领域, 因此非常关注语言的可靠性和安全性。它是一种强类型语言, 即在编译和运行时会进行大量的类型检查, 以防止发生数据类型不匹配的问题, 避免有出错倾向的不安全状态的出现。此外, Java 不允许使用指针。指针的优点是灵活, 可以方便地访问任意的内存单元, 但这对程序员提出了较高的要求。因为使用指针时稍有不慎, 就可能导致内存数据的紊乱。对于这一点, 相信那些编写过 C 语言程序, 并曾经花费几个小时来检查由于指针错误而引起内存冲突的人, 一定深有体会。另外, 在动态内存分配上, 程序员经常犯的错误是只记得申请, 却忘了释放。或者反过来, 在一处申请, 却在多处释放(拜指针所赐)。这些操作都会带来内存状态的紊乱, 而 Java 语言很好地解决了这个问题。它设计有自动收集垃圾功能, 程序员只需要申请及使用内存空间, 并不需要释放空间。如果一块空间不再被使用, 那么 Java 就会自动回收。这样就有效防止了内存分配的错误, 使程序员从烦琐的内存管理中解脱了出来。

再次, Java 是一种跨平台语言, 具有非常好的可移植性。Java 软件可以运行在各种不同类型的硬件平台上, 从 PC、超级计算机到手机、游戏机和平板电脑。这是 Java 最重要的特性之一, 也是它得到广泛应用的主要原因。如前所述, Java 的设计思想是“只需写一次, 便可四处运行”。如图 1.1 所示, 一个 Java 源程序(source code)经过编译后, 会转换为字节码(byte code), 这件事情只需要做一次。在运行时, 字节码可以放在不同机器上运行, 因为每一台机器上都有自己的 Java 虚拟机(Java Virtual Machine, JVM)。这样, 对于程序员来说, 软件的移植工作就变得非常轻松了——只要写一份源代码, 然后编译一次, 就可以在不同的硬件平台上运行。