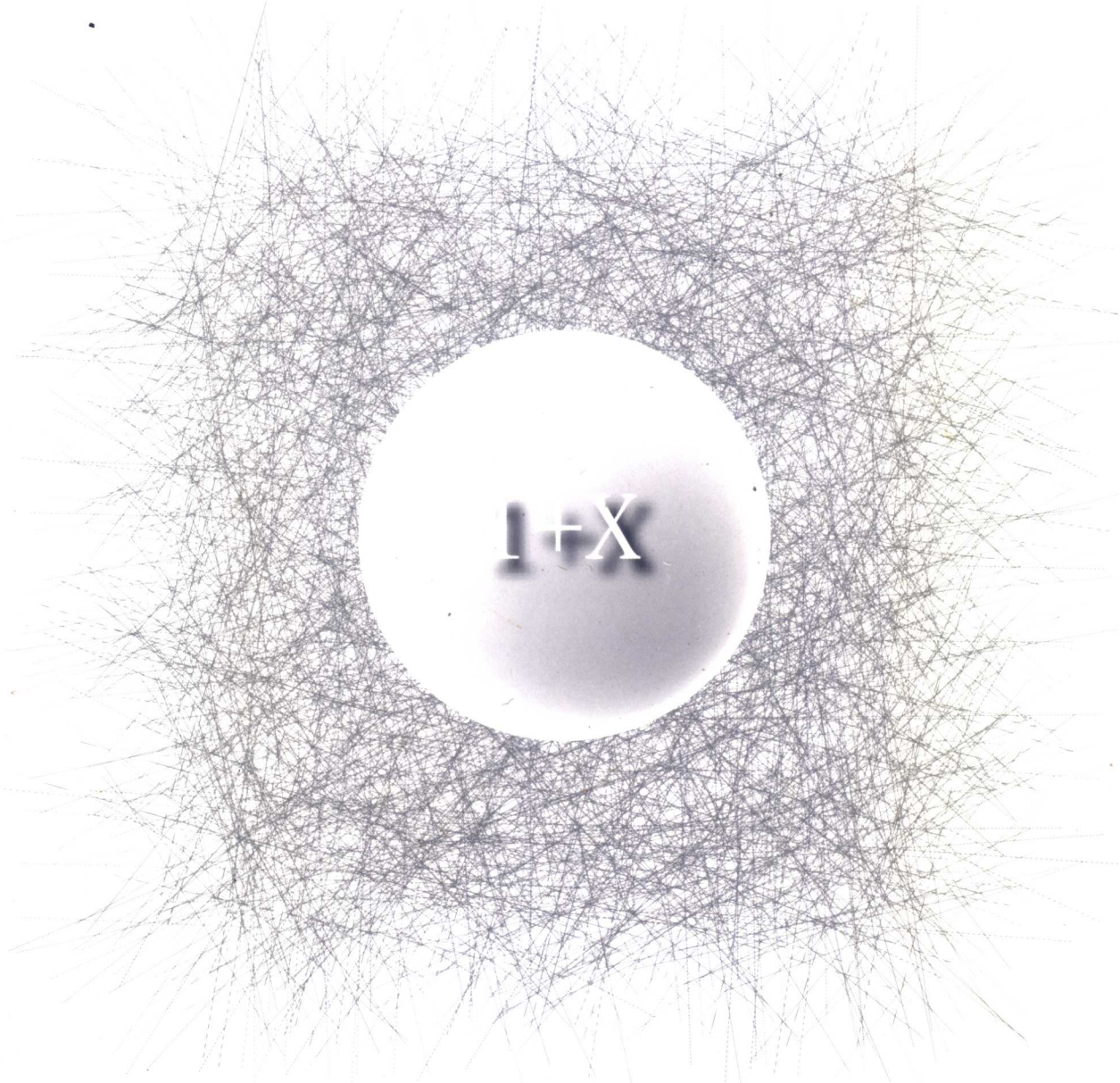


大学计算机基础教育规划教材

Java语言程序设计基础

柳西玲 许 斌 编著



1+X

清华大学出版社



大学计算机基础教育规划教材

食 商 容 內

Java语言程序设计基础

柳西玲 许斌 编著

附錄(CIP)目錄表

7. 2002. 清华大学出版社. 北京. 第1版. 第1次印刷. 16开. 300页. 7.95元

大學教育基金會

STETT 凡 林德一書字

书难出半大半皆：零 强 出

010-65376969 类别白蜜

另派：博德升表

天津送京非古总古计半继：清 子 世

清华大学出版社
北 京

内 容 简 介

本书共分为8章,前两章介绍面向对象编程概念和Java程序设计基础知识,其他几章介绍Java语言编程语法和技术,包括对象、类和接口的创建和使用,输入输出处理,异常处理,Applet程序设计,对象串行化和聚集,线程等。本书光盘中包括Sun(中国)公司授权分发的J2SE Development Kit 5.0和NetBeans IDE 4.2复合软件包,以及本书实例的源程序。书中的附录是NetBeans集成开发环境简介。

本书的主要读者对象为非计算机专业的本科学生,强调基本概念、基本技术、基本方法的阐述,强调理论联系实际。书中列举逾百个实例,每章都有案例分析,努力提高读者的实际解决问题能力。本书可作为大专院校的第一编程语言教材,也可作为Java编程爱好者的参考书。如想进一步深入学习,还可参考《Java语言应用开发基础》(清华大学出版社出版)一书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

Java语言程序设计基础/柳西玲,许斌编著. —北京:清华大学出版社,2005.7
(大学计算机基础教育规划教材)

ISBN 7-302-11264-9

I. J… II. ①柳… ②许… III. JAVA语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2005)第069782号

出 版 者:清华大学出版社

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

社 总 机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

责任编辑:张民

印 装 者:北京国马印刷厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:21.5 字数:501千字

版 次:2005年7月第1版 2005年7月第1次印刷

书 号:ISBN 7-302-11264-9/TP·7432

印 数:1~5000

定 价:33.00元(含光盘)

大学计算机基础教育规划教材

为了进一步推动高校计算机基础教育的发展,教育部高等学校非计算机专业计算机基础课程教学指导分委员会近期提出了《关于进一步加强高校计算机基础教学的若干意见》(以下简称《意见》)。《意见》针对计算机基础教学的现状与发展,提出了计算机基础教学的指导思想;按照分类、分层次组织教学的思路,《意见》的附件提出了计算机基础课教学内容的知识结构与课程设置。《意见》认为,计算机基础教学的典型核心课程包括:大学计算机基础、计算机程序设计基础、计算机硬件技术基础(微机原理与接口、单片机原理与应用)、数据库技术与应用、多媒体技术与应用、网络技术与应用。附件中介绍了上述六门核心课程的主要内容,这为今后的课程建设及教材编写提供了重要的依据。在下一步计算机课程规划工作中,建议各校采用“1+X”的方案,即:“大学计算机基础”+若干必修或选修课程。

教材是实现教学要求的重要保证。为了更好地促进高校计算机基础教育的改革,我们组织了国内部分高校教师进行了深入的讨论和研究,根据《意见》中的相关课程教学基本要求组织编写了这套“大学计算机基础教育规划教材”。

本套教材的特点如下:

- (1) 体系完整,内容先进,符合大学非计算机专业学生的特点,注重应用,强调实践。
- (2) 教材的作者来自全国各个高校,都是教育部高等学校非计算机专业计算机基础课程教学指导委员会推荐的专家、教授和教学骨干。
- (3) 注重立体化教材的建设,除主教材外,还配有多媒体电子教案、习题与实验指导,以及教学网站和教学资源库等。
- (4) 注重案例教材和实验教材的建设,适应教师指导下的学生自主学习的教学模式。
- (5) 及时更新版本,力图反映计算机技术的新发展。



Java

语言程序设计基础

本套教材将随着高校计算机基础教育的发展不断调整,希望各位专家、教师和读者不吝提出宝贵的意见和建议,我们将根据大家的意见不断改进本套教材的组织、编写工作,为我国的计算机基础教育的教材建设和人才培养做出更大的贡献。

“大学计算机基础教育规划教材”丛书主编
教育部高等学校非计算机专业计算机基础课程教学指导分委员会主任委员

冯博琴

2004 年 8 月

Java

前言

语言程序设计基础



1995 年 Java 语言诞生,在全世界立刻兴起了学习 Java 的热潮。随后,1998 年 12 月 J2SE 平台问世,2000 年,企业版 J2EE 推出。在不到两年的时间,美国 500 强企业都成功地将自己的原软件系统转为 J2EE 平台。至今,J2EE 1.4 版在网上被下载已逾百万次,充分显示了 Java 技术在 Web 服务开发领域的应用潜力。近年来 Java 技术已为全球 17 亿电子器件注入新的活力,快速地进入了有线电视、电子游戏、网络电话和汽车等行业。全球已有近 4 亿部 Java 电话,Sun 公司总裁兼首席运营官 Jonathan Schwartz 说,“Java 技术正在成为全球网络应用的事实标准,它将大大加快和简化提供移动、消费和企业市场的服务。”正是技术创新促进了 Java 技术的发展,使之成长为价值 1200 亿美元以上的 Java 经济。近来,又有几个具有创新意义的新产品,使 Java 技术进入新的市场。例如,德国宝马系列汽车和西门子 VDO 汽车内的导航与娱乐系统,MedicTouch 公司基于 Java 技术的移动健康诊断仪,Java 技术已经通过电视机顶盒进入了消费者的家庭,美国宇航局(NASA)利用 Java 3D 技术实现火星漫游等。现在 Java 平台仍继续为 Java 经济注入活力,并驱动了全球企业在桌面系统和服务器领域的技术创新。

如今美国许多著名大学也正在应用 Java 平台。如斯坦福大学用 J2EE 平台实现了图书馆管理系统,使图书资料可供全世界网络共享。美国马里兰大学学院综合大学(University of Maryland University College, UMUC)自 1972 年开始提供全球远程教学,选择了 Java 平台实现它的全球 PeopleSoft 应用。把 UMUC 现有的 3 个异构的系统在 J2EE 平台上整合为一个集中控制的 IT 数据库,将其全球的财务、人力资源和学生工作等系统统一管理起来,使 UMUC 能高度缩放地、安全可靠地扩展它的教学与学术研究工作。该项目的执行结果使全球学生、教师 and 所有教职员工可以更简单、更快速地访问在线工具与信息。UMUC 现在已在 28 个国家拥有 87 000 多名在线学生。自 1999 年以来,UMUC 的学生人数以每年 5%~10% 的速度激增,预计在 2005 年,全球在线学生总数可达到 10 万人以上。

由于 Web 技术的飞速发展,互联网的使用继续爆炸性地增长。面向 Web 应用的程序已成为软件系统的主流,Java 本身就设计为互联网编程的语言,而且,已为应用提供了许多成熟的软件开发工具和应用程序的开发包;体系结构中立的设计使 Java 具有很好的可移植性;Java 对安全与可靠的支持与其自身的设计融为一体,特别适合网络安全与可靠的需求;面向对象使得 Java 成为现代软件工程和编程概念的良好教学载体。因此,Java 非常适合作为普及编程语言的课程。

到 2003 年为止,全球获得 Java 认证的人员已有几百万,但根据 Sun 公司调查,在近

几年内,全球需要 Java 人才超过千万。我国 Java 学习热潮虽滞后几年,但从 2002 年,由用户需求的驱动也已兴起。我们的调查结果表明,许多用户都要求在 Java 平台或用 Java 编程完成自己的软件系统。市场推动了对 Java 人才的需求,有 Java 基础的人也更受到公司的欢迎。目前,虽然全国已有一百多所大学的计算机系开设了 Java 课程,但仅靠计算机专业培养的 Java 人才远远不能满足市场需求。实际上,现在有大量非计算机专业人员在 Java 编程岗位上工作,这也说明了 Java 语言作为大学一门公共课程的必要性。在教学指导委员会统一规划下,要求我们写这本非计算机专业的大学通用 Java 编程教材。能为普及和应用 Java 技术做点贡献,我们也深感欣慰。为此,我们调查了几个典型大学的 Java 课程情况和国外 Java 的各类教材。结合我们多年 Java 教学和培训经验,制定了本书的目标:希望本书既能成为各类读者的第一门编程语言教程,也能成为各大学非计算机专业的一门编程语言的入门教程。本书假定读者没有任何编程技能基础,要求的数学、计算机和科学知识,相当于大学一年级学生的水平。

说实话,人们学会任何一种编程语言都有一定困难。虽然 Java 语言已使编程得到很多的简化,但编程是一门“身教盛于言教”的课程,它与编写经验密切相关。编程在很多方面与写作很相似。学习写作要学习词汇、文章的组织构思,用优美而恰当的词汇表达。先从造句、段落开始,反复练习,直到能写出很长的文章、故事、报告甚至小说。写作技能在反复的实践中不断地增长。学习程序设计的方法也要先熟悉程序设计的基本知识、流程控制语句、已有类库主要内容等。然后,再练习编写程序段、模块和子系统程序,直到能编写一个完整的应用系统。开始先编写简单但能正确满足需求的软件,以后再编写复杂、优质的典型软件系统。教编程语言更难一些,要能通过与生活很接近的事例,让学生建立对象的基本概念和某些对象方法,结合软件工程原则,利用好的范例分析,很快地把学生领进门。同时能激发学生学习的兴趣,使学生感到 Java 程序编写是一件容易的事。然后,又如同培养外科医生一样,要强调多动手实践。多编写、多练习,培养良好的编程习惯和严谨的科学作风。在团队中学会沟通,学习他人长处,弥补自己的不足。通过实践不断提高编程技能和兴趣,共享成功的喜悦和累积经验。

本书为适合不同基础的读者,采用分析案例、大量的自我程序练习,给读者提供了设计、组织和编写代码的学习机会。并在讲解中随时提醒编程中容易出现错误,推荐一些编程的良好习惯。尽可能让读者在积极的参与中学习。

本书的总目标是:

(1) 教材首先要明确读者对象,才能合理取舍内容。本书的对象是没有程序设计实际经验的非计算机专业的大学生。当今,面临信息化的时代,计算机已深入人类生活的方方面面,并成为各行业的应用工具。作为一名大学生,应该具备一定的计算机知识和编程技能。学会 Java 编写简单程序,能为今后在各专业中进一步的应用,打下良好基础。

(2) 面向对象编程的概念和原则已成为现在编程的主流。因此,本书首先从人们认知规律出发,由浅入深,由具体到抽象地展示和鼓励使用面向对象范例。介绍用对象表示的方法。使读者一开始就可使用标准包中的对象,很快对一些有兴趣的问题,开发出有意义的程序。在这些概念基础上,再提出基本类和面向对象设计。在探索控制结构之后,对方法、类和面向对象设计进行了更深一步的介绍。使初学者不会因编程语言的许多基本

概念、算法而感到枯燥乏味,失去了兴趣。同时,通过一步步引导和分析,逐步掌握编写程序的规律、技巧和减少出错的方法,使读者建立编程的自信心。

(3) 注重解决问题的能力。许多初学者面临的障碍是不知道解决问题的基本方法。对于这个问题,本书在第1章就介绍了解决问题的基本技能,在以后的每一章里,把一些新概念融于满足不同读者要求而选择的问题中。读者先学习这些有效解决问题的例子,然后再提供独自解决类似问题的练习。特别在书中列举了上百个简单程序实例,都能在最基本的Java运行环境下执行。读者可作为范例进一步动手修改练习。

(4) 教给读者软件设计的概念。在书中每一章都有一个或多个案例,从对问题需求分析开始,教读者如何着手解决问题。在某种程度上,都有面向对象的分析和设计,以及实现算法的开发。同时介绍了Java的核心类库和典型的开发工具。每章都有练习题,使读者积极参与和应用。

(5) 只使用标准的Java类,不用购入或由作者自己写的类。这样做的目的是使初学者容易理解。

(6) 除了解释Java和面向对象编程,本书还对在编程风格和技巧上,怎样成为一个更好、更有见地的编程者提出一些忠告。在避免常见的编程错误,编写出可读性好的代码以及遵循软件工程原则等方面,都有一些很重要的技巧。

(7) 本书中带*的小节,可供有兴趣更深度学习的读者学习,初学者可以跳过不学。

本书由柳西玲和许斌编著,书中的实例都在JDK 1.5和NetBeans上运行过。对教育部高等学校非计算机专业计算机基础课程教学指导分委员会和清华大学出版社给予本书的重视和支持深表感谢。有不妥和错误之处,盼望读者给予指正。

作 者

2005年6月

第 1 章 概论	1
1.1 计算机与软件	1
1.2 Java 简介	3
1.2.1 编程语言的演化	3
1.2.2 Java 语言的特点和优势	4
1.3 面向对象技术基础	7
1.3.1 面向对象的基本概念	9
1.3.2 面向对象的基本特性	13
1.3.3 面向对象的程序设计方法	14
1.4 Java 体系结构	15
1.4.1 Java 的实现机制	15
1.4.2 JDK 的下载和使用	17
1.5 最简单的 Java Application 实例剖析	23
1.5.1 代码注释	24
1.5.2 定义一个类	24
1.5.3 要求 main 方法	26
1.5.4 实例编译与运行	28
1.6 问题求解过程	29
1.6.1 软件工程基本原理	29
1.6.2 问清需求	32
1.6.3 明确解决需求的关键	35
1.6.4 将复杂问题分解原理	36
习题	37
第 2 章 Java 程序设计基础	39
2.1 数据类型	39
2.1.1 标识符	39
2.1.2 数据类型的划分	40
2.1.3 基本数据类型	42
2.1.4 常量与变量	42



2.1.5 各类数据间的转换	48
2.2 运算符与表达式	50
2.3 表达式语句	66
2.4 数组	67
2.4.1 创建一维数组	67
2.4.2 初始化一维数组	69
2.4.3 复制一维数组	72
2.4.4 多维数组	72
2.5 字符和字符串	75
2.5.1 字符类	75
2.5.2 字符串类	77
2.6 控制流程语句	78
2.6.1 循环语句	78
2.6.2 分支语句	86
2.6.3 多分支语句	88
2.6.4 其他分支语句	92
* 2.6.5 递归	95
2.7 案例分析	97
习题	99
第3章 Java语言中面向对象编程的特征	102
3.1 概述	102
3.2 类	102
3.2.1 创建类	103
3.2.2 构造方法	107
3.2.3 继承	108
3.2.4 多态	109
* 3.2.5 内部类	113
3.2.6 抽象类	116
3.3 对象	117
3.3.1 对象的创建	117
3.3.2 对象的使用	119
3.3.3 对象的清除	123
3.3.4 定制化数据	123
3.4 接口和包	126
3.4.1 定义接口	127
3.4.2 实现接口	128
3.4.3 接口作为类型的使用	128
3.4.4 创建包	129

3.4.5 使用包中成员	130
3.5 字符串操作	131
3.5.1 访问字符串	131
3.5.2 字符串的修改	132
3.5.3 字符串的比较	133
3.5.4 其他操作	134
3.5.5 程序综合例子	134
3.6 案例分析	135
习题	139
第4章 输入输出流及文件操作	144
4.1 概述	144
4.2 文件	147
4.2.1 创建文件	147
4.2.2 文件类提供的方法	148
4.2.3 随机文件流	151
4.2.4 文件复制	153
4.3 字节 I/O 流	155
4.3.1 字节输入流	155
4.3.2 字节输出流	156
4.3.3 内存的读写	156
4.3.4 字节流实例	157
4.4 字符 I/O 流	160
4.4.1 字符输入流	160
4.4.2 字符输出流	163
4.5 过滤流	164
4.6 管道流	167
4.7 案例分析	169
习题	172
第5章 异常处理	175
5.1 概述	175
5.1.1 异常处理机制	176
5.1.2 Throwable 类及其子类	180
5.2 异常处理	182
5.2.1 捕获异常	182
5.2.2 声明抛出异常	183
5.2.3 抛出异常	184
5.2.4 处理异常	185



5.3 案例分析	192
习题	200
第 6 章 Applet 程序设计	205
6.1 概述	205
6.1.1 最简单的 Java Applet 实例剖析	205
6.1.2 Applet 的基本框架	209
6.1.3 Applet 的生命周期	210
6.1.4 Applet 的类层次结构	212
6.1.5 Applet 的安全机制	212
6.2 AppletViewer	215
6.3 Applet 类的 API	217
6.3.1 支持多媒体的方法	217
6.3.2 管理环境的方法	217
6.3.3 报告信息的方法	218
6.4 Applet 的显示	218
6.4.1 显示的主要方法	219
6.4.2 Graphics 类	221
6.5 Applet 的编写	222
6.5.1 Applet 的编写步骤	222
6.5.2 用户 Applet 类的定义	222
6.5.3 Applet 的参数	223
6.6 Applet 中的 GUI	226
6.6.1 基于 AWT 的 Applet 用户界面	227
6.6.2 基于 Swing 的 Applet 用户界面	229
6.6.3 Applet 中的事件处理	232
6.7 Applet 的多媒体支持	234
6.7.1 图像	235
6.7.2 动画制作	237
6.8 案例分析	239
习题	243
第 7 章 对象串行化和聚集框架	246
7.1 对象串行化的概念	247
7.2 对象串行化的方法	247
7.3 定制串行化	250
7.3.1 部分定制串行化	250
7.3.2 完全定制串行化	253
7.4 串行化中对敏感信息的保护	255

7.5 聚集框架的概念	256
7.6 简单聚集类	259
7.6.1 Vector	259
7.6.2 对象 Array	261
7.6.3 Stack	262
7.6.4 Hashtable	263
7.7 Collection	266
7.8 Set	268
7.9 List	270
7.10 Map	271
7.11 案例分析	273
习题	278
第8章 线程	281
8.1 线程概念	281
8.1.1 什么是线程	281
8.1.2 线程的优势	283
8.1.3 线程的模型	285
8.2 线程的创建	285
8.2.1 用 Thread 类创建线程	285
8.2.2 用实现 Runnable 接口创建线程	288
8.3 线程的调度与控制	289
8.3.1 线程的优先级	290
8.3.2 线程的基本方法	294
8.3.3 线程的控制	295
8.4 线程同步	298
8.4.1 简单同步	298
8.4.2 对象锁和类锁	300
8.4.3 避免死锁	304
8.5 线程的生命周期	306
8.6 案例分析	307
习题	311
附录 A NetBeans 集成开发环境简介	312
参考文献	325

第1章

概 论

互联网的飞速发展,使计算机更深入人心,也带来更多的机会和挑战。对软件的挑战包括开发在各种不同类型计算机上运行的程序,以及开发上百万行准确无误的代码软件(如全球电子商务系统、网上股票系统、网上拍卖系统等)。目前,面向对象编程方法和软件构件技术已成为应对这些挑战的开发软件的主流技术。本章介绍计算机和软件的发展、计算模式的演化、面向对象技术基础、Java 语言的由来、Java 的体系结构、最简单的 Java 应用实例的剖析,最后讨论用软件解决问题的过程。希望读者在一开始就建立起用软件解决问题的基本方法,并在学习编程技术过程中树立面向科学、面向服务、面向实践、面向系统的观点。

1.1 计算机与软件

计算机是快速电子计算设备,它的计算速度比人要快几百万倍甚至几亿倍。计算机硬件和软件构成完整的计算机系统。所谓硬件是计算机系统内的各种设备,如显示器、硬盘、键盘、主板等。所谓软件是在计算机中运行的程序和对程序加以说明的文档。近年来,由于电子芯片组件技术的突破和个人计算机工业的产业化,硬件价格急剧下降。而软件由于应用功能和复杂度的急剧上升,给开发增加了许多难度。虽然,软件学科已经历几十年的发展,开发方法也在不断改进和完善,但仍满足不了用户对软件的需求。计算机发展至今,软件已成为信息时代科学发展的焦点之一。

1999 年 2 月,在美国总统 IT 顾问委员会的一份报告中,列举了大量的事实,论证 IT 技术对社会和国家以及人民生活的重要作用。委员会建议政府重点支持四大项目,把软件列在首位。报告认为软件是信息时代的最重要的基础设施。把软件提到前所未有的高度,是因为现在人们生活、工作、学习等各方面已经离不开计算机,正如同人们离不开水和电一样。如今软件已是计算机系统的核心。人们对软件的认识,是在艰苦曲折的历程中不断提高的。自计算机问世以来,计算机的计算模式经历了 3 次浪潮的冲击。20 世纪 50 年代至 70 年代,计算机系统是基于主机的计算模式,在分时操作系统的管理下,用户通过终端与主机交互。这种计算模式的特征是:数据、用户界面和应用程序交融为一体,在主机上运行。在这个阶段,人们认为软件就是程序,程序就是数据结构和算法。计算机的用途从解决科学计算扩大到非数值计算的商业事务。结构化开发方法开始提出。到 1981

年,个人计算机的出现使计算机广泛用于工业和政府管理,开始了客户/服务器(C/S)计算模式的第2次浪潮。其特征是:客户端上有本地数据和用户界面运行,服务器上有共享数据和应用逻辑程序运行。客户端用个人计算机,服务器端用性能较高的计算机。在这个阶段,人们认为软件是程序和文档,软件的重点由单纯的编程转向构造系统方法。对软件生命周期有了认识。以数据库为中心的应用系统为主,计算机应用扩大到金融、保险等行业。结构化开发方法和面向过程的编程是主流,而面向对象的开发方法开始崛起。20世纪90年代中期,随着互联网和3W技术的飞速发展以及Java语言的诞生,计算模式的第3次浪潮出现,即分布式网络计算模式(B/S)。其特征是:以网络为中心的多层体系结构的应用软件,分布在异地、异构机上运行。这种模式为软件开发带来了新的巨大挑战。软件系统的集成、跨平台互操作、软件重用是关键。这时,人们认识到软件是人对知识的提炼和固化、是人类知识和经验的抽象。深刻剖析软件的本质特性是构造性和演化性。为使软件能简洁、精确地描述客观事务,软件的多层应用建立在公共的体系结构上,这是构件技术的基本出发点,构件体系结构是构造软件的基石,它体现了软件的构造性本质。另外,软件是对客观世界中问题求解的具体描述。而客观世界是在不断发展,不断变化的。软件系统也不可能一成不变,新需求、新技术的不断出现,要求软件系统不断升级。这就是软件的演化性本质。所有软件系统的开发方法应从这两个本质特性出发,才能大幅度地提高软件的质量和生产率,以满足软件飞速发展的需求。构件技术就是能支持软件构造性和演化性的开发方法,因而很快就成为当今软件开发的主流技术。

网络计算模式使计算机的使用方式发生了根本变化。对软件提出4A目标:要求软件能使任何人(any one),在任何地方(any where),在任何时间(any time),对任何电子设备(any device)都能使用的跨空间、跨时间、跨设备的目标。这也是软件业内几代人的梦想。要想实现这个梦想,要求软件能在异构平台上互操作,具有可伸缩性和重用性,并可实现即插即用等分布式计算的需求。

从软件的计算模式变化,可以总结对软件开发方法的影响。20世纪50年代至70年代,计算机系统是基于主机的计算模式,软件开发主要通过高级编程语言编写。20世纪80年代,计算机系统是客户/服务器(C/S)计算模式,大型软件系统开发引起的软件危机,导致Yourdon和De Marco的结构化方法的盛行。结构化方法的目标是保证软件开发的质量,提高软件的灵活性和软件生产效率,通过工程化方法建立系统的软件开发过程,使开发的软件具有好的结构,即可组装、可裁剪的模块化结构。这使软件开发在改善软件构造性和演化性方面得到很大的进步,但主要还是面向过程的思维方法,没有从根本上解决危机。20世纪80年代末,出现了面向对象的方法。其基本思路是用对象作为描写客观信息的基本单元,并将对象属性(数据)和对对象操作(方法、运算)封装在一起。从1995年开始,互联网和Web技术的高速发展,软件开发由Booch和Rumbaugh、Jacobson合作,共同提出了统一的建模语言UML,得到很多软件公司的支持,逐渐成为面向对象方法的一个事实上的标准。同时,Java语言的诞生,以及把不同时间、不同项目中成功地解决问题的解决方案总结为设计模式的复用技术的出现,结合构件分布式体系结构,形成了构件化软件开发方法。从而实现了分布式网络计算的跨空间、跨时间、跨设备的信息共享。使软件能在各层次上集成,异构环境下协同工作,向软件工业化道路前进了一大步。构件化

方法在短短几年来,已成为当今软件开发方法的主流。所谓软件构件化,就是要让软件开发像计算机硬件一样,可以用各种标准和非标准的构件来进行组装。软件的构件化和集成技术的目标是:软件可以由不同厂商提供,用不同语言开发,在不同硬件平台上实现构件的组装。软件构件可方便地、动态地集成。这些构件能互操作,可以放在本地的计算机上,也可以分布式地放置在网上异构环境下的不同结点上。实现软件的构件化,支持软件构造性,实现了软件高抽象层次重用,从而加速软件开发,提高软件生产率;实现软件的构件化,支持软件演化性,实现软件可局部替换、即插即用的动态集成,从而提高软件易维护、易使用和易伸缩的灵活性,提高软件可扩展、可移植的互操作性。这些都是软件业界多年来奋斗的目标,也是几代人努力的结果。

1.2 Java 简介

1.2.1 编程语言的演化

计算机之所以能帮助人们完成各种事情,完全依赖于计算机中运行的软件系统——程序和文档。其中,程序指挥计算机的每一步操作,最终完成指定的任务。所谓程序就是由一系列计算机指令组成的序列。这些序列都用某种特定语言编写,这种语言是专门为计算机发布命令而设计的,称为编程语言。

最原始的程序是直接利用计算机指令编写的,每一条指令对应于机器的一个基本操作。计算机按次序执行每一条指令。这种用机器语言编程的主要问题是:编程人员必须记住机器的指令系统,并直接用二进制代码编写程序,这是既烦琐枯燥,又很容易出错的过程。随后出现的汇编语言是把机器指令编码符号化的语言,仅对编写人员记住机器的指令系统方面有改进,但没有从根本上摆脱机器指令的烦琐和编程低效率的缺点。近几十年来,计算机专家一直在致力于开发更先进的编程语言,为的是让人们可以用一种更自然的方式编写程序。这种面向用户的编程语言被称为高级编程语言。所谓“高级”,是指这种编程语言比汇编语言的级别高。其显著特点是编程者不需要掌握具体的计算机指令知识,而只须根据语言的语法针对问题需求编写程序。而且一条高级语言的语句可完成许多条汇编语言的工作,使编程简练并提高了效率。由于编程语言与问题的类型密切相关,因此几十年来已有数百种高级编程语言问世。因为高级语言程序不是计算机直接执行的命令,所以必须经过翻译才能完成高级语言到机器语言的转换,这种转换程序称为编译器,其输入代码称为源程序,输出结果称为目标程序。

编程高级语言按编程思维方式不同可分为两大类:面向过程的编程语言和面向对象的编程语言。BASIC, FORTRAN, Pascal 和 C 都是早期流行的面向过程的编程语言。这类语言面向求解问题的过程,独立于计算机硬件。因此,移植性比汇编语言强。对问题不太复杂的应用还是有效的方法。但它有一个致命的缺点,就是太依赖于问题的解决过程,一旦问题发生微小变动,会导致整个程序的改动。使问题的解决与程序几乎是一一对应。已编写好的代码不能重用,使软件的维护成本昂贵。随着计算机应用范围的迅速扩大,面向过程的编程语言的缺点就更为突出。为克服这个缺点,出现了面向对象的编程语言。

面向对象的基本思想是用一种更类似人类思维的方式去解决客观问题。把客观世界的任何实体和概念都看作对象,每个对象都有自己的特点和行为。不同的对象之间可以互相通信。将对象的特点称为对象的属性,对象的行为称为对象的方法。至今,面向对象编程语言已有几十种,典型的面向对象编程语言有 Smalltalk, C++ 和 Java。其中, Smalltalk 是最早也是最纯的面向对象编程语言。由美国 Xerox 公司开发。C++ 由 AT&T 的贝尔实验室开发。它保留了 C 语言的所有成分,增加了许多面向对象的设计思想,很快成为流行的编程语言。现已修订了多次版本。使用时要注意不同版本的环境。Java 语言由 Sun 公司开发,它选取了 Smalltalk 的精华,删除了 C++ 中许多不合理的内容,如:全局变量、goto 语句、指针、结构和联合、头文件等。使其具有简单、安全、健壮、跨平台等特性。有关面向对象的各种概念在后面章节还会有更深入的讲解。这里想提醒读者,要掌握 Java,学习语法只是很小的一部分内容,更重要的是学习如何用面向对象的思维观点去分析和解决问题。这是编程的基本功。面向对象的方法大大提高了程序的重用性,也在一定程度上降低了程序的复杂性。特别是 Java 语言,以其安全可靠和跨平台等特性成为网络应用软件的主流编程语言。学习和掌握 Java 编程技术将在信息时代大有用武之地。

1.2.2 Java 语言的特点和优势

1991 年, Sun 公司的 Jame Gosling, Bill Joe 等人, 为在电视、控制烤面包机等家用电器的交互操作, 开发了一个 Oak 软件, 它是 Java 的前身。当时, Oak 并没有引起人们的注意, 直到 1994 年后期, 互联网和 3W 飞速发展, 他们用 Java 编写了 HotJava 浏览器, 得到 Sun 公司首席执行官 Scott McNealy 的支持, 并进一步研制和发展成编程语言。1995 年, Oak 更名为 Java。Java 的得名还有段小插曲, 一天, Java 小组成员正在喝 Java 咖啡时, 议论给新语言取名字的问题, 有人提议用 Java, 它的原意是印度尼西亚盛产咖啡的一个岛屿。这个提议得到其他成员的赞同, 于是就采用 Java 来命名此新语言。Java 问世后, 很快被工业界认可, 许多大公司(如 IBM, Microsoft, DEC 等)购买了使用权。1995 年, Java 被 PC Magazine 杂志评为十大优秀科技产品。从此, 开始了 Java 应用的新篇章。

Java 是一种网络编程语言, 因为问世比较晚, 故避免了许多其他编程语言的缺点, 更好地利用了当前软件新技术, 它的优势可从三方面来看。首先, 它作为编程语言, 简单易学, 以面向对象方法为基础, 但又独立于硬件结构, 具有可移植性、健壮性好、安全性高、高性能的优点。其次, 它围绕网络应用开发, 最大限度地利用网络资源。它的小应用程序(Applet)能在网络上传输而不受计算机 CPU 和环境的限制。第三, Java 提供了丰富的类库, 为编程人员提供快速和标准的应用接口, 是构件集成的范例, 大大提高了应用软件的生产率。从更具体细致的角度分析, Java 语言具有 12 项特点:

1. 简单易学

Java 的简单性首先表现在自身系统的精练, 它的基本解释程序和类库仅 40KB, 附加的基本标准类库和多线程支持也仅占 175KB。它力图用最小的系统完成尽可能多的功能。不要求庞大的硬件环境, 在很普通的计算机上就能运行。其次, 它只要求理解一些基本概念就能编程。无需任何编程基础就可学会, 它的基本语法与 C++ 类似, 但删除了