



普通高等教育“十一五”国家级规划教材

普通高等学校计算机专业特色教材

Java语言程序设计

(第2版)

殷兆麟 范宝德 朱长征 编著

王潜平 主审



高等教育出版社
Higher Education Press

普通高等教育“十一五”国家级规划教材
普通高等学校计算机专业特色教材

Java 语言程序设计

(第2版)

殷兆麟 范宝德 朱长征 编著
王潜平 主审

高等教育出版社
Higher Education Press

内容提要

本书是普通高等教育“十一五”国家级规划教材。本书的取材、编写立足于满足广大 Java 初学者入门和发展的需要。全书共分为 11 章,第 1~7 章介绍计算机语言概述、Java 开发环境基本知识、Java 语言基础、Java 面向对象的程序设计、异常处理、Java 的基本可重用类和图形用户界面。这些内容构成了 Java 语言面向对象程序设计的基础。在介绍上述内容的同时,还适当介绍 Eclipse 集成环境的安装、配置、Java 类的调试及类单元的测试。第 8~11 章是 Java 语言的特色部分,主要介绍多线程程序设计、Java 小应用程序、网络编程基础以及 Java 与数据库连接技术 JDBC。开设这门课程,可为学生学习后续 J2EE 中间件技术、Web 系统与技术等专业课程打下基础。

本书可作为高等院校计算机及相关专业的 Java 语言程序设计或 Java 网络编程基础等课程的教材,也可以作为应用开发人员的自学参考书。

图书在版编目(CIP)数据

Java 语言程序设计/殷兆麟,范宝德,朱长征编著.

—2 版. —北京:高等教育出版社,2007.7

ISBN 978 - 7 - 04 - 021937 - 1

I. J… II. ①殷…②范…③朱… 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(2007)第 089288 号

策划编辑 刘 茜 责任编辑 彭立辉 封面设计 张志奇 责任绘图 尹 莉
版式设计 张 岚 责任校对 胡晓琪 责任印制 陈伟光

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010 - 58581000

经 销 蓝色畅想图书发行有限公司
印 刷 北京市白帆印务有限公司

开 本 787 × 1092 1/16
印 张 23
字 数 520 000

购书热线 010 - 58581118
免费咨询 800 - 810 - 0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>
畅想教育 <http://www.widedu.com>
版 次 2002 年 12 月第 1 版
2007 年 7 月第 2 版
印 次 2007 年 7 月第 1 次印刷
定 价 28.70 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 21937 - 00

郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人给予严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话：(010) 58581897/58581896/58581879

传 真：(010) 82086060

E - mail：dd@hep.com.cn

通信地址：北京市西城区德外大街4号

高等教育出版社打击盗版办公室

邮 编：100011

购书请拨打电话：(010)58581118

第 1 版前言

Java 语言是网络时代广泛使用的面向对象的编程语言，它具有可移植性、安全性、多线程机制等众多优点，具有非常高的技术性能。因此，Java 语言得到越来越广泛的应用。为了满足广大计算机编程初学者入门和提高自己的需要，为了掌握 Java 进而达到实用，同时考虑满足 Java 编程 IT 国际认证考试的需要，我们编写了本教材。教材内容由浅入深，避免开始引入过多的技术术语和介绍过多的 Java 特点，避免介绍 Java 与 C++ 的比较，以减少编程初学者理解的困难；同时，增加编程示例，增加基本算法的介绍，以培养初学者的编程思想，逐步提高其编程能力。

Java 语言程序设计的教学分为两个阶段进行。第一阶段目的是使学生掌握 Java 语言的基本语法，培养学生的编程思想，具有初步的面向对象编程能力。第二阶段目的是进一步提高学生面向对象编程能力，认识、掌握 Java 的特点。

本书第 1~8 章为第一阶段讲授的内容。第 1 章介绍与计算机语言有关的基本知识；第 2 章介绍 Java 的开发环境；第 3 章介绍 Java 语言基础；第 4 章介绍 Java 面向对象的程序设计；第 5 章介绍异常处理；第 6 章介绍 Java 的基本可重用类；第 7 章介绍 Java 小应用程序 Applet；第 8 章介绍 Java 的图形用户界面。为了达到讲授目的，为后续计算机课程服务，第一阶段还介绍基本算法和单向链、双向链、队列、堆栈的 Java 编程，这是一般 Java 语言教程中所忽视的，这阶段教学约需要 40 学时（不包括上机）。

本书第 9~11 章为第二阶段讲授的内容。第 9 章介绍 Java 的多线程机制；第 10 章介绍 Java 网络编程基础；第 11 章介绍 Java 与数据库的连接——JDBC 技术。第 12 章通过综合示例进一步介绍 JBuilder6.0 的使用。这阶段按教学对象不同需要 36~48 学时（不包括上机）。

第二阶段的内容要求学生要具备基本的数据库和网络知识，教师可作适当补充。如对计算机专业高年级学生讲授第二阶段内容是不困难的。

与本教材配套的 CAI 课件包括三方面内容：教师讲课用的课件、学生上网浏览用的课件和 Java 上机实验指导（包括教材中的示例源程序）。有兴趣的读者可以 E-mail: zhlying@cumt.edu.cn 进行查询。

本教材第 12 章的 12.1、12.2 和 12.3 节引用了 JBuilder 6.0 Enterprise Help 中的部分内容。这里对原文的作者表示谢意。

上述工作得到中国矿业大学教务处、计算机科学与技术系的关心和支持。该教材列为我校“新世纪教材建设工程项目”之一，教材 CAI 课件得到江苏省教委多媒体课件项目的资助，获得江苏省教委 2002 年多媒体课件三等奖，在此表示衷心感谢。

由于时间仓促，作者水平有限，书中难免有不妥和错误之处，恳请广大读者指正。

主编殷兆麟

2002.5.18

第 2 版前言

本书第 2 版被列入普通高等教育“十一五”国家级规划教材，与第 1 版相比，做了如下修改：

- Java 教学环境选择开放的 Java 集成开发环境 Eclipse。教材介绍了 Eclipse 的下载、安装、配置、开发步骤、程序调试以及 Java 单元测试等操作过程。
- 加强了面向对象分析设计的介绍。学生面向对象程序设计的能力必须与面向对象分析设计同步进行，从简单示例到比较复杂的示例，螺旋式推进。面向对象分析设计阶段适当引入不严格的可视化类图来描述类的内容与类之间的关联，这样有助于学生对面向对象分析设计的理解。
- 图形用户界面设计调整到 Java Applet 之前介绍。图形用户界面设计以 Frame 容器布置组件为主，重视界面、系统逻辑、系统控制三部分之间的清晰关系，强调事件驱动与程序驱动编程的不同。

上述内容符合教育部高等学校计算机科学与技术教学指导委员会编制的《高等学校计算机科学与技术专业发展战略研究报告暨专业规范（试行）》中有关部分知识单元、知识点的要求。

与本教材配套的 CAI 课件可以自由地按下列方式下载：进入 <http://www.cumt.edu.cn> 主页，单击下方的“网络教学”，在网络教学页面选择“Java 语言程序设计 CAI”。此外，也可以与主编（zhlyin@cumt.edu.cn）联系。

刘振海、孙磊和耿伟协助编写教学示例，开发课件，做了大量的工作。

另外，本书的改编工作得到中国矿业大学和烟台大学等有关单位的关心和支持，在此一并表示衷心的感谢。

由于时间仓促，作者水平有限，书中难免有不妥和错误之处，恳请广大读者指正。

殷兆麟

2007.3.18

目 录

第 1 章 计算机语言概述1	
1.1 机器语言与汇编语言.....1	
1.2 高级程序设计语言.....1	
1.3 语言翻译程序.....4	
1.4 虚拟机、Java 虚拟机与 Java 运行环境.....6	
1.5 Java 程序的运行.....7	
1.6 Java 程序的开发、执行流程.....8	
小结.....9	
习题.....9	
第 2 章 Java 开发环境基本知识10	
2.1 Java 开发环境概述.....10	
2.2 JDK 的安装、环境配置及其 使用.....10	
2.2.1 JDK 的安装与环境配置.....10	
2.2.2 JDK 的环境工具及其 使用.....11	
2.2.3 JDK 下 Java 编程.....14	
2.3 Eclipse 集成开发环境简介.....15	
2.3.1 Eclipse 工作台简介.....19	
2.3.2 菜单栏.....19	
2.3.3 工具栏.....37	
小结.....43	
习题.....43	
第 3 章 Java 语言基础44	
3.1 一个简单的 Java 程序实例.....44	
3.1.1 类首部.....44	
3.1.2 类体.....44	
3.2 标识符、注释和分隔符.....45	
3.2.1 标识符.....45	
3.2.2 注释.....46	
3.2.3 分隔符.....46	
3.3 变量和常量.....46	
3.3.1 变量.....46	
3.3.2 常量.....47	
3.4 基本类型.....47	
3.4.1 整型.....48	
3.4.2 浮点型.....48	
3.4.3 布尔型.....49	
3.4.4 字符型.....50	
3.5 运算符及表达式.....51	
3.5.1 表达式.....51	
3.5.2 运算符.....51	
3.5.3 算术运算符.....51	
3.5.4 关系运算符.....52	
3.5.5 逻辑运算符.....52	
3.5.6 位运算符.....53	
3.5.7 赋值运算符.....53	
3.5.8 条件运算符.....53	
3.6 运算符的优先级.....54	
3.7 数据类型转换.....55	
3.7.1 自动类型转换.....56	
3.7.2 强制类型转换.....56	
3.8 数组.....56	
3.8.1 一维数组.....56	
3.8.2 数组的初始化.....57	
3.8.3 数组元素的引用.....58	
3.9 简单语句和复合语句.....58	
3.10 控制语句.....62	
3.10.1 选择语句.....62	

3.10.2	switch 语句	66	4.6.3	包的使用	124
3.10.3	循环语句	69	4.7	综合应用示例	125
3.10.4	循环的嵌套	74	4.7.1	单向链表	125
3.10.5	break 和 continue 语句	76	4.7.2	利用单向链表实现 人员管理系统	132
3.11	综合应用举例	78	4.7.3	队	135
3.12	语法错误的排除	81	4.7.4	栈	136
小结		85	小结		137
习题		85	习题		137
第 4 章	Java 面向对象的程序设计	88	第 5 章	异常处理	143
4.1	面向对象技术的理论基础	88	5.1	异常概述	143
4.2	类和对象的基本概念	88	5.2	抛出异常与处理异常语句	144
4.2.1	对象	88	5.2.1	throws 异常说明	144
4.2.2	对象类	89	5.2.2	throw 异常说明	145
4.2.3	类和对象的图示化	89	5.2.3	try...catch...finally 语句	147
4.2.4	类之间的关联	91	5.3	综合应用示例	149
4.2.5	接口	95	5.3.1	人员管理系统进一步 讨论	149
4.2.6	包与类	96	5.3.2	单向链表类	150
4.3	Java 类	96	5.3.3	InputPerson 类	150
4.3.1	类首部说明	97	5.3.4	人员类 Person	152
4.3.2	类体	99	5.3.5	人员管理类 PersonManager_2	154
4.3.3	由简单类图、对象图 编写 Java 类实例	107	5.3.6	项目的后续工作	156
4.3.4	创建内类	108	5.4	程序运行错误的排除及异常 处理的调试	156
4.4	继承	109	5.4.1	被调试程序概述	156
4.4.1	类继承语法形式	109	5.4.2	调试工具的使用	157
4.4.2	null、this、super	111	小结		161
4.4.3	多态性	112	习题		162
4.4.4	final 和 static	114	第 6 章	Java 的基本可重用类	165
4.5	接口	116	6.1	Java 可重用类的结构	165
4.5.1	接口的说明	116	6.2	Java 语言包	166
4.5.2	接口的使用	118	6.2.1	Object 类	166
4.5.3	抽象类和方法	120	6.2.2	包装类	167
4.6	包	123			
4.6.1	包说明	123			
4.6.2	包的层次结构	124			

6.2.3	字符串类	168	7.3.2	事件处理	224
6.2.4	Math 类	170	7.3.3	事件适配器	231
6.3	java.util 包	171	7.3.4	事件驱动程序综合示例	232
6.3.1	java.util 包的构成	171	7.4	绘图组件	239
6.3.2	日期时间类	171	7.4.1	画布	239
6.3.3	向量类及其使用	172	7.4.2	paint()方法	239
6.3.4	哈希表类及其应用	176	7.4.3	重新绘制	240
6.3.5	对象集合类	178	7.4.4	用 Graphics 对象绘图	240
6.4	输入/输出包	179	7.4.5	常用绘图方法	241
6.4.1	流	179	7.5	综合示例	244
6.4.2	基本输入/输出流	181	7.6	Java 类单元测试	249
6.4.3	File 类	184	7.6.1	Junit 的安装	249
6.4.4	文本文件读/写	187	7.6.2	创建项目 junitTest	249
6.4.5	字节流文件	189	7.6.3	创建被测试的类	251
6.4.6	过滤流	190	7.6.4	创建用于单独测试的 TestCase	253
小结		195	7.6.5	运行 TestCase	257
习题		195	7.6.6	创建用于集成测试的 TestSuite	259
第 7 章	图形用户界面	201	7.6.7	运行 TestSuite	260
7.1	图形用户界面组件	201	小结		261
7.1.1	图形用户界面概述	201	习题		261
7.1.2	AWT 容器	201	第 8 章	多线程程序设计	264
7.1.3	框架	203	8.1	线程概述	264
7.1.4	布局管理器	204	8.1.1	继承 Thread 类	265
7.1.5	Swing 概述	205	8.1.2	实现 Runnable 接口	266
7.2	Frame 容器加入基本组件	205	8.2	线程控制	267
7.2.1	按钮	205	8.2.1	线程状态及其转换	267
7.2.2	标签	207	8.2.2	创建状态	268
7.2.3	文本框和多行文本框	209	8.2.3	就绪状态	268
7.2.4	复选框	211	8.2.4	等待状态	269
7.2.5	单选框	212	8.2.5	撤销状态	269
7.2.6	下拉式列表	213	8.2.6	线程类其他常用的方法	270
7.2.7	滚动条	215	8.2.7	线程的同步	270
7.2.8	组合布局示例	216	8.2.8	线程通信	272
7.3	事件与事件处理	221			
7.3.1	Event 对象	222			

小结	275	10.1 网络编程概述	313
习题	275	10.1.1 网络编程的概念	313
第 9 章 Java 小应用程序	278	10.1.2 Internet 网络 通信协议	313
9.1 Applet 概述	278	10.2 Java 网络包	314
9.1.1 Eclipse 下 Applet 开发步骤	278	10.2.1 获取 IP 地址	314
9.1.2 创建 HTML 文件	280	10.2.2 URL 编程	315
9.1.3 JDK 下使用 appletviewer 运行 JavaWorld.htm	280	10.2.3 套接口类	321
9.1.4 使用浏览器观察 JavaWorld.htm 运行	280	10.2.4 使用 TCP 的 Socket 网络编程	324
9.1.5 Applet 类的方法	281	小结	338
9.1.6 Applet 的 4 种常用方法	281	习题	339
9.1.7 Applet 的安全	282	*第 11 章 Java 与数据库连接 技术 JDBC	340
9.2 Applet 的应用	282	11.1 概述	340
9.2.1 利用 Applet 接收从 HTML 网页中传递过来 的参数	282	11.1.1 JDBC 的基本功能	340
9.2.2 利用 Applet 显示图像	285	11.1.2 JDBC 接口	340
9.2.3 利用 Applet 播放声音	288	11.2 JDBC 应用	343
9.2.4 网页跳动的文字	292	11.2.1 数据库建立连接	343
9.2.5 网页中的动画	296	11.2.2 执行查询语句	344
9.3 Applet 的事件及其处理	302	11.2.3 检索结果集	347
9.3.1 Applet 中的鼠标操作	302	11.2.4 更新数据库操作	348
9.3.2 Applet 中的键盘操作	306	11.2.5 参数的输入和输出	349
小结	311	11.3 综合应用	351
习题	311	小结	356
第 10 章 网络编程基础	313	习题	357
		参考文献	358

第1章 计算机语言概述

在计算机技术发展过程中，出现过各种各样的程序设计语言。从程序设计语言的发展过程来看，可分为机器语言、汇编程序设计语言、高级程序设计语言（如 BASIC、FORTRAN、PASCAL、C、C++、Prolog、Java、C#等）、数据库管理语言（如 SQL）、网络数据交换语言（如 XML）等。高级程序设计语言分为面向过程的语言和面向对象的语言。面向过程的语言适合描述问题的解决过程：复杂问题分解成相对简单的问题、从上而下或从下而上分析构建系统。面向对象的语言使用面向对象的方法，对分析、设计同样的概念（类、对象、消息）具有独到的优点。

1.1 机器语言与汇编语言

仅由硬件组成的计算机只能接收由“0”和“1”组成的二进制信息。要计算机执行一定的操作，就要编写一系列的二进制代码。这种不需翻译即可由计算机直接执行的指令叫做机器指令，这些指令的集合叫做机器语言。每一条机器指令都是一串二进制代码，因此要记住每一条指令及其含义十分困难，编写出来的程序难以阅读，而且由于它完全依赖于硬件系统，不同的机器有不同的指令系统，因此它不具有兼容性。一台机器上编制的程序在另一台机器上根本无法运行，一个问题要在多个机器上求解，就必须重复地编写多个应用程序。这种程序直观性差，难以编写、调试、修改、移植和维护，因此人们期待用更接近于自然语言与数学语言的语言代替机器语言，汇编语言应运而生。

汇编语言是一种面向机器的程序设计语言，它用符号表示机器指令，例如用 ADD 代替机器语言中的加法运算。这种语言编写的程序不能直接运行，要经过编译程序翻译成机器语言才能运行，一般来说汇编语言指令与机器语言指令之间是一一对应的。由于汇编语言一般都是为特定计算机或计算机系统设计的，因此它虽然比机器语言好学、便于记忆，比用机器码编写程序方便，但仍然没有解决对硬件依赖的问题。

1.2 高级程序设计语言

计算机高级程序设计语言诞生以来，它与计算机硬件、计算机应用、计算机网络相互依存，一同得到了空前的发展。目前比较流行的高级语言有以下几种。

1. FORTRAN 语言

FORTRAN 语言是世界上最早出现的高级语言，它所支持的数据类型比较丰富，特别适合

于科学和工程中的数值计算。

一个 FORTRAN 程序由一个主程序与若干个子程序组成。主程序与每一个子程序都是一个独立的程序单位，称为一个程序模块。在 FORTRAN 语言中，子程序是实现模块化的有效途径。

2. ALGOL 语言

ALGOL 语言是另一种早期研制出来的高级语言，它有严格的文法规则。ALGOL 是一种分程序结构的语言，分程序的结构可以是嵌套的，即分程序内可以含有别的分程序。过程也可以看成是一个分程序，这个分程序可以在别的分程序中被调用。

3. COBOL 语言

COBOL 语言是专门为处理商业事务而设计的一种通用语言，应用于商业和管理领域。其特点是针对商务和管理过程中要处理大量记录形式数据的特殊需要，侧重于数据项的输入、输出记录的处理。

COBOL 语言把数据描述引入语言，数据描述与具体计算机无关，这一点对后来数据库管理系统的发展产生了重大影响。这种语言至今仍在商务处理领域得到一定的应用。

4. PASCAL 语言

PASCAL 语言是最早出现的结构化程序设计语言，在 PASCAL 语言中分程序和过程这两个概念合二为一，统一为过程，PASCAL 过程可以嵌套和递归。用它编写出来的程序结构性好，易于阅读。这种语言广泛用于一些计算机教学和科学计算。

5. C 语言

C 语言是 20 世纪 70 年代发展起来的一种通用的结构化程序设计语言，在系统软件开发上应用很广，它具有高级语言的诸多特点，也具有汇编语言控制硬件的能力。

字符、整数和浮点数是 C 语言的基本数据对象，用户可以构造指针、数组和联合等复杂的数据类型。C 语言是一种较低级的语言，它提供了指针和地址操作的能力以及书写结构良好程序所需的控制结构。C 语言与 UNIX 操作系统紧密相关，UNIX 操作系统及其上的许多软件都是用 C 语言编写的。

上述语言主要擅长描述计算机问题的解决过程，因此称为面向过程的语言，也称为第三代语言（3GL）。

6. 面向对象程序语言 C++

C++ 语言是在 C 语言的基础上发展起来的，与 C 语言兼容，但大幅扩充，改善了 C 语言的性能。在 C++ 语言中，最主要的是增加了类功能，使其成为面向对象的程序设计语言。一般认为，面向对象程序语言至少包含下面一些概念：

① 对象：对象是世界上客观存在的事物，它具有特征和方法两方面的属性。对象是相互关联的，通过这种关联来相互影响（改变对象的属性）。面向对象语言把对象属性和方法封装在一起，对象内部才能对对象的“私有数据”进行操作，对象外部只能按提供的公开的方法，才能查询和修改对象的属性。这样，对象属性的具体表示和方法的具体实现都是隐蔽的。

② 类：类是对象共同特征和行为的描述，对象是类的实例。

③ 继承：对象有自己的属性，但是对象更有共性，共性存在于特性之中。继承反映了上述对象共性与自己的属性的关系。子类继承父类的意思是子类具有父类的属性和行为。子类又可以对父类的特征和行为重定义或进行部分修改。

④ 消息：一个对象利用自己或其他对象的方法改变自己或其他对象的属性，这些方法的引用叫对象间消息的交互。

7. 纯面向对象的程序设计语言 Java

Java 是一种跨平台的、适合于分布式计算机环境的面向对象的编程语言。它具有可移植、稳定、简单、高性能、动态执行等特性。

① Java 语言完全面向对象：C++等面向对象的程序设计语言是由原来面向过程的语言改造而来的，而 Java 是纯面向对象的程序设计语言。

② 移植性：程序的移植性指的是程序不经修改而在不同硬件或软件平台上运行的特性。

③ 稳定性：Java 不支持指针数据类型，它提供了字节码的检查机制，还提供了自动的“内存垃圾”收集功能，这些机制大幅提高了 Java 的稳定性和安全性。

④ 简单性：Java 避免了使用 C++等语言的操作符重载、多继承和数据类型自动转换等，这些都有利于语言简单化。

⑤ 高性能：Java 通过支持多线程、高效的字节码来提高语言性能。

⑥ 动态特性：Java 的动态特性是其面向对象设计的延伸，这种特性使得 Java 程序能够适应不断变化的执行环境。Java 通过“滞后联编”，支持类库升级，相应的应用程序不必重新编译，也一样可以利用升级后类库的新增功能。Java 的动态特性还体现在其对动态数据类型和动态协议的支持上。利用一种特殊的 Applet，编程人员可很方便地支持新的数据类型。Java 的动态性的价值是使程序员真正拥有“即插即用”（Plug-and-Play）的软件模块功能。

⑦ 分布式：Java 支持数据分布和操作分布。

8. 函数式程序设计语言

函数式程序是由一些原始函数、定义函数等组成的函数表达式。传统程序设计语言中的赋值等概念，在函数式程序设计语言中消失。函数式程序的一个最本质的特性，就是函数值唯一地由其参数值所确定。只要使用相同的参数值，对此程序的不同的调用总是得到相同的结果。这种性质称为引用透明性，有助于程序的模块化。

一般认为表处理语言（LISP）是最早的函数式程序设计语言。但是，LISP 的重点是将函数应用于对象，以产生新的对象，必要时再上升为函数。巴克斯所提出的函数式程序设计，则是引用函数型产生新函数，程序设计时从一般的对象空间上升到函数空间，因而具有优越的数学性质，有助于程序的理解、推理和验证。

函数式程序设计语言具有简明性和独特的表达能力，可用它来研究传统程序设计语言的语义。一种方法是用于确定一个解释程序的定义，作为被研究的语言的语义；另一种方法是将被研究的语言写成的程序转换成与之等价的函数式程序。在人工智能领域中，需要用复杂的算法去处理一些复杂的（通常是符号的）数据结构。LISP 语言成功地应用于这一领域，说明了函数式程序设计的独特优越性。巴克斯分析了传统程序设计语言的缺陷，认为这些缺陷主要是由于冯·诺依曼系统结构所造成的。他所提出的函数式程序设计（简称 FP），摆脱了传统的冯·诺依曼计算机结构，需要一种新的非冯·诺依曼式的系统结构为后援。一些具有新概念的计算机，

如归约机、数据流机以及专为某种函数式语言设计的计算机正在研究和发展中。现代既需要研究在冯·诺依曼式计算机上如何更有效地实现函数式程序设计语言的问题,也需要研究适应这种语言的新型计算机结构。

函数式程序设计受到重视的原因是:首先由于产生了“软件危机”,人们企图探讨一种摆脱这种困境的新型程序设计方式,而函数式程序设计具有不少独特之处。其次,超大规模集成电路技术的发展,为发挥函数式程序设计语言的潜在并行性提供了物质基础。可以预期,一些具有诸如高度并行性等特点的非冯·诺依曼式计算机将会出现。随着硬件技术的发展、软件方法的研究以及应用范围的不断扩大,函数式程序设计将得到发展,并在新一代计算机系统中起到重要的作用。

9. 脚本语言

脚本语言是 Web 的普及与发展的产品,一般指 HTML、VBScript、JavaScript、Java Applet 等。XML 是定义 HTML 的元语言,它也可属于脚本语言范围。脚本语言程序由客户代理(浏览器)解释、运行。脚本语言标准化主要是为了实现网页独立于网页浏览器,使客户与浏览器更友好地交互,动态控制网页元素。

10. 描述性语言

软件人员都希望有朝一日,只要描述系统的需求、设计、使用平台,就可以生成系统代码,而不需要编程。许多研究机构、大的公司在这方面已迈出了非常重要的一步。模型驱动结构 MDA 就是这一领域研究成果的体现,它以模型的观点研究软件系统,其中 UML(通用建模语言)已成为人们描述软件系统开发不同阶段使用的语言。

传统的程序设计过程是抽象思维过程,只有程序设计完成后,通过上机调试,方知程序的正确与错误。可视化程序开发环境,使程序一边开发,一边就能看到程序运行的结果,大幅提高了软件的开发效率。可视化程序开发是以面向对象程序设计为基础的,可视化面向对象的程序设计方法已得到普遍的认可。许多传统的非面向对象语言都经过扩充、改造成为支持可视化的面向对象的程序设计语言。例如,PASCAL 演化为 Delphi, BASIC 演化为 VB, C 语言演化为 VC++、C++Builder 等。这些可视化集成开发环境成为当前软件开发环境的主流。

1.3 语言翻译程序

计算机只能识别、运行本身的机器语言(0、1 构成的指令系统),另一方面,人们又希望使用高级程序设计语言描述要计算机解决的问题。因此,如何使高级程序设计语言程序映射成等价的机器语言就成为解决问题的关键,语言翻译程序就能起到这种作用,如图 1-1 所示。

按语言程序的翻译方式不同,语言翻译程序分为解释型翻译程序和编译型翻译程序。

解释型翻译程序的工作方式是,读入一句源程序,翻译一句,执行一句,这样反复操作直到最终完成全部源程序的翻译。BASIC 语言是典型的解释型语言。

编译型翻译程序也叫编译程序,它的工作方式是,加工整个源程序,最终翻译成机器语言

代码，然后交给计算机执行。编译程序有利于目标程序的优化，也有利于提高目标程序的运行速度。FORTRAN 语言、C 语言、PASCAL 语言都是编译型语言。

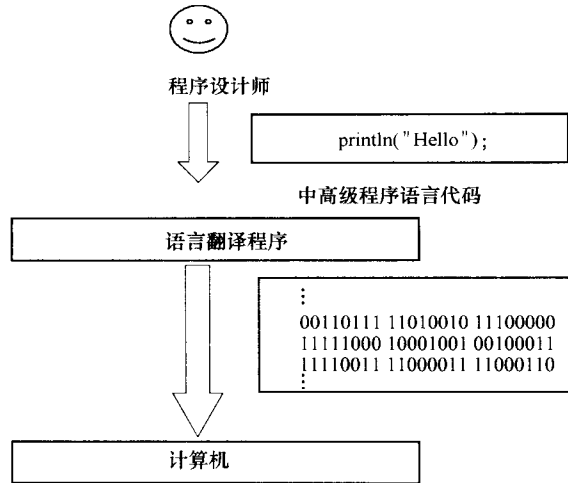


图 1-1 语言翻译程序的作用示意图

有时人们形象地把解释型翻译程序比做“口译”，把编译程序比做“笔译”。由此可见，由于计算机有了一种语言的翻译程序，因此可使其能“明白”用这种语言编写的程序。因此，配置了某种语言的翻译程序的计算机又被人们称为这种语言的虚拟机。不言而喻，语言翻译程序与操作系统和计算机硬件有关，由翻译程序产生的目标程序也与操作系统和计算机硬件有关。不同操作系统下同一种语言的语言翻译程序是不一样的；同一操作系统下（如 Windows 环境）硬件不一样（如 x86 和 PowerPC），同一种语言的翻译程序也不一样，如图 1-2 所示。

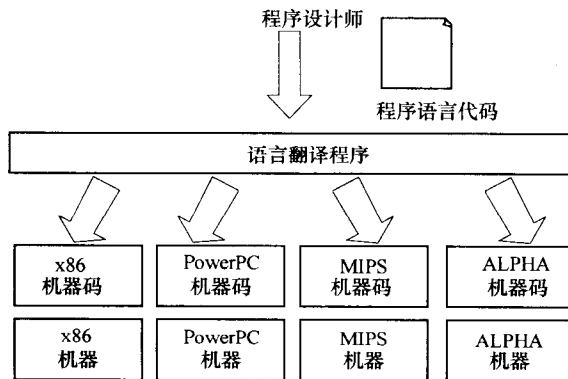


图 1-2 语言翻译程序与机器硬件有关

1.4 虚拟机、Java 虚拟机与 Java 运行环境

由于编译程序与操作系统和计算机硬件有关，因此为了提高编译程序的可移植性，人们提出了虚拟机的理论。虚拟机好似通用的计算机，有自己的指令系统，但本身没有实际的硬件。为了可以执行虚拟机代码，必须有虚拟机实时运行支持系统把虚拟机代码转换成相应硬件的机器代码，然后加以执行，如图 1-3 所示。有了虚拟机，编译程序首先把高级语言程序翻译成虚拟机代码，这样就能大幅提高编译程序的可移植性。虚拟机代码与机器无关，它不仅可以在本机上执行，也可以通过网络传输到其他配置了相应的虚拟机实时运行支持系统的网点上执行。这一特征，显示了虚拟机在网络时代的无限生机。

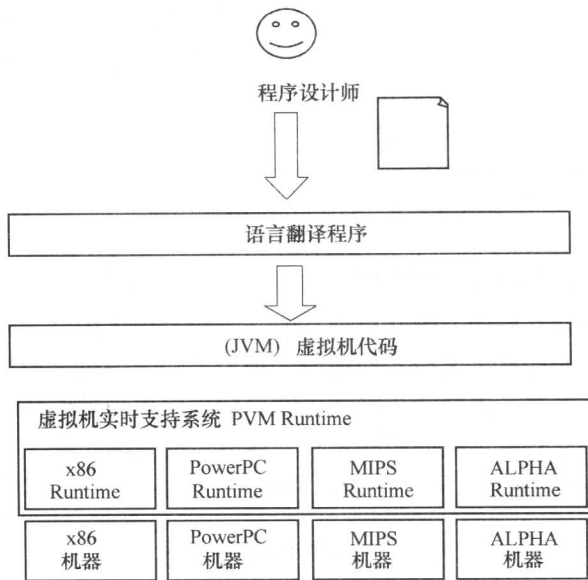


图 1-3 虚拟机原理示意图

计算机网络的发展对计算机语言不断提出新的要求，特别是要求语言具有可移植性、安全性，这正是 Java 成为网络开发主流语言的原因。

JVM (Java Virtual Machine) 是一种 Java 虚拟机，从结构上看它与实际的计算机相似，其作用是使得一台实际的计算机能够运行 Java 字节码程序。Java 字节码是 Java 源程序编译后的程序，它不能被计算机直接执行，但可以被所有的 Java 虚拟机执行，这就是 Java 字节码程序可以在网络上移植的原因。

Java 程序必须有自己的运行环境 (Java 平台)。一个 Java 运行环境包括实际计算机、适配

器、Java 虚拟机、Java 基本软件和 Java 应用程序接口，如图 1-4 所示。JVM 是 Java 运行环境的核心，JVM 的下层是和实际计算机的接口，被称为适配器，不同类型的计算机其适配器是不同的。Java 的基本软件也称为基本类，而 Java 应用程序接口（Application Program Interface, API）是已编译好的程序代码库，可以直接使用，以节约编程时间。事实上，Java 的基本类和 API 的规模并不固定，许多 Java 平台对这两部分内容进行了补充，即除了基本类外，还有扩展类。

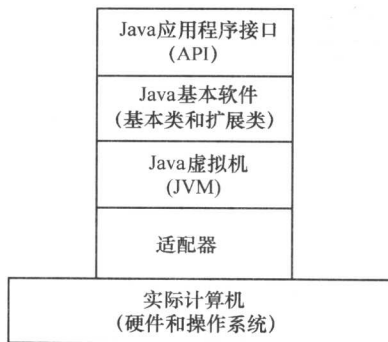


图 1-4 Java 运行环境

由此可见，Java 虚拟机与操作系统和计算机硬件无关。Java 经过编译后的字节码程序可以在网络上传输到任何支持 Java 虚拟机的网点机上运行。

利用 Java 语言可以开发两种形式的应用程序：Java 应用程序（Java Application）和 Java 小应用程序（Java Applet）。Java 小应用程序是借助浏览器运行的程序。

1.5 Java 程序的运行

Java 程序是半编译半解释型语言，其程序的运行过程与编译型和解释型语言都不同。

首先，编译程序将 Java 源程序编译成与实际计算机无关的字节码，然后 Java 运行系统解释并执行字节码。Java 编译系统和运行系统的功能如图 1-5 所示。其中，运行系统解释 Java 应用程序的执行过程分三步进行。

第一步由类装载器完成字节码的装载，装载时运行系统确定程序的内存分配。

第二步“字节码检验器”对字节码进行安全性检查，这种检查可以排除字节码可能存在的违法访问权限、不规范数据类型和非法调用等问题。

第三步完成字节码的翻译和执行。这里有两种途径：一种是走运行系统左边的路径，通过“解释器”将一个个字节码翻译成机器码，然后由“实时运行部件”立即将机器码送到硬件执行，这是一种解释工作方式，边翻译边执行，Java 系统一般采用这种方式；另一种是走运行系统右边的路径，通过“代码生成器”先将所有字节码翻译成适用于本系统的机器码，然后送硬件执