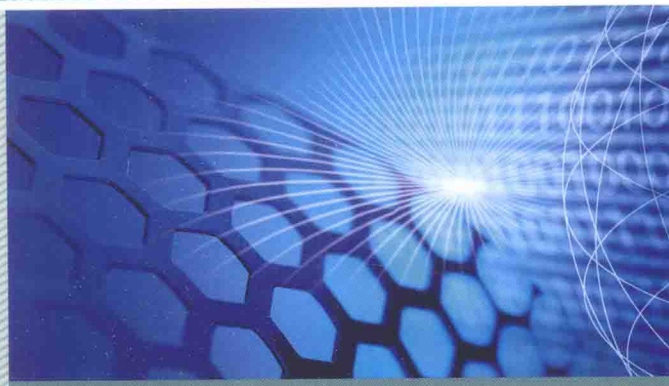


Java 语言程序设计 实用教程

◎ 陈艳平 徐受蓉 主编

Java YUYAN CHENGXU SHEJI
SHIYONG JIAOCHENG



Java 语言程序设计实用教程

主 编 陈艳平 徐受蓉

副主编 (排名不分先后)

董 明 尚 晋

张 岚 胡秀丽

 **北京理工大学出版社**
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

本书从 Java 程序开发必备的能力出发,将教学内容分为三个单元,即语言基础、技术基础和技能基础,书中所有章节依次贯穿这三个单元。每一章均按照相应知识点储备、案例分析、任务训练、知识拓展、思考与练习等环节进行组织。书中案例以通俗易懂、实用为原则,在组织形式上通过任务驱动、效果演示来激发学生兴趣,从而将知识融入任务之中。书中内容涵盖了 Java 语言概述、数据类型与运算符、流程控制结构、数组与字符串、面向对象程序设计、异常处理、输入/输出及文件处理、多线程、图形用户界面、数据库编程等。

本书注重理论与实践的统一,以实用案例为载体,按照任务驱动的方式进行编排。全书内容充实、案例易懂、讲解清晰,能使初学者在短时间内掌握 Java 的核心概念与技术。本书既可以作为高等院校计算机类专业及相关专业的教材,也可供广大计算机爱好者阅读和参考。

版权专有 侵权必究

图书在版编目(CIP)数据

Java 语言程序设计实用教程 / 陈艳平, 徐受蓉主编. —北京: 北京理工大学出版社, 2015. 6

ISBN 978 - 7 - 5682 - 0745 - 4

I. ①J… II. ①陈…②徐… III. ①JAVA 语言 - 程序设计 - 教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 134218 号

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (总编室)

(010) 82562903 (教材售后服务热线)

(010) 68948351 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 三河市天利华印刷装订有限公司

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 20.5

字 数 / 481 千字

版 次 / 2015 年 6 月第 1 版 2015 年 6 月第 1 次印刷

定 价 / 56.00 元

责任编辑 / 王玲玲

文案编辑 / 王玲玲

责任校对 / 周瑞红

责任印制 / 李志强

图书出现印装质量问题,请拨打售后服务热线,本社负责调换

前 言

Java 语言是目前流行的一种网络编程语言，它的安全性、平台无关性等特点给编程人员带来了一种崭新的设计理念。如今很多的流行技术，例如 Android 技术等，都和 Java 语言有着直接的联系，可以说，学好 Java 语言是成为一个优秀的软件开发工程师的基本要求。

Java 语言程序设计是高等院校计算机类专业的核心课程之一。为适应 IT 行业的迅速发展和课程改革的迫切需要，本书作者与企业专家基于行业技术进行了深入合作，以实用任务为载体，并结合职业资格认证考试，编写了这本知识全面、内容适度、技术先进的理论和实践一体化的教材。

本书从 Java 程序开发必备的能力出发，根据技术的发展和学生认知规律，将教学内容分为三个单元：语言基础、技术基础和技能基础。书中所有章节均按照相应知识点储备、案例分析、任务训练、知识拓展、思考与练习等环节进行组织。每一章节中的案例以通俗易懂和实用为原则，在组织形式上通过任务驱动、效果演示来激发学生兴趣，从而将知识融入任务之中。本书能很好地指导学生进行实践，有利于学生理解和巩固知识，并在实践中培养技术应用能力。

本书由陈艳平、徐受蓉担任主编，董明、尚晋、张岚、胡秀丽担任副主编。具体编写分工为：张岚、胡秀丽编写第 1、2、6 章，尚晋编写第 3 章，陈艳平编写第 4、5、9 章，董明编写第 7、8 章，徐受蓉编写第 10 章。本书在编写过程中，得到了重庆市酷智软件开发公司张晋先生的技术指导，在此表示衷心的感谢。

本书配套的电子教案、课件、课程标准、源代码、试题样卷、开发环境软件等素材，可通过登录北京理工大学出版社网站的下载专区免费下载，也可邮件联系编者（htdbteam@163.com）进行索取。

由于编者水平有限，书中疏漏之处在所难免，敬请读者批评指正。

编 者

目 录

第1章 Java 语言概述	1
1.1 软件开发基础	1
1.1.1 软件运行原理	2
1.1.2 软件开发流程	2
1.1.3 程序设计语言	4
1.2 Java 语言	5
1.2.1 Java 语言的发展	5
1.2.2 Java 语言的组成	5
1.2.3 Java 语言的版本	6
1.3 Java 开发环境	6
1.3.1 下载和安装 JDK	6
1.3.2 下载和安装 Eclipse	12
1.4 第一个 Java 程序	13
1.4.1 命令方式开发第一个 Java 应用程序	13
1.4.2 Eclipse 环境中开发第一个 Java 应用程序	15
1.4.3 Java 语言开发过程	18
1.4.4 Java 的体系结构	19
1.5 案例分析	20
1.5.1 案例情景——编写 Java Applet, 输出 “Hello, 欢迎进入精彩的 Java 世界!”	20
1.5.2 运行结果	20
1.5.3 实现方案	20
1.6 任务训练——编写简单 Java 程序	22
1.6.1 训练目的	22
1.6.2 训练内容	22
1.7 知识拓展	23
思考与练习	23
第2章 数据类型与运算符	24
2.1 常量、变量与数据类型	24
2.1.1 常量	25
2.1.2 变量	25
2.1.3 数据类型	26
2.1.4 数据类型的转换	30



2.2	运算符	32
2.2.1	算术运算符	32
2.2.2	关系运算符	33
2.2.3	逻辑运算符	34
2.2.4	位运算符	35
2.2.5	赋值运算符	36
2.2.6	条件运算符	36
2.2.7	其他运算符	36
2.2.8	运算符优先级	36
2.3	表达式	37
2.4	简单的输入/输出	38
2.4.1	输出	38
2.4.2	输入	38
2.5	编程风格	42
2.5.1	Java 语言书写规范	42
2.5.2	注释	42
2.6	案例分析	43
2.6.1	案例情景——计算圆的周长和面积	43
2.6.2	运行结果	43
2.6.3	实现方案	43
2.7	任务训练——Java 基本数据类型、运算符与表达式	44
2.7.1	训练目的	44
2.7.2	训练内容	44
2.8	知识拓展	46
	思考与练习	46
第3章	流程控制结构	48
3.1	分支结构	48
3.1.1	if 语句	49
3.1.2	switch 语句	52
3.2	循环结构	55
3.2.1	while 语句	55
3.2.2	do - while 语句	56
3.2.3	for 循环语句	57
3.2.4	多重循环	59
3.3	跳转语句	60
3.3.1	break 语句	60
3.3.2	continue 语句	61
3.3.3	return 语句	62



3.4 程序的断点调试	63
3.5 案例分析	64
3.5.1 案例情景——猜数游戏	64
3.5.2 运行结果	64
3.5.3 实现方案	65
3.6 任务训练——流程控制语句	66
3.6.1 训练目的	66
3.6.2 训练内容	66
3.7 拓展知识	67
思考与练习	68
第4章 数组与字符串	70
4.1 一维数组	70
4.1.1 一维数组的声明和创建	71
4.1.2 一维数组的初始化	72
4.1.3 一维数组的引用	73
4.2 多维数组	75
4.2.1 二维数组的定义	75
4.2.2 二维数组的初始化	75
4.2.3 二维数组的引用	76
4.2.4 数组的常用方法	78
4.3 字符串	82
4.3.1 String 类	82
4.3.2 StringBuffer 类	85
4.3.3 StringTokenizer 类	87
4.3.4 main() 方法的参数	88
4.4 案例分析	89
4.4.1 案例情景——冒泡排序	89
4.4.2 运行结果	89
4.4.3 实现方案	90
4.5 任务训练——数组与字符串的使用	91
4.5.1 训练目的	91
4.5.2 训练内容	91
4.6 知识拓展	94
思考与练习	94
第5章 面向对象程序设计	96
5.1 面向对象概述	96
5.1.1 面向对象基本概念	97
5.1.2 面向对象的基本特征	97



5.2 类	99
5.2.1 定义类	99
5.2.2 成员变量	102
5.2.3 成员方法	103
5.2.4 类的对象	106
5.2.5 构造方法	108
5.2.6 修饰符	110
5.2.7 静态属性、静态方法与静态初始化器	111
5.2.8 最终类、最终属性、最终方法与终结器	113
5.2.9 包	115
5.3 类的继承	118
5.3.1 类继承的实现	118
5.3.2 this 和 super 关键字	119
5.3.3 抽象类与抽象方法	123
5.3.4 类对象之间的类型转换	125
5.4 类的多态	127
5.4.1 方法重载	128
5.4.2 方法重写	129
5.5 接口	130
5.5.1 接口的定义	130
5.5.2 接口的实现	131
5.5.3 接口的继承	132
5.5.4 接口的多态	134
5.6 案例分析	135
5.6.1 案例情景——模拟 ATM 自动取款机	135
5.6.2 运行结果	135
5.6.3 实现方案	136
5.7 任务训练——面向对象程序设计	142
5.7.1 训练目的	142
5.7.2 训练内容	142
5.8 知识拓展	144
思考与练习	144
第6章 异常处理	146
6.1 异常和异常类	146
6.1.1 异常的定义	147
6.1.2 Java 异常类及其层次结构	147
6.2 异常处理	149
6.2.1 异常处理机制	150



6.2.2 捕获异常	150
6.2.3 声明异常	155
6.2.4 抛出异常	156
6.2.5 自定义异常类	157
6.3 案例分析	158
6.3.1 案例情景——身份证验证程序	158
6.3.2 运行结果	158
6.3.3 实现方案	159
6.4 任务训练——异常及其处理	160
6.4.1 训练目的	160
6.4.2 训练内容	160
6.5 知识拓展	162
思考与练习	162
第7章 输入/输出及文件处理	163
7.1 输入/输出流概念	163
7.2 输入/输出流类	164
7.2.1 字节流 InputStream 类和 OutputStream 类	164
7.2.2 字符流 Reader 类和 Writer 类	166
7.3 标准输入/输出	167
7.3.1 标准输入流	167
7.3.2 标准输出流	168
7.3.3 标准错误输出流	168
7.4 常用的文件处理	168
7.4.1 文件的顺序访问	168
7.4.2 文件的随机读写	170
7.4.3 目录和文件管理	171
7.5 案例分析	172
7.5.1 案例情景——读取文件到内存，在修改后输出	172
7.5.2 运行结果	173
7.5.3 实现方案	173
7.6 任务训练——文件访问	174
7.6.1 训练目的	174
7.6.2 训练内容	174
7.7 拓展知识	176
思考与练习	178
第8章 多线程	180
8.1 多线程的基本概念	180
8.2 多线程的实现机制	181



8.2.1 继承 Thread 类	181
8.2.2 实现 Runnable 接口	182
8.3 线程的状态和线程的控制	184
8.3.1 线程的状态和生命周期	184
8.3.2 线程的控制	185
8.4 线程的同步	186
8.4.1 共享受限资源	186
8.4.2 线程间的协作	187
8.4.3 线程的调度和优先级	194
8.5 案例分析	194
8.5.1 案例情景——模拟排队买票	194
8.5.2 运行结果	195
8.5.3 实现方案	195
8.6 任务训练——多线程使用	198
8.6.1 训练目的	198
8.6.2 训练内容	198
8.7 拓展知识	200
思考与练习	200
第9章 图形用户界面	202
9.1 GUI 概述	202
9.1.1 AWT 简介	203
9.1.2 Swing 简介	204
9.2 常用容器	205
9.2.1 JFrame (框架)	206
9.2.2 JPanel (面板)	208
9.3 简单 GUI 组件	209
9.3.1 标签和按钮	209
9.3.2 单行文本框和多行文本框	213
9.4 布局管理	218
9.4.1 流式布局	219
9.4.2 网格布局	220
9.4.3 边界布局	222
9.4.4 卡片布局	223
9.4.5 网格袋布局	224
9.4.6 空布局	227
9.5 事件处理	229
9.5.1 Java 事件模型	229
9.5.2 Java 事件类型	230



9.5.3 事件、监听器接口及适配器	230
9.5.4 典型事件处理	231
9.6 复杂 GUI 组件	233
9.6.1 单选按钮和复选框	233
9.6.2 列表框和组合框	238
9.6.3 菜单和工具栏	242
9.7 高级 GUI 组件	246
9.7.1 对话框	246
9.7.2 表格	250
9.7.3 树	252
9.8 案例分析：简易计算器	255
9.8.1 案例情景——简易计算器	255
9.8.2 运行结果	256
9.8.3 实现方案	256
9.9 任务训练——图形用户界面的设计	263
9.9.1 训练目的	263
9.9.2 训练内容	263
9.10 拓展知识	266
思考与练习	266
第 10 章 数据库编程	268
10.1 JDBC 编程技术概述	268
10.1.1 数据库基础知识	269
10.1.2 SQL 语言	269
10.1.3 JDBC	269
10.2 使用 JDBC 驱动程序编程	270
10.2.1 JDBC 程序模板	271
10.2.2 使用专用 JDBC 驱动程序连接数据库	272
10.2.3 执行 SQL 语句	276
10.3 数据库的基本操作	277
10.3.1 数据查询	277
10.3.2 数据添加、修改和删除	280
10.4 案例分析：用户信息管理	283
10.4.1 案例情景——用户信息管理	283
10.4.2 运行结果	283
10.4.3 实现方案	283
10.5 任务训练	288
10.5.1 训练目的	288
10.5.2 训练内容	288



10.6 拓展知识	291
思考与练习	291
思考与练习参考答案	292
第1章 Java 语言概述	292
第2章 数据类型与运算符	293
第3章 流程控制结构	294
第4章 数组与字符串	296
第5章 面向对象程序设计	300
第6章 异常处理	305
第7章 输入/输出及文件处理	307
第8章 多线程	309
第9章 图形用户界面	310
第10章 数据库编程	314
参考文献	315

第 1 章

Java 语言概述



【知识点】程序设计语言分类；Java 特点；Java 实现机制；Java 的体系结构；集成开发环境 Eclipse。



【能力点】理解 Java 实现机制；安装并掌握开发工具 JDK；掌握搭建集成开发环境 Eclipse；熟练运用集成开发环境 Eclipse 编写 Java 程序。



【学习导航】

Java 语言是 Sun 公司于 1995 年推出的面向对象程序设计语言，它集安全性、平台无关性等特性于一身。在网络的发展和智能技术的冲击下，Java 语言得到了更广泛的应用。本章内容在 Java 程序开发能力进阶必备中的位置如图 1-0 所示。

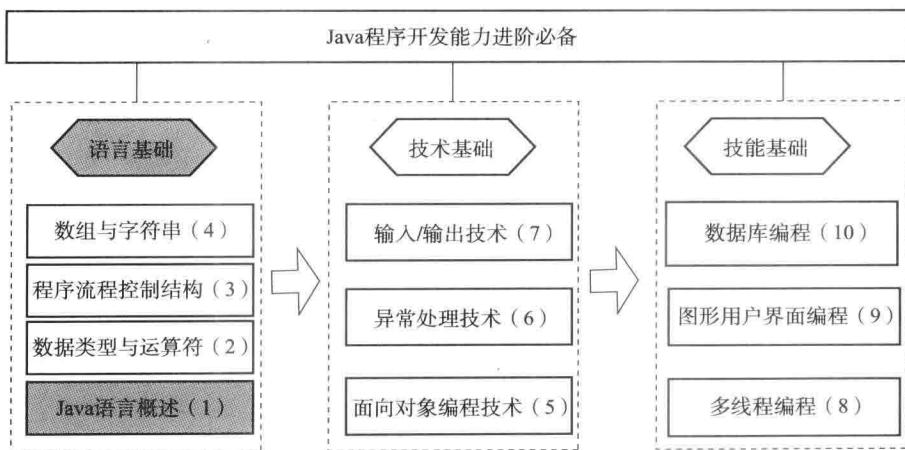


图 1-0 本章内容在 Java 程序开发能力进阶必备中的位置

1.1 软件开发基础

学习程序设计语言之前，需要对软件运行原理有所理解，并对软件开发过程有所了解，这对于初学者尤为重要。



1.1.1 软件运行原理

计算机是人类 20 世纪最伟大、最重要的发明之一，其伟大之处就在于它能够以惊人的效率和前所未有的智能化来辅助人们更好地完成认识自然和改造自然的工作。它是有史以来第一种能够完成真正意义上的复杂的“学习”功能的机器，这就使得它具有了某种更接近人类的“思考”能力；与此同时，计算机所特有的超人的计算能力可以把人们的工作效率和生产效率提升成千上万倍，从而把人从最直接、最原始的生产第一线上解放出来，转而从事使用和操纵计算机的工作。

计算机是由不同部分组成的非常复杂的系统，如果把它比作是一项工作，硬件工程师将负责其身体各部分健康、完好；软件工程师将教会它如何学习和工作；计算机的操纵人员将向这个身体健康并学有所长的“工人”布置任务并监督其保质保量地完成。换句话说，计算机由硬件工程师赋予生命，由软件工程师注入灵魂，并最终在千千万万的操作人员手中发挥威力和作用。计算机由中央处理单元、算术逻辑单元、内存单元、输入单元、输出单元和外存单元组成。计算机的基本原理是存储程序和控制程序，开发人员预先把指挥计算机如何进行操作的指令序列（称为程序）和原始数据通过输入设备输送到计算机内存储器中。每一条指令中明确规定了计算机从哪个地址取数，进行什么操作，然后送到什么地址去等步骤。软件在运行之前将指令保存到内存中的过程称为内存加载或调入内存，这个内存加载的步骤是由 CPU 执行的。加载成功之后，CPU 将从内存中依次取出该软件程序的每一条指令并顺序执行。在执行过程中，CPU 可能需要内存中这个软件或其他软件的数据，可能需要调动输入、输出单元完成输入、输出操作，也可能要调度它的软件指令配合工作。这一切，都取决于开发人员事先编写好并已经加载到内存中的程序指令。计算机系统软件运行的基本原理如图 1-1 所示。

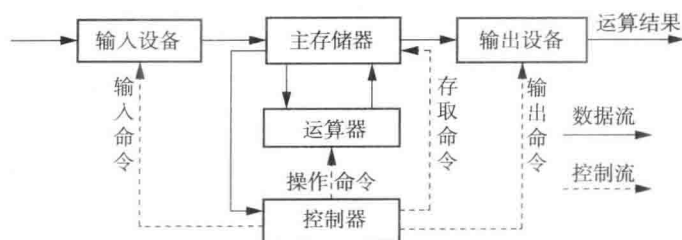


图 1-1 计算机工作原理图

程序设计语言是软件开发人员与计算机进行沟通和交流的工具，是计算机能够识别的语言。只有掌握了程序设计语言，软件开发人员才能指挥计算机按照自己的意志完成种种复杂的工作。

1.1.2 软件开发流程

由于软件系统被划分成操作系统软件、系统软件和应用软件，从事软件开发的人员也进行了相应的分工。操作系统软件是硬件裸机和其他软件或用户之间的必由接口，它的性能将决定整个计算机系统的性能，所以其开发要求很高，需要精深的专业知识与技能，正因如此，相应的从业人员也最少。系统软件是操作系统软件和应用软件之间的接口，从事系统软件开发一方面需要开发人员对操作系统有足够深入的了解，以便能充分利用操作系统提供的服务；另一方



面,系统软件自身也需要为其上的应用软件提供方便、充分的服务,使应用软件可以不必了解操作系统的细节而直接使用系统软件的功能。从事系统软件开发的人员也较少。应用软件针对某个具体问题或实体,所以功能的专用性最强,软件间的差异性最大,开发的需求量最大,从业人员的人数也最多。开发操作系统软件或系统软件多注重于软件的性能、效率,而开发应用软件则注重用户的需求,即充分研究应用软件的最终用户和操作人员希望这个软件具有何种功能,能解决何种问题,并在明确需求的基础之上再去寻找一个能满足这个需求的解决方案:是直接将系统建筑在操作系统之上,还是寻找一个合适的系统软件作为基础,选择何种计算结构等。无论何种情况,开发人员都需要对即将研发的应用软件所立足的基础层次有足够的了解,并掌握这个层次的相应开发工具,为此,我们需要对软件开发流程有所了解。

软件开发流程(Software Development Process)即软件设计思路和方法的一般过程,包括设计软件的功能和实现的算法和方法、软件的总体结构设计和模块设计、编程和调试、程序联调和测试以及编写、提交程序。软件开发流程如图1-2所示。

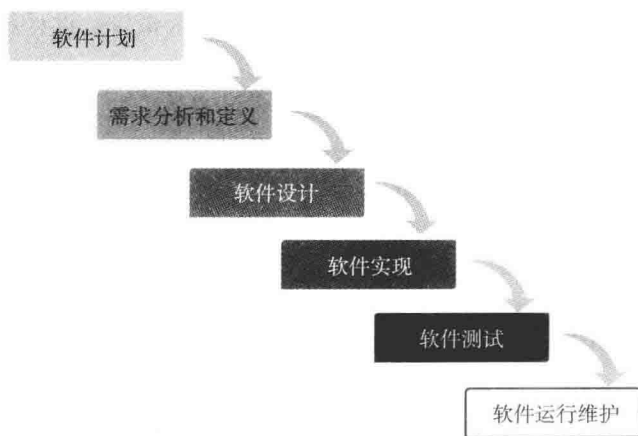


图1-2 软件开发流程

1. 需求调研分析

● 相关系统分析员和用户初步了解需求,然后用 Word 列出要开发的系统的大功能模块,以及每个大功能模块有哪些小功能模块。对于有些需求比较明确的相关界面,在这一步里面可以初步定义好少量的界面。

● 系统分析员深入了解和分析需求,根据自己的经验和需求用 Word 或相关的工具再做出一份系统的功能需求文档。这次的文档会清楚地列出系统大致的大功能模块,大功能模块有哪些小功能模块,并且还列出相关的界面和界面功能。

● 系统分析员和用户再次确认需求。

2. 概要设计

首先,开发者需要对软件系统进行概要设计,即系统设计。概要设计需要对软件系统的设计进行考虑,包括系统的基本处理流程、系统的组织结构、模块划分、功能分配、接口设计、运行设计、数据结构设计和出错处理设计等,从而为软件的详细设计提供基础。

3. 详细设计

在概要设计的基础上,开发者需要进行软件系统的详细设计。详细设计需要描述实现具



体模块所涉及的主要算法、数据结构、类的层次结构及调用关系,需要说明软件系统各个层次中的每一个程序(每个模块或子程序)的设计考虑,以便进行编码和测试,还应当保证软件的需求完全分配给整个软件。详细设计应当足够详细,使开发者能够根据详细设计报告进行编码。

4. 编码

在软件编码阶段,开发者根据《软件系统详细设计报告》中对数据结构、算法分析和模块实现等方面的设计要求,开始具体的程序编写工作,分别实现各模块的功能,从而实现对目标系统的功能、性能、接口、界面等方面的要求。

5. 测试

测试编写好的系统。交付给用户使用,用户使用后一一地确认每个功能。

6. 软件交付准备

在软件测试证明软件达到要求后,软件开发者应向用户提交开发的目标安装程序、数据库的数据字典、《用户安装手册》、《用户使用指南》、需求报告、设计报告、测试报告等双方合同约定的内容。《用户安装手册》应详细介绍安装软件对运行环境的要求、安装软件的定义和内容、安装软件在客户端和服务端及中间件的具体安装步骤、软件安装后的系统配置。《用户使用指南》应包括软件各项功能的使用流程、操作步骤、相应业务介绍、特殊提示和注意事项等方面的内容,在需要时还应举例说明。

7. 验收

用户验收。

1.1.3 程序设计语言

程序设计语言是能够被计算机和编程人员双方所理解和认可的交流工具。当软件开发人员希望计算机完成一件工作,或解决一个问题时,他(她)首先需要把这个问题的实质彻底研究清楚,确定解决问题的方法和步骤;然后再把这个方法和步骤用计算机能够理解和执行的程序设计语言表述出来,形成一组语句的集合,即程序。

程序设计语言并不唯一,在计算机技术发展的 50 年中,先后形成了数百种不同的程序设计语言。按照其发展历史,程序设计语言按其级别可以划分为机器语言、汇编语言和高级语言、第四代语言四大类。

1. 第一代语言(机器语言)

机器语言就是计算机的指令系统,用机器语言编写的程序可以被计算机直接执行。由于不同类型计算机的指令系统(机器语言)不同,因而在一种类型计算机上编写的机器语言程序,在另一种类型的计算机上也可能无法运行。机器语言程序全部用二进制(八进制、十六进制)代码编制,人们不易记忆和理解,也难于修改和维护,所以现在已不用机器语言编制程序了。

2. 第二代语言(汇编语言)

汇编语言用助记符来代替机器指令的操作码和操作数,如用 ADD 表示加法、SUB 表示减法、MOV 表示传送数据等。这样就能使指令使用符号表示而不再使用二进制表示。用汇编语言编写的程序与机器语言程序相比,虽然可以提高一点效率,但仍然不够直观简便。



3. 第三代语言（高级语言）

高级语言是面向用户的、基本上独立于计算机种类和结构的语言。其最大的优点是：形式上接近于算术语言和自然语言，概念上接近于人们通常使用的概念。高级语言的一个命令可以代替几条、几十条甚至几百条汇编语言的指令。因此，高级语言易学易用、通用性强、应用广泛。从描述客观系统来看，程序设计语言可以分为面向过程语言和面向对象语言。

（1）面向过程语言

以“数据结构 + 算法”程序设计范式构成的程序设计语言，称为面向过程语言。前面介绍的程序设计语言大多为面向过程语言。

（2）面向对象语言

以“对象 + 消息”程序设计范式构成的程序设计语言，称为面向对象语言。目前比较流行的面向对象语言有 Delphi、Java 和 C++ 等。

4. 第四代语言（简称 4GL）

4GL 是非过程化语言，编码时只需说明“做什么”，不需描述算法细节。数据库查询是 4GL 的典型应用。用户可以用结构化查询语言（SQL）对数据库中的信息进行复杂的操作，如用户只需将要查找的内容在什么地方、根据什么条件进行查找等信息告诉 SQL，SQL 将自动完成查找过程。第四代程序设计语言是面向应用，为最终用户设计的一类程序设计语言。它具有缩短应用开发过程、降低维护代价、最大限度地减少调试过程中出现的问题以及对用户友好等优点。

1.2 Java 语言

Java 语言是一种简单的，面向对象的，分布式的，具备解释性、健壮性、安全与系统无关性、可移植性的，高性能、多线程的动态语言。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 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。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。

1.2.1 Java 语言的发展

Java 的前身是 Sun Microsystems 公司开发的一种用于智能化家电的名为“橡树”（Oak）的语言。Oak 语言当时几近失败，直到 1993 年才随着 WWW（万维网）的迅速发展而重现生机。Sun 公司发现可以利用这种技术创造含有动态内容的 WWW 网页，便组织人力对其进行重新开发和改造。1995 年 5 月 23 日，Java 这种定位于网络应用的程序设计语言被正式推出，自那以后，Java 逐步从一种单纯的高级编程语言发展成为一种重要的基于 Internet 的开发平台，并进而带动了 Java 产业的发展和壮大，成为当今计算机不可忽视的力量和重要的发展潮流。

1.2.2 Java 语言的组成

Java 语言由语法规则和类库两部分组成。语法规则确定了 Java 程序的书写规范；类库，或称为运行时库，则提供了 Java 程序与运行它的系统软件（Java 虚拟机）之间的接口。如