高等学校计算机专业
“十二五”规划教材

面向对象程序设计—Java
（第三版）

张白一　崔尚森　张　辰　编著

西安电子科技大学出版社
http://www.xduph.com

高等学校计算机专业“十二五”规划教材

面向对象程序设计
——Java

(第三版)

张白一　崔尚森　张辰　编著

西安电子科技大学出版社

内 容 简 介

本书将面向对象的理论与Java语言程序设计技术相结合，意在培养读者正确运用面向对象的思维方法分析问题和解决问题的能力。全书共分16章。第1章介绍了编程语言的发展、Java语言的特点和NetBeans集成开发环境。第2章～第6章主要介绍了面向对象的基本理论、原理、技术方法和Java语言基础知识，阐述了面向对象程序设计的基本原则和特点。从第7章开始的以后各章介绍了Java的常用标准类库及编程技巧，主要包括字符串类、集合类、GUI设计、Swing组件、异常处理、多线程技术、输入/输出技术、网络编程技术和JDBC数据库应用编程技术等。

本书可作为高等院校计算机类、软件工程类、信息类专业相关课程的教材，也可作为对面向对象编程技术和Java语言感兴趣的读者的自学用书。

为方便教学和实践，西安电子科技大学出版社网站上提供了可免费使用的电子教案和示例程序源代码。

图书在版编目(CIP)数据

面向对象程序设计：Java / 张白一，崔尚森，张辰编著. —3版.
—西安：西安电子科技大学出版社，2013.7
高等学校计算机专业“十二五”规划教材
ISBN 978-7-5606-3063-2

Ⅰ. ① 面… Ⅱ. ① 张… ② 崔… ③ 张… Ⅲ. ① JAVA语言—程序设计—高等学校—教材 Ⅳ. ① TP312

中国版本图书馆CIP数据核字(2013)第127297号

策　　划　云立实
责任编辑　买永莲　云立实
出版发行　西安电子科技大学出版社（西安市太白南路2号）
电　　话　(029)88242885　88201467　　邮　　编　710071
网　　址　www.xduph.com　　电子邮箱　xdupfxb001@163.com
经　　销　新华书店
印刷单位　陕西华沐印刷科技有限责任公司
版　　次　2013年7月第1版　2013年7月第1次印刷
开　　本　787毫米×1092毫米　1/16　印张25
字　　数　589千字
印　　数　58 001～61 000册
定　　价　43.00元
ISBN 978－7－5606－3063－2／TP
XDUP 3355003-13

前　言

承蒙读者厚爱，我们于2006年修订的《面向对象程序设计——Java》一书已经使用了多年并多次重印。在此期间，Java语言又有了长足的发展，尤其是多种可视化的编程开发工具已广为使用。为适应这种变化，我们在该书第二版的基础上又进行了一些必要的增删和修改。

首先，在本次改版中，我们启用了可视化的集成开发环境NetBeans，介绍了NetBeans IDE的安装和使用方法。

其次，我们将第二版新增的第8章“链表”更换为更实用也更能体现“复用”思想的“集合类”。这是因为我们都知道，软件的设计与数据的逻辑结构有关，而软件的程序实现依赖于数据的存储结构。数据的逻辑结构主要有线性结构、树结构、图结构和集合四大类。基本的存储结构是顺序存储和链接存储。而java.util包中的集合类就是专门解决线性结构和集合结构问题的容器类，其中最基本的容器有List(列表)、Set(集)、Queue(队列)、Map(映射)等。因此，在第8章中，我们在扼要地介绍了数据的逻辑结构、存储结构等相关基本概念后，讲述了应用java.util包中的集合类解决线性结构和集合结构问题的编程技术。这样，Java程序员在开发软件时，就不必过多地考虑相关数据结构和算法的实现细节，只需创建相应的集合对象，然后直接引用该对象提供的方法完成相应的操作，从而轻松地实现所需的数据结构和高性能、高质量的算法，不仅可以大大提高编程效率，提高程序的质量和运行速度，而且还可以实现软件的重用。

第三，对第9章至第11章的内容进行了必要的调整和增改。

第四，对书中所附全部例题程序在NetBeans环境下进行了重新调试和改写，增加了注释与图解等。

希望这些修改能为学习Java语言的读者提供更好的帮助。

本书的编写和修改过程，是笔者不断学习Java并向同行学习、向学生学习的过程。在此，对使用该书的教师、学生，以及热心向我们提出宝贵意见和建议的读者深表谢意！希望继续得到大家的支持和帮助。对于本书的各种意见和建议可直接发送E-mail到byzhang@chd.edu.cn。

编　者

2012年12月

第二版前言

Java语言推出至今已有10个年头了，在这10年中，尤其是近几年来，网络编程技术的飞速发展使得Java语言受到了广泛的欢迎并得到迅速的发展，许多网络软件开发人员将Java语言作为首选的开发工具，国内许多高校也相继开设了Java编程方面的课程。承蒙读者厚爱，我们于2002年编写并出版的《面向对象程序设计——Java》一书已多次印刷，许多读者借助该书学会了Java编程，同时也提出了一些宝贵的意见和建议。此外，随着Java版本的升级换代，书中的一些内容也需要更新。为适应这种变化，我们对该书进行了修改和补充。

与第一版相比，首先，我们在开发环境上使用了运行于Windows XP操作系统上的最新版本jdk-1_5_0_04对各个章节进行了修改，对书中所附算法和示例程序全部进行了重新调试和改写，增加了注释与图解等。其次，根据Java 2D API的特性，对与GUI设计有关的章节进行了彻底的更新。第三，为了使读者对面向对象技术有一个更深入的理解，我们在本书中新增了“链表”一章，介绍了链接存储结构的概念和特点以及在没有指针类型的Java语言中进行链表操作的技术。

全书共分16章。第1章介绍Java的特点和运行环境。第2章和第3章讲述程序设计的基本语法规则和程序流程控制。第4章和第5章全面讲述面向对象的理论和程序设计技术，包括类与对象、抽象与封装、消息、继承、多态等诸多概念及其在程序设计中的具体应用。第6章和第8章介绍数组和链表这两种常用的数据结构在Java中的应用技术。第7章介绍字符串类。从第9章开始的以后各章介绍Java的常用标准类库及其编程技巧。其中，第9～11章以最新的Java 2D API和Swing GUI组件为主讲述Java图形用户界面的设计与编程实现技术；第12章介绍Java的异常处理；第13章讲述多线程技术；第14章讲述程序的输入与输出技术；第15章和第16章分别介绍Java的网络编程技术和数据库应用编程技术。

本书在第一版的基础上，由张白一完成了原第1～7章的改写和完善工作，由崔尚森完成了原第8～15章(本版为第9～16章)的改写和完善工作，新增的第8章由张白一编写。

虽然这已是该书的第二版，但因为Java开发环境的不断发展及作者水平有限，书中仍难免有疏漏之处，我们热诚地欢迎各位同行和广大读者批评指正。对于本书的各种意见和建议可直接发送E-mail到byzhang@chd.edu.cn。

作　者

2005年10月

第一版前言

面向对象技术引起了程序设计方法学的一场革命，它已经替代面向过程的程序设计技术，成为当今计算机应用开发领域的主流技术。其原因主要在于面向对象技术能够比较客观地模拟现实世界，能够使软件开发人员运用人类认识事物所采用的一般思维方法进行软件开发；其次是在面向对象的程序设计中将数据与操作捆绑在一起，符合现代大规模软件开发的高可靠性和易维护性等方面的要求。计算机网络的发展，要求程序设计语言具有安全性强、可移植性好、与具体的操作平台无关等特性。Java 语言正是为满足这些要求而设计与研发的，并以网络为发展方向，随着网络的发展而兴盛。

1995 年 Java 语言刚一推出，便以其纯面向对象、平台无关性、多线程、高安全性、良好的可移植性和可扩展性等特征，受到了计算机界的普遍欢迎，并得到了广泛的应用和发展。近几年来，Java 的应用已经扩展到各个应用领域，加上各种功能配件的推陈出新，使得 Java 能够满足产品开发的需求，成为网络时代最流行的程序设计语言。利用 Java 来开发软件，具有跨平台、易整合、易扩展的优点。有人预言，不久的将来全世界 90%的程序代码将用 Java 语言书写或改写。

本书是为大专院校的学生及其他对面向对象编程技术和 Java 语言感兴趣的读者编写的，意在培养读者正确使用面向对象的思维方法分析问题和解决问题的能力，让读者学会利用当今最先进的软件开发工具开发软件产品，以适应网络时代社会对人才的需求。根据作者多年来讲授“面向对象程序设计”及其相关课程的经验，本书在内容的取舍上作了精心的选择，确保有一定的深度和广度；在内容的编排上体现了由浅入深、循序渐进的学习规律；在写作风格上立足于理论与实践相结合，将复杂的面向对象理论融会于众多的实例之中，使读者学会面对一个具体的问题时，能够用面向对象的思维方法分析问题，并利用面向对象的语言编写解决实际问题的计算机程序。本书可作为大专院校相关课程的教材，也可作为对面向对象编程技术和 Java 语言感兴趣的读者的自学用书。

全书共分 15 章。第 1 章介绍 Java 的特点和运行环境。第 2 章和第 3 章讲述程序设计的基本语法规则和程序流程控制。第 4 章和第 5 章全面讲述面向对象的理论和程序设计技术，包括类与对象、抽象与封装、消息、继承、多态等诸多概念及其在程序设计中的具体应用。第 6 章则介绍数组这种在各种程序设计语言中常用的数据结构在 Java 中的应用技术。从第 7 章开始介绍 Java 的常用标准类库及其编程技巧。第 7 章介绍字符串类。第 8～10 章讲述 Java 图形用户界面的设计与编程实现技术，并以最新的 Swing GUI 组件为主。第 11 章介绍 Java 的异常处理。第 12 章讲述多线程技术。第 13 章讲述程序的输入与输出技术。第 14 章和第 15 章分别介绍 Java 的网络编程技术和数据库应用编程技术。

要想很好地掌握面向对象技术，学习用 Java 程序设计语言编写出高质量的程序，先要学习其语法规则，这是编写 Java 程序的基本功；还要学习使用类库，这是提高编程效率和质量的必由之路，甚至从一定程度上来说，是否能熟练自如地掌握尽可能多的 Java 类库，

决定了一个 Java 程序员编程能力的高低。此外，计算机学科是注重实践的学科，优秀的软件人员无不经过了大量的上机实践的磨练，因此，读者只有在学习书本内容的同时辅以相应的实际练习，才能真正掌握书中介绍的知识和技能。

本书第 1～7 章由张白一编写，第 8～15 章由崔尚森编写。在本书的编写过程中，参考了许多相关书籍和网站，得到了许多同仁和同事的支持与帮助，在此我们一并表示感谢。首先应该感谢支持和参与本课程教学的长安大学的同学们，正是他们活跃的思维和永无止境的求知欲帮助作者发现内部试用版的错误，鞭策作者不断改进与完善书稿。同时，特别感谢赵文静教授在百忙之中仔细审阅了本书全稿，并提出了许多宝贵的修改意见，使本书更趋完善。此外，还要衷心感谢陕西省计算机教育学会、西安电子科技大学出版社以及长安大学的领导和同事们，正是由于他们的大力支持，才使得本书与广大读者见面。

尽管书稿几经修改，但由于作者水平所限，书中难免有疏漏之处，我们热诚地欢迎各位同行和广大读者批评指正。

作　者

2002 年 8 月

目　录

第 1 章　Java 系统环境概述 1
1.1　编程语言的发展 1
1.1.1　机器语言 2
1.1.2　汇编语言 2
1.1.3　高级语言 2
1.1.4　面向对象的语言 3
1.1.5　面向对象语言的发展 4
1.2　网络时代的编程语言——Java 4
1.2.1　Java 的产生 5
1.2.2　Java 的特点 5
1.3　Java 的开发运行环境 11
1.3.1　下载和安装 JDK 与 NetBeans 11
1.3.2　运行 NetBeans IDE 12
1.4　Java 程序的运行步骤 12
1.4.1　JVM 的体系结构及工作原理 13
1.4.2　Java Application 程序的建立及运行 14
1.4.3　Java Applet 程序的建立及运行 18
第 2 章　Java 语言基础 22
2.1　Java 符号集 22
2.1.1　标识符及其命名 22
2.1.2　关键字 23
2.1.3　运算符 23
2.1.4　分隔符 24
2.1.5　注释 24
2.2　数据类型、常量与变量 24
2.2.1　数据类型的概念 24
2.2.2　常量 25
2.2.3　变量 27
2.2.4　引用类型 32
2.3　表达式和语句 32
2.3.1　算术表达式 32
2.3.2　赋值表达式 36
2.3.3　表达式语句 37
2.3.4　关系表达式 37
2.3.5　逻辑表达式 38
2.3.6　位运算 39
2.3.7　运算符的优先级 40
第 3 章　程序流程控制 43
3.1　选择结构程序设计 43
3.1.1　if 语句 43
3.1.2　switch 语句 49
3.1.3　条件运算符 50
3.2　循环结构程序设计 51
3.2.1　while 语句 51
3.2.2　do-while 语句 52
3.2.3　for 语句 53
3.2.4　for 语句头的变化与逗号运算符 54
3.2.5　循环语句比较 56
3.2.6　循环控制要点 56
3.2.7　循环嵌套 58
.3　break 和 continue 语句 60
3.3.1　break 语句 61
3.3.2　continue 语句 63
第 4 章　类与对象 66
4.1　类与对象的概念 66
4.1.1　抽象原则 66
4.1.2　对象 69
4.1.3　类 69
4.1.4　类与对象的关系 70
4.1.5　定义类的一般格式 70
4.1.6　Java 类库 72
4.1.7　创建对象 74
4.1.8　使用对象 76

4.1.9 对象的初始化与构造方法 78
4.2 封装机制 80
4.2.1 封装的概念 80
4.2.2 类的严谨定义 80
4.2.3 类修饰符 81
4.3 数据成员 84
4.3.1 数据成员的声明 84
4.3.2 用 static 修饰的静态数据成员 84
4.3.3 静态数据成员的初始化 85
4.3.4 用 final 修饰的最终数据成员 86
4.4 成员方法 87
4.4.1 成员方法的分类 87
4.4.2 声明成员方法的格式 88
4.4.3 方法体中的局部变量 88
4.4.4 成员方法的返回值 89
4.4.5 形式参数与实际参数 90
4.4.6 成员方法的引用方式 92
4.4.7 引用成员方法时应注意的事项 92
4.4.8 成员方法的递归引用 93
4.4.9 用 static 修饰的静态方法 95
4.4.10 数学函数类方法 96
4.4.11 用 final 修饰的最终方法 97
4.4.12 用 native 修饰的本地方法 98
第 5 章 消息、继承与多态 100
5.1 消息 100
5.1.1 消息的概念 100
5.1.2 公有消息和私有消息 101
5.1.3 特定于对象的消息 101
5.2 访问控制 103
5.2.1 公共访问控制符 public 103
5.2.2 缺省访问控制符 105
5.2.3 私有访问控制符 private 106
5.2.4 保护访问控制符 protected 107
5.3 多态机制 109
5.3.1 多态的概念 109
5.3.2 重载 109
5.3.3 覆盖 110
5.4 继承机制 110
5.4.1 继承的概念 110
5.4.2 继承的特征 111
5.4.3 Java 用 extends 指明继承关系 112
5.4.4 this 与 super 114
5.4.5 构造方法的重载与继承 118
5.4.6 向方法传递对象 121
5.4.7 继承与封装的关系 122
5.5 抽象类、接口与包 122
5.5.1 抽象类 122
5.5.2 接口 126
5.5.3 包与程序复用 132
第 6 章 数组 140
6.1 一维数组 140
6.1.1 一维数组的声明 140
6.1.2 创建一维数组对象 140
6.1.3 一维数组的引用 142
6.2 一维数组引用举例 142
6.2.1 测定数组的长度 142
6.2.2 数组下标的灵活使用 143
6.2.3 数组名之间的赋值 146
6.2.4 向成员方法传递数组元素 147
6.2.5 向成员方法传递数组名 148
6.2.6 数组元素排序 150
6.2.7 对象数组 154
6.3 二维数组 156
6.3.1 二维数组的声明 157
6.3.2 创建二维数组对象 157
6.4 二维数组的引用 159
6.4.1 测定数组的长度及数组赋值 159
6.4.2 数组名作为成员方法的参数 161
第 7 章 字符串类 .....
7.1 String 类 164
7.1.1 直接赋值创建 String 对象 164
7.1.2 String 类的构造方法 164
7.1.3 String 类的常用方法 166
7.1.4 访问字符串对象 167
7.1.5 字符串比较 169
7.1.6 字符串操作 170
7.1.7 其他类型的数据转换成字符串 171
7.1.8 main 方法中的参数 172

7.2 StringBuffer 类 175
7.2.1 创建 StringBuffer 对象 175
7.2.2 StringBuffer 类的常用方法 175
7.2.3 StringBuffer 类的测试缓冲区长度的方法 176
7.2.4 StringBuffer 类的 append( )方法 177
7.2.5 StringBuffer 类的 insert( )方法 177
7.2.6 StringBuffer 类的 setcharAt( )方法 178
第 8 章 集合类 181
8.1 线性结构简介 181
8.2 集合与集合框架 182
8.2.1 集合(Collection) 182
8.2.2 集合框架 183
8.3 Collection 接口 184
8.3.1 Collection 接口简介 184
8.3.2 Iterator 迭代器 185
8.4 List 接口与实现类 187
8.4.1 List 接口简介 187
8.4.2 ArrayList 类 190
8.4.3 LinkedList 类 190
8.5 Set 接口 192
8.5.1 Set 接口简介 192
8.5.2 SortedSet 接口 195
8.6 Map 接口 196
8.6.1 Map 接口简介 196
8.6.2 Map 接口的常用操作 196
8.6.3 Map.Entry 接口的常用操作 197
8.6.4 SortedMap 接口 199
8.7 Collections 算法类 200
8.7.1 为集合增加元素的 addAll( )方法 200
8.7.2 sort( )和 reverse( )方法 201
8.7.3 实现混排的 shuffle( )方法 202
8.7.4 替换集合中元素的 replaceAll( )方法 202
8.7.5 二分查找的 binarySearch( )方法 203
8.7.6 交换指定位置元素的 swap( )方法 204
第 9 章 文字与图形 GUI 设计 206
9.1 GUI 设计概述 206
9.1.1 JFC 简介 206
9.1.2 图形用户界面元素分类 207
9.1.3 Swing 的组件 209
9.1.4 控制 Applet 运行状态的基本方法 211
9.1.5 JApplet 类 212
9.1.6 Java 2D API 212
9.1.7 Graphics2D 对象 213
9.2 绘制文字 213
9.2.1 绘制文字的成员方法 214
9.2.2 Font 类 215
9.3 Color 类 217
9.3.1 Color 类的构造方法 218
9.3.2 Color 类的数据成员常量 218
9.3.3 Color 类的成员方法 219
9.3.4 应用举例 219
9.4 绘制形状图形 220
9.4.1 绘制几何图形的方法与步骤 220
9.4.2 绘制线段与矩形 220
9.4.3 绘制椭圆、圆及弧 223
9.4.4 绘制多边形 225
9.4.5 图形重叠时的色彩设置 227
9.4.6 绘制剪切文字图形 229
第 10 章 常用组件 GUI 设计 233
10.1 布局管理器 233
10.1.1 BorderLayout 234
10.1.2 FlowLayout 235
10.1.3 CardLayout 236
10.1.4 GridLayout 237
10.1.5 BoxLayout 238
10.2 窗口与面板容器 242
10.2.1 JFrame 容器 242
10.2.2 JPanel 容器 243
10.3 事件响应原理 245
10.3.1 委托事件模型 245
10.3.2 java.awt.Event 事件类的继承关系 245
10.3.3 事件与事件源的关系 246

10.3.4 Swing 中的事件与事件监听器.....247
10.4 JLable 组件.....248
10.5 JButton 组件与 JToggleButton 组件.....249
10.5.1 常用组件的继承关系.....249
10.5.2 AbstractButton 类的常用成员方法.....250
10.5.3 JButton 类的构造方法.....251
10.5.4 JToggleButton 类的构造方法.....251
10.5.5 ActionEvent 事件及其响应.....252
10.5.6 应用举例.....253
10.6 JCheckBox 和 JRadioButton 组件.....255
10.6.1 JCheckBox 类的构造方法.....256
10.6.2 JRadioButton 类的构造方法.....256
10.6.3 ItemEvent 事件及其响应.....256
10.6.4 应用举例.....258
10.7 JComboBox 组件.....260
10.7.1 JComboBox 类的构造方法和成员方法.....260
10.7.2 事件响应.....260
10.7.3 应用举例.....261
10.8 JList 组件.....262
10.8.1 JList 类的构造方法和成员方法.....262
10.8.2 ListSelectionEvent 事件及其响应.....263
10.8.3 应用举例.....263
10.9 JTextField 与 JTextArea 组件.....264
10.9.1 JTextField 组件的构造方法和成员方法.....264
10.9.2 JTextArea 组件的构造方法和成员方法.....265
10.9.3 事件响应及应用举例.....266
习题 10.....267
第 11 章 高级组件 GUI 设计.....269
11.1 事件适配器.....269
11.2 KeyEvent 事件及其响应.....270
11.2.1 KeyEvent 事件.....270
11.2.2 获取事件源的方法.....270
11.2.3 键盘事件的响应.....270
11.2.4 应用举例.....271
11.3 MouseEvent 事件及其响应.....272
11.3.1 MouseEvent 事件.....272
11.3.2 获取事件源的方法.....273
11.3.3 鼠标事件的响应.....273
11.3.4 应用举例.....274
11.4 WindowEvent 事件及其响应.....275
11.4.1 WindowEvent 事件.....276
11.4.2 获取事件源的方法.....276
11.4.3 窗口事件的响应.....276
11.4.4 应用举例.....277
11.5 JScrollPane 与 JScrollBar 组件.....279
11.5.1 JScrollPane 组件.....279
11.5.2 JScrollBar 组件.....280
11.5.3 AdjustmentEvent 事件应用举例.....281
11.6 JTabbedPane 容器.....283
11.6.1 JTabbedPane 容器简介.....283
11.6.2 应用举例.....284
11.7 菜单设计.....285
11.7.1 菜单栏.....286
11.7.2 菜单.....286
11.7.3 菜单项.....286
11.7.4 制作菜单的一般步骤.....287
11.7.5 菜单设计应用举例.....287
11.8 对话框设计.....288
11.8.1 JOptionPane 概述.....289
11.8.2 Message Dialog.....289
11.8.3 Confirm Dialog.....290
11.8.4 Input Dialog.....291
11.8.5 Option Dialog.....291
11.8.6 JOptionPane 应用举例.....292
11.8.7 JDialog 对话框.....296
11.8.8 JDialog 应用举例.....297
第 12 章 异常处理.....300
12.1 Java 的异常处理机制.....300
12.1.1 异常处理机制的结构.....301
12.1.2 异常类的继承关系.....302
12.2 Java 的异常处理语句.....304
12.2.1 try-catch-finally 语句.....304
12.2.2 嵌套 try-catch-finally 语句.....306

12.2.3 抛出异常的 throw 语句与 throws 语句 307
第 13 章 多线程 310
13.1 Java 中的多线程实现技术 310
13.1.1 线程的生命周期 310
13.1.2 Thread 类的方法 312
13.1.3 通过继承 Thread 类方式创建线程 314
13.1.4 通过实现 Runnable 接口方式创建线程 315
13.2 多线程的管理 319
13.2.1 线程调度 319
13.2.2 线程优先级 320
13.2.3 线程同步 321
13.2.4 线程组 323
第 14 章 输入与输出 325
14.1 基本输入/输出流类 325
14.1.1 InputStream 类 325
14.1.2 OutputStream 类 329
14.1.3 Reader 类和 Writer 类 333
14.2 文件的输入/输出 336
14.2.1 File 类 337
14.2.2 FileInputStream 类和 FileOutputStream 类 340
14.2.3 字节文件输入/输出流的读/写 341
14.2.4 FileReader 类和 FileWriter 类 344
14.2.5 RandomAccessFile 类 347
第 15 章 网络编程 352
15.1 URL 通信 352
15.1.1 URL 类 353
15.1.2 利用 URL 类访问网上资源示例程序 354
15.1.3 使用 URLConnection 类访问网上资源 356
15.2 Socket 通信 357
15.2.1 Socket 的概念及通信机制 358
15.2.2 Socket 类与 ServerSocket 类 359
15.2.3 流式 Socket 通信的示例程序 360
15.2.4 URL 通信与 Socket 通信的区别 364
15.3 UDP 通信 364
15.3.1 UDP 通信机制 364
15.3.2 DatagramSocket 类 365
15.3.3 DatagramPacket 类 366
15.3.4 UDP 通信示例程序 366
第 16 章 JDBC 连接数据库 370
16.1 关系型数据库与 SQL 370
16.1.1 关系型数据库的基本概念 370
16.1.2 数据定义语言 371
16.1.3 数据操纵语言 372
16.1.4 数据查询语言 372
16.2 使用 JDBC 连接数据库 373
16.2.1 JDBC 结构 373
16.2.2 四类 JDBC 驱动程序 374
16.2.3 JDBC 编程要点 375
16.2.4 常用的 JDBC 类与方法 375
16.2.5 安装 ODBC 驱动程序示例 378
16.3 JDBC 编程实例 381
16.3.1 创建数据表 381
16.3.2 向数据表中插入数据 382
16.3.3 更新数据 383
16.3.4 删除记录 385

第1章 Java 系统环境概述

Java 语言是美国加州 Sun Microsystem 公司于 1995 年正式推出的纯面向对象(Object-Oriented，OO)的程序设计语言。由于它很好地解决了网络编程语言中的诸多问题，因此一经推出，便受到了计算机界的普遍欢迎和接受，并得到了广泛的应用和发展，成为目前网络时代最为流行的程序设计语言。

面向对象的编程语言使程序能够比较直观地反映客观世界的本来面目，并且使软件开发人员能够运用人类认识事物所采用的一般思维方法进行软件开发，是当今计算机领域中软件开发和应用的主流技术。所有面向对象的程序设计语言都支持对象、类、消息、封装、继承、多态等诸多概念，而这些概念是人们在进行软件开发、程序设计的过程中逐渐提出来的。因此，要弄清面向对象及其相关概念，就得先从程序设计语言的发展谈起。

1.1 编程语言的发展

自从 1946 年第一台电子计算机问世以来，人们一直在探索着自然语言与计算机语言之间的映射问题。我们知道，人类的任何思维活动都是借助于人们所熟悉的某种自然语言进行的。若希望借助计算机完成人类的一种思维活动，就需要把用自然语言表达的东西转换成计算机能够理解和执行的语言形式，这便是编程语言或程序设计语言。毫无疑问，电子计算机毕竟是一种机器，它能够理解和执行的编程语言和自然语言之间存在着较大的差距，这种差距被人们称做“语言的鸿沟”。这一鸿沟虽不可彻底消除，但可以使其逐渐变窄。事实上，从计算机问世至今，各种编程语言的发展变迁，其目的就是为了缩小这一鸿沟。图 1.1 引自参考文献[3]，笔者稍作修改，该图展示了从机器语言发展到面向对象的语言使“语言的鸿沟”变窄的情形。

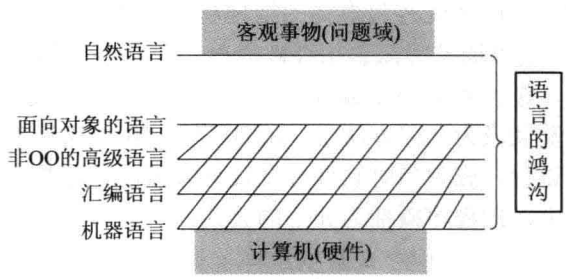

图 1.1 编程语言的发展与“语言的鸿沟”的变化

1.1.1 机器语言

电子计算机是一种机器，这种机器主要由电子元器件构成。对于电子元器件来说，最容易表达的是电流的通/断或电位的高/低两种状态。因此，在电子计算机问世之初，人们首先想到的是用“0”和“1”两种符号来代表电路的通和断两种状态，这便是最早的编程语言——机器语言。

机器语言是计算机能够理解并直接执行的唯一语言，整个语言只包含“0”和“1”两种符号。用机器语言编写的程序，无论是它的指令、数据还是其存储地址，都是由二进制的“0”和“1”组成的。这种语言离计算机最近，机器能够直接执行它。然而，由“0”和“1”组成的二进制串没有丝毫的形象意义，因此，它离人类的思维最远，“语言的鸿沟”最宽。所以，用机器语言编写程序的效率最低，并且在编写程序时很容易发生错误。

1.1.2 汇编语言

为了克服机器语言的缺陷，人们设想用一些易于理解和记忆的符号来代替二进制码，这便是汇编语言。由于汇编语言用符号构成程序，而这些符号表示指令、数据、寄存器、地址等物理概念，因而，使用汇编语言编程在适合人类形象思维的道路上前进了一步。但是，使用汇编语言编写程序时，编程人员依然需要考虑寄存器等大量的机器细节，即汇编语言仍然是一种与具体机器硬件有关的语言，是一种面向机器的语言，因此，人们也把它称为符号化的机器语言。

1.1.3 高级语言

由于机器语言和汇编语言都离不开具体的机器指令系统，用它们编程时要求程序员必须熟悉所用计算机的硬件特性，因而，用它们编写程序的技术复杂、效率不高，且可维护性和可移植性都很差。为了从根本上摆脱语言对机器的依附，人们经过多年的潜心研究，终于在 1956 年推出了一种与具体机器指令系统无关、表达方式接近自然语言的计算机语言——FORTRAN 语言。在 FORTRAN 语言程序中，采用了具有一定涵义的数据命名和人们容易理解的执行语句，屏蔽了机器细节，使得人们在书写和阅读程序时可以联系到程序所描述的具体事物。所以，人们把这种“与具体机器指令系统无关，表达方式接近自然语言”的计算机语言称为高级语言。高级语言的出现是编程语言发展史上的一大进步，它缩小了编程语言与自然语言之间的鸿沟。

此后，高级语言进一步向体现客观事物的结构和逻辑涵义的方向发展。结构化数据、结构化语句、数据抽象、过程抽象等概念相继被提出。以 1971 年推出的 Pascal 为典型代表的结构化程序设计语言，进一步缩小了编程语言和自然语言的距离。在此后的十几年中，结构化程序设计进一步发展成为一门方法学。在 20 世纪 70 年代到 80 年代，各种结构化程序设计语言及方法非常流行，成为当时软件开发设计领域的主流技术。

在结构化程序设计中，把程序概括为如下的公式：

$$\text{程序} = \text{数据结构} + \text{算法}$$

其中，数据结构是指利用计算机的离散逻辑来量化表达需要解决的问题，而算法则研究如

何高效、快捷地组织解决问题的具体过程。可见，以结构化程序设计为代表的高级语言是一种面向数据/过程的程序设计语言，人们把这类语言也称为面向过程的语言。

面向过程的语言可以精确地用计算机所理解的逻辑来描述和表达待解决问题的具体解决过程。然而，它把数据和过程分离为相互独立的实体，使程序中的数据和操作不能有效地组织成与问题域中的具体事物相对应的程序成分，所以它很难把一个具有多种相互关系的复杂事物表述清楚。程序员在编写算法时，必须时刻考虑所要处理问题的数据结构，如果数据结构发生了轻微的变化，那么对处理这些数据的算法也要做出相应的修改，甚至完全重写，否则这个算法就不可再用。因而，用这种程序设计方法编写的软件，其重用性较差。为了较好地解决软件的重用性问题，使数据与程序始终保持相容，人们又提出了面向对象的程序设计方法。

1.1.4　面向对象的语言

面向对象的编程语言(Object-Oriented Programming Language，OOPL)的设计出发点是为了能更直接地描述问题域中客观存在的事物(即对象)以及它们之间的关系。面向对象技术追求的是软件系统对现实世界的直接模拟，是将现实世界中的事物直接映射到软件系统的解空间。它希望用户最大程度地利用软件系统，花费少量的编程时间来解决需要解决的问题。

在面向对象的程序设计语言中，可以把程序描述为如下的公式：

程序 = 对象 + 消息

面向对象的语言对现实世界的直接模拟体现在下面几个方面：

(1) 对象(object)。只要我们仔细研究程序设计所面对的问题域——客观世界，就可以看到，客观世界是由一些具体的事物构成的，每个事物都具有自己的一组静态特征(属性)和一组动态特征(行为)。例如，一辆汽车有颜色、型号、马力、生产厂家等静态特征，又具有行驶、转弯、停车等动态特征。要把客观世界的这一事实映射到面向对象的程序设计语言中，则需把问题域中的事物抽象成对象，用一组数据描述该对象的静态特征(即属性，在 Java 中称之为数据成员)，用一组方法来刻画该对象的动态特征(即行为)。

(2) 类(class)。客观世界中的事物既具有特殊性又具有共同性。人类认识客观世界的基本方法之一就是对事物进行分类，即根据事物的共同性把事物归结为某些类。考虑一下所有的汽车和一辆汽车之间的关系就很容易理解这一点。OOPL 很自然地用类(class)来表示一组具有相同属性和方法的对象。

(3) 继承(inheritance)。在同一类事物中，每个事物既具有同类的共同性，又具有自己的特殊性。OOPL 用父类与子类的概念来描述这一事实。在父类中描述事物的共性，通过父类派生(derive)子类的机制来体现事物的个性。考虑同类事物中每个事物的特殊性时，可由这个父类派生子类，子类可以继承父类的共同性，又具有自己的特殊性。

(4) 封装(encapsulation)。客观世界中的事物是一个独立的整体，它的许多内部实现细节是外部所不关心的。例如，对于一个只管开车的驾驶员来说，他可能根本不知道他所驾驶的这辆汽车内部用了多少根螺钉或几米导线，以及它们是怎样组装的。OOPL 用封装机制把对象的属性和方法结合为一个整体，并且屏蔽了对象的内部细节。

(5) 关联(association)。客观世界中的一个事物可能与其他事物之间存在某种行为上的联系。例如，一辆行驶中的汽车遇到红色信号灯时要刹车停止，OOPL 便通过消息连接来表示对象之间的这种动态联系，也称之为关联。

(6) 组合体(composite)。拥有其他对象的对象被称为组合体。客观世界中较为复杂的事物往往是由其他一些比较简单的事物构成的。例如，一辆自行车是由车架、车轮、把手等构件构成的。OOPL 也提供了描述这种组合体的功能。

综上所述，面向对象的编程语言使程序能够比较直接地反映客观世界的本来面目，并且使软件开发人员能够运用人类认识事物所采用的一般思维方法来进行软件开发。面向对象的语言和人类认识、理解客观世界所使用的自然语言之间的差距是比较小的。当然，二者之间仍然存在着一定的差距，自然语言的丰富多样和借助人脑的联想思维才能辨别的语义，仍是目前任何一种计算机编程语言无法相比的。

1.1.5 面向对象语言的发展

面向对象的语言是在软件开发的实践中逐步提出并不断得到完善的。1967 年由挪威计算中心开发的 Simula 67 语言首先引入了类的概念和继承机制，被认为是面向对象语言的鼻祖。

20 世纪 70 年代出现的 CLU、并发 Pascal、Ada 和 Modula-2 等编程语言，对抽象数据类型理论的发展起到了重要作用。这些语言支持数据与操作的封装。

1980 年提出的 Smalltalk-80 是第一个完善的、能够实际应用的面向对象语言。它在系统的设计中强调对象概念的统一，并引入和完善了类、方法、实例等概念和术语，应用了继承机制和动态链接。它被认为是一种最纯粹的面向对象的程序设计语言。

从 20 世纪 80 年代中期到 90 年代，是面向对象语言走向繁荣的阶段。其主要表现是大批比较实用的 OOPL 的涌现，例如 C++、Objective-C、Object Pascal、COLOS(Common Lisp Object System)、Eiffel、Actor 及 Java 等。

综观所有的面向对象程序设计语言，我们可以把它们分为两大类：

(1) 纯粹的面向对象语言，如 Smalltalk、Java。在这类语言中，几乎所有的语言成分都是“对象”。这类语言强调的是开发快速原型的能力。

(2) 混合型的面向对象语言，如 C++、Object Pascal。这类语言是在传统的过程化语言中加入了各种面向对象的语言机构，它们强调的是运行效率。

1.2 网络时代的编程语言——Java

Internet 将世界各地成千上万的计算机子网连接成一个庞大的整体，而这些子网是由各种各样不同型号、不同规模、使用不同操作系统、具有不同应用软件平台的计算机组成的。这就很自然地提出了一个问题：有没有一种语言，使得程序员用这种语言编写的程序可以在不同的计算机上运行，从而减少编程工作量，提高程序的可移植性，使 Internet 能够发挥更多、更大的作用呢？Java 正是顺应了这种需求，因而得到了广泛的使用。它以其平台无关性、面向对象、多线程、半编译半解释等特点而成为网络时代的编程语言。

1.2.1　Java 的产生

1991 年初，美国加州的 Sun Microsystem 公司(以下简称 Sun 公司)成立了一个以 James Gosling 为首、名为 Green 的项目研发小组，其目标是开发一个面向家用电器市场的软件产品，用软件实现一个对家用电器进行集成控制的小型控制装置。他们首先注意到这个产品必须具有平台独立性，即让该软件在任何 CPU 上都能运行。为达到此目的，Gosling 首先从改写 C++ 语言的编译器着手。但是，他们很快便意识到这个产品还必须具有高度的简洁性和安全性，而 C++ 在这方面显然无法胜任。因此，Gosling 决定自行开发一种新的语言，并将该语言命名为 Oak(橡树)。

Oak 是 Green 项目小组开发的名为“*7”(StarSeven)产品中的一个组成部分。StarSeven 是一个集成了 Oak、GreenOS(一种操作系统)、用户接口模块和硬件模块四个部分的类似于 PDA(Personal Digital Assistant，个人数字助理)的设备。StarSeven 的第一个原型于 1992 年 8 月问世。尽管这个原型非常成功，但在竞争激烈的家用电器市场上却败给了竞争对手。失败的原因固然是多方面的，但笔者认为这与 Sun 公司的主业是计算机产品而不是家用电器产品这一因素密切相关。

“有心栽花花不开，无心插柳柳成荫。”有趣的是，在这段时间里，WWW 的发展却如日中天。1993 年 7 月，伊利诺斯大学的 NCSA 推出了一个在 Internet 上广为流行的 WWW 浏览器 Mosaic 1.0 版。然而，这时的 WWW 页面虽然内容丰富，声、图、文并茂，但它却是静态的，若想增强 WWW 的动感，需要通过一种机制来使它具有动态性。其解决方案显然是嵌入一种既安全可靠，又非常简练的语言。Oak 完全能满足这一要求，但是要将它推向市场，为人们所广泛接受，还必须采用一种合适的策略。在这种情况下，1994 年，Sun 公司的创始人之一 Bill Joy 的介入，促成 Oak 成为 Java 并得以走红。

Bill Joy 早年曾参与过 UNIX 的开发，深知网络对 UNIX 的推广所起的作用。因此，他不仅指定 Gosling 继续完善 Oak(发布时改名为 Java)，同时要求 Naughton 用 Oak 编写一个真正的应用程序——WebRunner，也就是后来被命名为 HotJava 的 WWW 浏览器。1994 年底，两人均出色地完成了各自的任务。这时，在这个产品的发布问题上，Bill Joy 力排众议，采取了“让用户免费使用来占领市场份额”的策略，促成了 Java 与 HotJava 于 1995 年初在 Internet 上的免费发布。由于 Java 确实是一种分布式、安全性高、内部包含编译器又非常小的适合网络开发环境的语言，因而一经发布，立即得到包括 Netscape 在内的各 WWW 厂商的广泛支持。工业界一致认为：“Java 是(20 世纪)80 年代以来计算机界的一件大事。”微软总裁 Bill Gates 认为：“Java 是长期以来最卓越的程序设计语言。”而今，Java 已成为最流行的网络编程语言。

Java 名称的由来：由于 Oak 这个名称与其他产品的名称雷同，因此开发小组后来为这个新语言取了一个新名称——Java(爪哇)。据说取这个名称的灵感来自于这样一个故事：研发小组的成员经常在公司附近的一家咖啡厅喝咖啡，而咖啡的原产地是 Java，于是就将这一新语言取名为 Java。

1.2.2　Java 的特点

Sun 公司在“Java 白皮书”中对 Java 的定义是：“Java: A simple, object-oriented, distributed,