

普通高等教育“计算机类专业”规划教材

Java程序设计 实用教程

高飞 陆佳炜 编著
徐俊 赵小敏

//



清华大学出版社

013046121

TP312JA

1501

普通高等教育“计算机类专业”规划教材

Java程序设计 实用教程

高飞 陆佳炜 徐俊 赵小敏 编著



清华大学出版社
北京



北航

C1652894

TP312JA
1501

013048151

内 容 简 介

本书主要介绍 Java 语言概述,Java 基础语法,类和对象,类的封装性、继承性、多态性及接口,数组、字符串和枚举,Java 常用类及接口,异常处理,流和文件,图形用户界面编程,多线程,网络编程,数据库编程,XML 及程序打包等内容,知识点新,重点突出,实例翔实。

本书既可作为高等院校计算机科学与技术、软件工程等相关专业的本科生和研究生的教学用书,又可作为软件开发人员知识培训与继续教育参考用书。

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 程序设计实用教程/高飞等编著. —北京:清华大学出版社,2013.6

(普通高等教育“计算机类专业”规划教材)

ISBN 978-7-302-31695-4

I. ①J… II. ①高… 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(2013)第 045861 号

责任编辑:白立军

封面设计:常雪影

责任校对:梁毅

责任印制:李红英

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:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:29

字 数:726千字

版 次:2013年6月第1版

印 次:2013年6月第1次印刷

印 数:1~2000

定 价:49.00元

产品编号:051704-01

Java 技术具有卓越的通用性、高效性、平台移植性和安全性,经过 20 多年的发展,目前已广泛应用于 PC、数据中心、游戏控制台、超级计算机、移动电话和互联网,同时拥有全球最大的开发者专业社群。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。可以说,Java 是互联网时代目前最强势、最具代表性的语言。

本书作者所在的浙江工业大学每年约有 1500 名学生会课堂上学习 Java 语言程序设计,Java 语言的发展以及知识点的更新迭代给教学工作提出了新的挑战,这是本书编写的初衷。在本书编写过程中,作者团队结合多年从事 Java 程序设计教学和科研项目开发的经验,尽量使本书的内容重点突出,文字浅显易懂,力求提供尽可能丰富翔实的实例和较全面的注释,注重知识点的更新,摒弃 Applet 等过时的知识,力争做到让读者知其然且知其所以然。

本书共分为 13 章。第 1 章 Java 语言概述,主要介绍程序设计语言的发展史、Java 语言的发展史,以及 Java 语言开发环境的配置等。第 2 章 Java 基础语法,主要介绍标识符和关键字、8 种基本数据类型、常量与变量、运算符、控制语句、常用输入输出语句等。第 3 章类和对象,主要介绍面向对象编程的基本概念、特征、Java 类的定义、成员变量与成员方法、构造方法、Java 对象的生成与使用、对象的内存分配机制、方法参数传递及对象清除、关键字 `this`\`static`\`final`\`import`、包。第 4 章类的封装性、继承性、多态性及接口,主要介绍 Java 类的面向对象三大特性、接口、抽象类以及特殊的类。第 5 章数组、字符串和枚举,主要介绍一维和多维数组的声明、实例化、初始化及其内存分配原理,不可变字符串 `String` 与可变字符串的概念、方法与内存分配原理及它们之间的异同,枚举的概念与应用。第 6 章 Java 常用类及接口,主要介绍 Java API 类库中位于 `java.lang` 包中的 `Object` 类、`Math` 类、`System` 类、`Runtime` 类,`java.util` 包中的 `Date` 类、`Calendar` 类、`Random` 类以及各类集合,for 循环在数组、集合中的应用。第 7 章异常处理,主要介绍异常处理的动机、异常处理的 `try-catch-finally` 模式、`throws` 和 `throw` 语句和异常处理原则。第 8 章流和文件,主要介绍字节流和字符流的概念、常用的字节流和字符流类、字节流和字符流的异同、文件类、对象序列化、Java 中的乱码问题。第 9 章图形用户界面编程,主要介绍 `java.swing` 包、容器组件、菜单与树、基本组件、组件常用方法、布局管理器、事件处理模型、鼠标事件处理、键盘事件处理以及事件适配器类。第 10 章多线程,主要介绍通过继承 `Thread` 类与实现 `Runnable` 接口两种多线程编程方法、线程互斥与同步、后台线程。第 11 章网络编程,主要介绍网络编程基础概念、URL 编程、Socket 编程、`InetAddress` 类。第 12 章数据库编程,主要介绍 JDBC 概念、JDBC API、JDBC 编程实例。第 13 章 XML 及程序打包,主要介绍 XML 的概念、XML 在

Java 程序中的应用、Java 程序的发布。

全书由高飞负责策划、组织、整理和统稿，参与编写的老师包括陆佳炜、徐俊、赵小敏等，由于时间仓促，各位教师在各章节的例程命名上不统一，还望读者见谅，在后续版本中将统一调整。书中内容虽为作者多年从事教学和科研工作的总结和体会，但由于 Java 仍在不断发展之中，新知识日新月异，作者的理论与实践水平有限，难免存在错误和不足之处，敬请读者批评指正。

此外，本书还配有专门的网站，其网址是 <http://www.zjut-java.com>，读者可通过该网站找到课件、所有例程、在线测试等相关资料，上述资料将同时在清华大学出版社网站 (<http://www.tup.com.cn>) 上提供。

高 飞
2013 年 1 月

FOREWORD

本书主要介绍 Java 语言及其在 Web 开发中的应用。全书共分 10 章，第 1 章介绍 Java 语言的发展历史、特点、运行平台及基本语法；第 2 章介绍 Java 语言的类与接口、封装、继承、多态、异常、集合、多线程、网络编程等；第 3 章介绍 Java 语言的异常处理、集合、多线程、网络编程等；第 4 章介绍 Java 语言的异常处理、集合、多线程、网络编程等；第 5 章介绍 Java 语言的异常处理、集合、多线程、网络编程等；第 6 章介绍 Java 语言的异常处理、集合、多线程、网络编程等；第 7 章介绍 Java 语言的异常处理、集合、多线程、网络编程等；第 8 章介绍 Java 语言的异常处理、集合、多线程、网络编程等；第 9 章介绍 Java 语言的异常处理、集合、多线程、网络编程等；第 10 章介绍 Java 语言的异常处理、集合、多线程、网络编程等。

第 1 章 Java 语言概述 /1

- 1.1 程序设计语言的发展史 /1
- 1.2 Java 语言的发展史 /4
- 1.3 Java 开发环境的配置 /6
- 1.4 Java 程序开发过程及常用工具介绍 /9
 - 1.4.1 Java 程序工作原理 /9
 - 1.4.2 用记事本开发 /9
 - 1.4.3 用 Eclipse 开发 /11
 - 1.4.4 用 JCreator 开发 /15

第 2 章 Java 基础语法 /18

- 2.1 标识符和关键字 /18
- 2.2 基本数据类型 /20
 - 2.2.1 布尔型 /20
 - 2.2.2 字符型 /21
 - 2.2.3 整数型 /22
 - 2.2.4 浮点型 /24
 - 2.2.5 类型转换 /25
- 2.3 常量与变量 /29
- 2.4 运算符 /31
 - 2.4.1 算术运算符 /32
 - 2.4.2 关系运算符 /34
 - 2.4.3 逻辑运算符 /35
 - 2.4.4 位运算符 /37
 - 2.4.5 赋值类运算符 /38
 - 2.4.6 条件运算符 /39
 - 2.4.7 对象运算符 /39
- 2.5 语句 /40
 - 2.5.1 分支语句 /40
 - 2.5.2 循环语句 /44
 - 2.5.3 跳转语句 /47
 - 2.5.4 注释语句 /51
- 2.6 输入参数方式 /52

第 3 章 类和对象 /57

3.1 面向对象技术基础 /57

3.1.1 面向对象基本概念 /57

3.1.2 面向对象基本特征 /59

3.2 类 /61

3.2.1 类的定义 /61

3.2.2 成员变量与成员方法 /63

3.2.3 构造方法 /64

3.2.4 main 方法 /66

3.3 对象 /67

3.3.1 对象的生成与使用 /67

3.3.2 变量的作用域 /69

3.3.3 对象的内存分配机制 /70

3.3.4 方法参数的传递 /72

3.3.5 对象的清除 /76

3.4 this 关键字 /77

3.5 static 关键字 /79

3.6 final 关键字 /84

3.7 import 和包 /86

第 4 章 类的封装性、继承性、多态性及接口 /90

4.1 封装性 /90

4.1.1 类的访问控制方式 /90

4.1.2 类成员的访问控制方式 /90

4.1.3 封装性的设计原则 /92

4.2 继承性 /92

4.2.1 extends 关键字 /92

4.2.2 super 关键字 /95

4.2.3 构造方法的继承性 /97

4.3 多态性 /102

4.3.1 方法重载 /102

4.3.2 方法覆盖 /104

4.4 抽象类 /108

4.5	接口	/111
4.5.1	接口的定义	/111
4.5.2	接口的实现	/113
4.5.3	接口的作用	/116
4.5.4	接口与抽象类的区别	/120
4.6	特殊的类	/121
4.6.1	实名内部类	/121
4.6.2	匿名内部类	/125
4.6.3	泛型类	/128
4.6.4	Class 类	/132
第 5 章 数组、字符串和枚举 /134		
5.1	数组	/134
5.1.1	一维数组	/134
5.1.2	二维数组	/137
5.1.3	数组的注意事项	/139
5.1.4	数组的应用	/139
5.2	字符串	/141
5.2.1	不可变字符串: String	/142
5.2.2	可变字符串	/149
5.2.3	String 与 StringBuffer 的异同	/151
5.3	字符串与其他数据类型的转换	/153
5.3.1	将其他数据转换成字符串	/153
5.3.2	将字符串转换成其他数据	/156
5.4	枚举	/158
5.4.1	枚举定义	/158
5.4.2	枚举变量和常量	/158
5.4.3	枚举的常见用法	/159
第 6 章 Java 常用类及接口 /163		
6.1	Java API 类库	/163
6.2	java.lang 包	/164
6.2.1	Object 类	/166

6.2.2	Math 类	/167
6.2.3	System 类	/168
6.2.4	Runtime 类	/172
6.3	java.util 包	/174
6.3.1	Date 类	/176
6.3.2	Calendar 类	/177
6.3.3	Random 类	/180
6.3.4	无序集合: Collection 接口和 Collections 类	/183
6.3.5	有序集合: List 接口和 ArrayList、LinkedList 和 Vector 类	/184
6.3.6	非重复集合: Set 接口和 HashSet、TreeSet 及 LinkedHashSet 类	/191
6.3.7	映射集合: Map 接口和 TreeMap 类	/192
6.3.8	for 循环简化写法在集合、数组中的应用	/193
第 7 章 异常处理 /198		
7.1	为什么要进行异常处理	/198
7.2	Java 中的异常类	/199
7.3	异常处理模式	/201
7.3.1	try-catch-finally 语句	/201
7.3.2	异常类成员方法	/203
7.3.3	异常捕获与处理	/203
7.4	重新抛出异常	/205
7.4.1	throws 语句	/205
7.4.2	throw 语句	/206
7.5	异常处理原则	/207
第 8 章 流和文件 /208		
8.1	流的基本概念	/208
8.2	字节流	/209

8.2.1	输入字节流	/209
8.2.2	输出字节流	/210
8.3	字符流	/212
8.3.1	输入字符流	/212
8.3.2	输出字符流	/213
8.3.3	字符缓冲流	/216
8.3.4	字节流和字符流的异同	/218
8.4	文件	/218
8.4.1	文件属性类	/218
8.4.2	随机访问文件类	/222
8.4.3	文件过滤接口	/224
8.5	对象序列化	/226
8.5.1	序列化是什么	/226
8.5.2	什么情况下需要序列化	/226
8.5.3	对象序列化时发生了什么	/227
8.5.4	实现序列化的步骤	/227
8.5.5	序列化对象的条件	/227
8.5.6	反序列化	/228
8.5.7	序列化注意事项	/229
8.6	Java 中的乱码问题	/230
8.6.1	Java 中字符的表达式	/230
8.6.2	Unicode 简介	/230
8.6.3	Unicode 编码方式	/231
8.6.4	Unicode 实现方式	/232
8.6.5	字节序	/237
8.6.6	其他编码方式	/238
8.6.7	Java 中的 Unicode	/241
8.6.8	如何处理中文乱码问题	/248
第 9 章 图形用户界面编程 /252		
9.1	AWT 与 Swing	/252
9.1.1	AWT	/252
9.1.2	Swing	/253

9.2	容器组件	/255
9.2.1	JFrame	/255
9.2.2	JPanel	/261
9.2.3	JScrollPane	/262
9.2.4	JSplitPane	/263
9.3	菜单和工具条	/264
9.3.1	菜单组件	/264
9.3.2	工具栏组件	/267
9.4	基本组件	/269
9.4.1	标签	/269
9.4.2	单行文本框	/271
9.4.3	按钮	/272
9.4.4	下拉框	/274
9.4.5	列表框	/276
9.4.6	多行文本框	/278
9.4.7	表格组件	/279
9.4.8	树形组件	/283
9.4.9	进度条组件	/291
9.5	组件常用方法	/294
9.5.1	颜色	/294
9.5.2	透明性	/295
9.5.3	边框	/295
9.5.4	字体	/296
9.5.5	大小与位置	/299
9.5.6	激活与可见性	/300
9.6	布局管理器	/300
9.6.1	流式布局	/301
9.6.2	边界布局	/302
9.6.3	盒式布局	/303
9.6.4	网格布局	/304
9.6.5	卡片布局	/305
9.6.6	网格包布局	/307

- 9.6.7 布局基本原则及复杂布局举例 /313
- 9.6.8 界面风格的选择 /315
- 9.7 事件处理模型 /317
 - 9.7.1 事件处理机制 /317
 - 9.7.2 事件对象 /317
 - 9.7.3 监听器接口 /319
 - 9.7.4 编写事件处理程序 /320
- 9.8 鼠标事件处理 /324
- 9.9 事件适配器类 /326
- 9.10 键盘事件处理 /328

第 10 章 多线程 /331

- 10.1 线程简介 /331
 - 10.1.1 进程与线程 /331
 - 10.1.2 线程生命周期 /331
- 10.2 编写线程程序 /333
 - 10.2.1 第一种方法：继承 Thread 类 /333
 - 10.2.2 第二种方法：实现 Runnable 接口 /334
 - 10.2.3 两种方法比较 /336
 - 10.2.4 线程基本控制方法 /337
- 10.3 线程互斥与同步 /342
 - 10.3.1 多线程同步的基本原理 /344
 - 10.3.2 多线程同步实例 /344
- 10.4 后台线程 /347

第 11 章 网络编程 /349

- 11.1 网络编程基础 /349
 - 11.1.1 网络编程的两个基本问题 /349
 - 11.1.2 网络编程相关的基本概念 /349
- 11.2 URL 编程 /350
 - 11.2.1 URL 简介 /350
 - 11.2.2 URL 类 /351

11.2.3	从 URL 读取万维网资源	/352
11.2.4	网络编程的乱码问题	/354
11.2.5	利用 URLConnection 实现双向通信	/355
11.3	Socket 编程	/357
11.3.1	Socket 编程的过程	/357
11.3.2	利用 Socket 实现断点续传	/359
11.3.3	利用 Socket 实现聊天程序	/369
11.4	IntelAddress 类	/374
11.4.1	获取本机的计算机名与 IP 地址	/375
11.4.2	获取 Internet 上主机的 IP 地址	/375
第 12 章 数据库编程 /378		
12.1	JDBC 概述	/378
12.1.1	JDBC 模型	/378
12.1.2	JDBC 驱动方式	/379
12.2	JDBC API	/381
12.3	JDBC 编程实例	/383
12.3.1	JDBC 驱动程序设置	/383
12.3.2	建立数据库连接	/384
12.3.3	添加记录	/393
12.3.4	查询记录	/397
12.3.5	删除记录	/401
12.3.6	修改记录	/404
12.3.7	数据库操作综合实例	/408
12.3.8	SQL 数据库常用命令	/419
第 13 章 XML 及程序打包 /422		
13.1	XML 简介	/422
13.2	XML 在 Java 程序中的应用	/426
13.2.1	DOM 编程	/426
13.2.2	加载 XML 文件	/430

13.2.3	访问 XML 元素和属性	/432
13.2.4	利用 XML 文件存储信息	/434
13.3	Java 程序的发布	/440
13.3.1	利用 cmd 工具打包	/441
13.3.2	利用 Eclipse 打包	/445
	参考文献	/449

第1章 Java语言概述

1.1 程序设计语言的发展史

计算机语言的发展是一个不断演化的过程,其根本的推动力就是对抽象机制更高的要求,以及对更好的支持程序设计思想的追求。具体地说,就是把机器能够理解的语言提升到能够很好地模仿人类思考问题的高度。计算机语言的演化过程是从最开始的机器语言到汇编语言,再到各种结构化高级语言,最后到支持面向对象技术的面向对象语言。

1. 机器语言

电子计算机所使用的是由0和1组成的二进制数,二进制是计算机语言的基础。计算机发明之初,人们通过写出一串串由0和1组成的指令序列交由计算机执行,这就是机器语言。使用机器语言是十分痛苦的,特别是在程序有错需要修改时更是如此。而且,由于每台计算机的指令系统往往各不相同,所以,在一台计算机上执行的程序,要想在另一台计算机上执行,必须另编程序,造成了重复工作。但由于使用的是针对特定型号计算机的语言,故而运算效率是所有语言中最高的。机器语言是第一代计算机语言。

2. 汇编语言

为了减轻使用机器语言编程的痛苦,人们进行了一种有益的改进:用一些简洁的英文字母、符号串来替代一个特定的指令的二进制串,例如,用ADD代表加法,MOV代表数据传递等。这样,人们很容易读懂并理解程序在干什么,纠错及维护都变得方便了。这种程序设计语言就称为汇编语言,即第二代计算机语言。然而计算机是不认识这些符号的,所以这就需要有一个专门的程序,专门负责将这些符号翻译成二进制数的机器语言,这种翻译程序被称为汇编程序。

汇编语言同样十分依赖于机器硬件,移植性不好,但效率仍十分高。针对计算机特定硬件而编制的汇编语言程序,能准确发挥计算机硬件的功能和特长,程序精练而质量高,所以汇编语言至今仍是一种常用而强有力的软件开发工具。

3. 高级语言

从最初与计算机交流的痛苦经历中,人们意识到,应该设计这样一种语言,既接近于数学语言或人的自然语言,同时又不依赖于计算机硬件,即用该语言编出的程序能在所有机器上通用。经过努力,1954年,第一个完全脱离机器硬件的高级语言——FORTRAN问世了。50多年来,共有几百种高级语言出现。

高级语言的发展也经历了从早期语言到结构化程序设计语言,从面向过程语言到非过程化程序语言的过程。相应地,软件的开发也由最初的个体手工作坊式的封闭式生产,发展为产业化、流水线式的工业化生产。

20世纪60年代中后期,软件越来越多,规模越来越大,而软件的生产基本上是人自为战,缺乏科学规范的系统规划与测试、评估标准,其恶果是大批耗费巨资建立起来的软件系统,由于含有错误而无法使用,造成巨大损失。软件给人的感觉是越来越不可靠,几乎没有

不出错的软件。这一切,极大地震动了计算机界,史称“软件危机”。人们认识到:大型程序的编制不同于小程序的编写,它是一项新的技术,我们应该像处理工程一样处理软件研制的全过程。程序的设计应易于保证正确性,也便于验证正确性。1969年,计算机学家们提出了结构化程序设计方法。1970年,第一个结构化程序设计语言——Pascal 语言出现,标志着结构化程序设计时期的开始。

20 世纪 80 年代初,人们对于软件设计思想又发起了一次革命,最终成果就是提出了面向对象的程序设计。在此之前的高级语言,几乎都是面向过程的,程序的执行是流水线式的,在一个模块被执行完成前,人们不能干别的事,也无法动态地改变程序的执行方向。这和人们日常处理事务的方式是不一致的。对人而言是希望发生一件事就处理一件事,也就是说,不能仅仅面向过程,而应是面向具体的应用功能,也就是对象(Object)。面向对象的程序设计方法就是软件的集成化,如同硬件的集成电路一样,生产一些通用的、封装紧密的功能模块,称之为软件集成块,它不能单独实现具体应用,但能通过相互组合来完成具体的应用功能,同时,软件集成块还可以重复使用。使用者只需要关心它的接口(输入量、输出量)及能实现的功能,至于如何实现的,那是它内部的事,使用者完全不用关心。C++、Java、C# 就是面向对象编程语言的典型代表。

4. 几种常见的高级语言

1) C 语言

C 语言是由 UNIX 的研制者丹尼斯·里奇(Dennis Ritchie)于 1970 年在由肯·汤普逊(Ken Thompson)所研制出的 B 语言的基础上发展和完善起来的。目前,C 语言编译器普遍存在于各种不同的操作系统中,例如,UNIX、MS-DOS、Microsoft Windows 及 Linux 等。C 语言的设计影响了许多后来的编程语言,例如,C++、Objective-C、Java、C# 等。

C 语言是一种通用的、过程式的编程语言,具有高效、灵活、功能丰富、表达力强和可移植性较高等特点,在程序员中备受青睐,被广泛用于系统与应用软件的开发。C 语言的设计目标是提供一种能以简易的方式编译、处理低级内存、产生少量的机械码以及不需要任何执行环境支持便能执行的编程语言。C 语言也很适合搭配汇编语言来使用。尽管 C 语言提供了许多低级处理的功能,但仍然保持着良好跨平台的特性,以一个标准规格写出的 C 语言程序可在许多计算机平台(甚至包含一些嵌入式处理器以及超级计算机等作业平台)上进行编译。

2) C++

贝尔实验室的比雅尼·斯特劳斯特鲁普博士在 20 世纪 80 年代发明并实现了 C++。起初,这种语言被称为 C with Classes(包含类的 C 语言),作为 C 语言的增强版出现。随后,C++ 不断增加新特性。虚函数(Virtual Function)、操作符重载(Operator Overloading)、多重继承(Multiple Inheritance)、模板(Template)、异常处理(Exception)、RTTI(Runtime Type Information)、命名空间(Namespace)逐渐纳入标准。1998 年国际标准化组织(ISO)颁布了 C++ 程序设计语言的国际标准 ISO/IEC 14882—1998。

C++ 是一种使用非常广泛的计算机程序设计语言。它是一种静态数据类型检查的,支持多范型的通用程序设计语言。C++ 支持程序化程序设计、数据抽象化、面向对象程序设计、泛型程序设计、基于原则设计等多种程序设计风格。

C++ 语言发展大概可以分为三个阶段:第一阶段从 20 世纪 80 年代到 1995 年。这一

阶段 C++ 语言基本上传统意义上的面向对象语言,并且凭借着接近 C 语言的效率,在工业界使用的开发语言中占据了相当大份额。第二阶段从 1995 年到 2000 年。这一阶段由于标准模板库(STL)和后来的 Boost 等程序库的出现,泛型程序设计在 C++ 中占据了越来越大的比重。当然,同时由于 Java、C# 等语言的出现和硬件价格的大幅度下降,C++ 受到了一定的冲击。第三阶段从 2000 年至今。由于以 Loki、MPL 等程序库为代表的产生式编程和模板元编程的出现,C++ 出现了发展历史上又一个新的高峰。这些新技术的出现以及和原有技术的融合,使 C++ 成为当今主流程序设计语言中最复杂的一员。

3) Visual Basic(VB)

VB 是由微软公司开发的包含协助开发环境的事件驱动编程语言。它源自于 BASIC 编程语言。1991 年 4 月,Visual Basic 1.0 for Windows 版本发布,它可是第一个“可视”的编程软件。这使得程序员欣喜之极,都尝试在 VB 的平台上进行软件创作。VB 拥有图形用户界面(GUI)和快速应用程序开发(RAD)系统,可以轻易地使用 DAO、RDO、ADO 连接数据库,也可以轻松地创建 ActiveX 控件。程序员可以轻松地使用 VB 提供的组件快速建立一个应用程序。

VB 的中心思想就是要便于程序员使用,无论是新手或者专家。VB 使用了可以简单建立应用程序的 GUI 系统,但是又可以开发相当复杂的程序。VB 的程序是一种基于窗体的可视化组件安排的联合,并且可通过增加代码来指定组件的属性和方法。因为默认的属性和方法已经有一部分定义在了组件内,所以程序员不用写多少代码就可以完成一个简单的程序。

4) Delphi

Delphi 最早由美国 Borland(宝兰)公司于 1995 年开发。Delphi 是一个集成开发环境(IDE),使用的核心是由传统 Pascal 语言发展而来的 Object Pascal,以图形用户界面(Graphical User Interface,GUI)为开发环境,通过 IDE、VCL 工具与编译器,配合连接数据库的功能,构成一个以面向对象程序设计为中心的应用程序开发工具。Delphi 所编译的可运行档,虽然容量较大,但因为产生的是真正的原生物理码,效能上比较高效。除了使用数据库的程序之外,Delphi 程序不需安装即可运行,在使用上相当方便。

Delphi 在本质上应该归类为软件开发工具,而非程序语言,但 Delphi 几乎是目前市场上唯一使用 Pascal,并持续推出新版本的商业产品,因此,有时人们会把 Delphi 视为 Object Pascal 的代名词。Borland 公司因而把 Object Pascal 改称为 Delphi。

5) Java

Java 拥有跨平台、面向对象、泛型编程的特性,于 1995 年 5 月由 SUN 公司正式对外发布。Java 伴随着互联网的迅猛发展而发展,逐渐成为重要的网络编程语言。

Java 编程语言的风格十分接近 C++ 语言。Java 继承了 C++ 语言面向对象技术的核心,舍弃了 C++ 语言中容易引起错误的指针,改以引用取代,同时移除多重继承特性,改用接口取代,另外,增加垃圾回收器功能。在 Java SE 1.5 版本中引入了泛型编程、类型安全的枚举、不定长参数和自动装/拆箱特性。SUN 公司对 Java 语言的解释是:“Java 编程语言是一个简单、面向对象、分布式、解释性、健壮、安全与系统无关、可移植、高性能、多线程和动态的语言。”

Java 不同于一般的编译语言和直译语言。它首先将源代码编译成字节码(Bytecode),