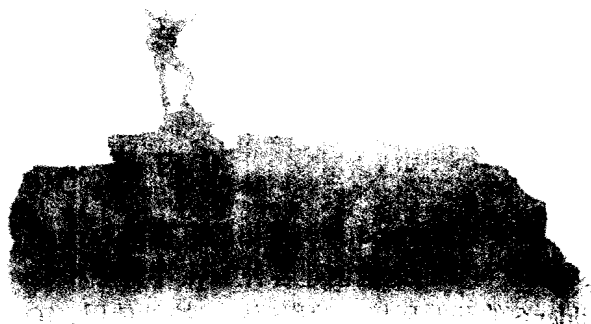


123

计算机技术入门与提高系列

TP312JA
x76



JBUILDER 6

入门与提高

r u m e n y u t i g a o

◆ 求是科技 徐雅静 编著

人民邮电出版社

图书在版编目 (CIP) 数据

JBuilder 6 入门与提高/求是科技, 徐雅静编著. —北京: 人民邮电出版社, 2002.6

ISBN 7-115-10283-X

I. J... II. ①求... ②徐... III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 029039 号

内 容 提 要

JBuilder 6 是 Borland 公司推出的可视化 Java 语言编程工具, 它提供了一个功能强大的集成开发环境, 可以迅速地创建各种 Java 小程序和 Java 应用程序。本书共分 11 章, 分别介绍了 JBuilder 6 的开发环境、编程语法、控件使用、IO 流的使用、多线程的应用以及 Java 在多媒体、网络和数据库中的应用, 并配以大量实用性强、易学易用的实例, 帮助读者循序渐进地从 Java 的基础着手, 逐渐深入, 达到轻松掌握 Java 的目的。

本书可作为学习 Java 编程的入门教材和教学参考书, 亦可供软件开发人员参考。

计算机技术入门与提高系列

JBuilder 6 入门与提高

-
- ◆ 编 著 求是科技 徐雅静
责任编辑 张立科
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67180876
北京汉魂图文设计有限公司制作
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
 - ◆ 开本: 787×1092 1/16
印张: 20.75
字数: 499 千字 2002 年 6 月第 1 版
印数: 1-5 000 册 2002 年 6 月北京第 1 次印刷

ISBN 7-115-10283-X/TP · 2860

定价: 28.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

目 录

第 1 章 概述	1
1.1 初识 Java	1
1.1.1 Java 的产生	1
1.1.2 什么是 Java	2
1.2 Java 的特性	2
1.2.1 Java 的特点	3
1.2.2 丰富的类库	4
1.2.3 Java 和 C、C++	4
1.2.4 应用前景	5
1.3 Java 体系结构	6
1.3.1 Java 虚拟机	6
1.3.2 Java 体系	6
1.4 本章小结	7
第 2 章 JBuilder 6 编程环境	8
2.1 JBuilder 6 开发环境概述	8
2.2 JBuilder 6 的安装	9
2.2.1 安装环境	9
2.2.2 安装步骤	9
2.3 JBuilder 6 的使用	13
2.3.1 主窗口	13
2.3.2 主菜单	16
2.3.3 工具栏	31
2.4 编写第一个 Java 程序	32
2.4.1 创建 HelloWorld 程序	32
2.4.2 代码分析	36
2.5 本章小结	39
第 3 章 Java 编程基础	41
3.1 类	41
3.1.1 类声明	41
3.1.2 类的实体	42
3.2 修饰符	48
3.2.1 类修饰符	48
3.2.2 成员变量修饰符	48

3.2.3	类方法修饰符	49
3.3	对象	50
3.3.1	对象的创建	50
3.3.2	对象的撤销	51
3.4	接口	52
3.4.1	接口的定义	52
3.4.2	接口的实现	53
3.4.3	接口继承	54
3.5	包	54
3.5.1	包的定义	55
3.5.2	包的引入	55
3.5.3	Java 常用包	56
3.6	Java 特殊变量	58
3.6.1	null	58
3.6.2	this	59
3.6.3	super	59
3.7	本章小结	60
第 4 章	Java 编程语法	61
4.1	数据类型	61
4.1.1	注释和标识符	61
4.1.2	基本数据类型	62
4.1.3	常量	63
4.1.4	变量	64
4.2	表达式和运算符	66
4.2.1	算术运算符	66
4.2.2	逻辑运算符	67
4.2.3	关系运算符	68
4.2.4	位运算符	68
4.2.5	赋值运算符	69
4.2.6	条件运算符	69
4.2.7	运算符的优先级	69
4.3	字符串和数组	70
4.3.1	String 字符串	70
4.3.2	StringBuffer 字符串	73
4.3.3	数组	74
4.4	类型转换	77
4.4.1	自动类型转换	77
4.4.2	强制类型转换	77
4.5	流程控制	79

4.5.1	分支流程	79
4.5.2	循环流程	81
4.5.3	跳转流程	83
4.6	异常处理	85
4.6.1	抛出异常	86
4.6.2	捕获异常	87
4.6.3	用 try-catch 来处理异常	87
4.7	事件处理	90
4.7.1	鼠标事件	90
4.7.2	键盘事件	94
4.7.3	其他事件	96
4.8	编程规范	100
4.8.1	程序结构	101
4.8.2	命名规范	101
4.8.3	编码格式	101
4.9	本章小结	102
第 5 章	Java 图形用户界面	103
5.1	Java 小程序 (Applet) 编程	103
5.1.1	Applet 概述	103
5.1.2	使用 Applet 类	104
5.2	Java 基本控件	107
5.2.1	按钮 (Button) 控件	110
5.2.2	标签 (Label) 控件	112
5.2.3	编辑 (TextField 和 TextArea) 控件	112
5.2.4	复选框 (Checkbox) 控件	113
5.2.5	单选钮 (CheckboxGroup) 控件	117
5.2.6	下拉框 (Choice) 控件	120
5.2.7	列表框 (List) 控件	122
5.2.8	滚动条 (Scrollbar) 控件	125
5.2.9	面板 (Panel) 控件	129
5.3	布局管理器	134
5.3.1	BorderLayout 布局管理器	135
5.3.2	FlowLayout 布局管理器	137
5.3.3	VerticalFlowLayout 布局管理器	138
5.3.4	GridLayout 布局管理器	139
5.3.5	GridBagLayout 布局管理器	140
5.3.6	PaneLayout 布局管理器	143
5.3.7	XYLayout 布局管理器	144
5.3.8	CardLayout 布局管理器	146

5.4 本章小结	151
第 6 章 JavaBeans 组件	152
6.1 Java 应用程序	152
6.1.1 创建第一个应用程序	152
6.1.2 代码分析	155
6.2 JavaBeans 组件概述	158
6.3 常用 JavaBeans 组件	163
6.3.1 下拉菜单 (JMenuBar) 组件	163
6.3.2 工具栏 (JToolBar) 组件	168
6.3.3 树形控件 (JTree) 组件	171
6.3.4 快捷菜单 (JPopupMenu) 组件	178
6.3.5 表格控件 (JTable) 组件	182
6.4 本章小结	190
第 7 章 控制 Java 的输入输出	191
7.1 利用 java.io 包进行输入输出	191
7.1.1 输入类	191
7.1.2 输出类	192
7.1.3 File 类	193
7.2 文件的输入输出	193
7.2.1 读文件	193
7.2.2 写文件	200
7.3 磁盘输入输出	203
7.4 本章小结	209
第 8 章 Java 多线程	210
8.1 多线程概述	210
8.1.1 多线程定义	210
8.1.2 多线程相关概念	210
8.2 线程 (Thread) 类	212
8.2.1 Thread 类的方法	212
8.2.2 Thread 类的使用	213
8.3 Runnable 接口	216
8.3.1 Runnable 接口的方法	216
8.3.2 Runnable 接口的使用	218
8.4 线程同步	219
8.5 线程通信	222
8.5.1 使用管道类	222
8.5.2 使用标志变量	229
8.5.3 使用 wait 和 notify 方法	233

8.6 本章小结	236
第 9 章 Java 多媒体	237
9.1 图形技术	237
9.1.1 基础概念	237
9.1.2 图形 (Graphics) 类	238
9.1.3 颜色 (Color) 类及绘图示例	239
9.1.4 字体 (Font) 类及示例	243
9.2 声音技术	247
9.2.1 AudioClip 类	248
9.2.2 声音技术示例	248
9.3 图像技术	250
9.3.1 图像的装入和显示	250
9.3.2 图像装入跟踪	252
9.3.3 内存图像映射	254
9.3.4 图像过滤	256
9.4 动画技术	261
9.4.1 动画的实现	262
9.4.2 消除动画闪烁	264
9.5 本章小结	266
第 10 章 Java 网络编程	267
10.1 概述	267
10.1.1 通信协议	267
10.1.2 通信端口	268
10.1.3 IP 地址	269
10.2 IP 地址的获取	269
10.3 TCP 和套接字	270
10.3.1 Socket 通信基础	270
10.3.2 Client/Server 程序	272
10.4 UDP 和数据报	279
10.4.1 数据报通信基础	280
10.4.2 Chat 程序	281
10.5 WWW 应用	286
10.5.1 URL 类	286
10.5.2 获取 Internet 资源	287
10.6 FTP 应用	290
10.6.1 FTP 组件	290
10.6.2 制作 FTP 客户端	292
10.7 本章小结	296

第 11 章 Java 数据库编程	297
11.1 JDBC 概述	297
11.1.1 JDBC 类	297
11.1.2 JDBC API 实例	302
11.2 JDataStore 概述	303
11.2.1 JDataStore Explorer	303
11.2.2 JDataStore 数据库应用	306
11.3 数据库组件	308
11.3.1 DataExpress 组件	308
11.3.2 dbSwing	311
11.3.3 More dbSwing	312
11.3.4 数据库应用示例	312
11.4 本章小结	319

第1章 概述

1.1 初识 Java

最近几年,在 Internet 上出现的特别吸引人的事件就是 Java 语言了。短短几年时间,Java 这个诞生在信息世界的宠儿,以它所拥有的无与伦比的优越性,在全球信息网中得到广泛的应用,那么 Java 是如何产生的呢?

1.1.1 Java 的产生

1991 年, SUN 公司的 JameGosling、BillJoe 等人,为在电视、烤面包箱等家用消费类电子产品上进行交互式操作而开发了一个名为 Oak 的软件(即一种橡树的名字),但当时并没有引起人们的注意。

1994 年下半年, Internet 的迅猛发展, WWW 的快速增长,促进了 Java 语言的研制和发展,使得它逐渐成为 Internet 上最受欢迎的开发与编程语言。一些著名的计算机公司(如 Microsoft、IBM、Netscape、Novell、Apple 等)纷纷购买 Java 语言的使用权。

1995 年, Java 语言被美国的著名杂志《PC Magazine》评为 1995 年十大优秀科技产品之一(计算机类就此一项入选),随之大量出现的用 Java 语言编写的软件产品,也受到工业界的重视与好评,“Java 是 80 年代以来计算机界的一件大事”。微软公司总裁比尔·盖茨在悄悄地观察了一段时间后,不无感慨地说:“Java 是长时间以来最卓越的程序设计语言”,并确定微软公司整个软件开发的战略从 PC 单机时代向着以网络为中心的计算时代转移,而购买 Java 则是他的重大战略决策的实施部署。

因此, Java 的诞生必将对整个计算机产业产生深远的影响,对传统的计算模型提出了新的挑战。SUN 公司的总裁 ScottMcNealy 认为 Java 为 Internet 和 WWW 开辟了一个崭新的时代。WWW 的创始人 Berners-Lee 说:“计算机事业发展的下一个浪潮就是 Java,并且将很快会发生”。由此可见, Java 已成为信息时代的网络骄子。

1996 年以后, Java 的发展堪称火爆。Microsoft 和 IBM 两大公司都计划在 Internet 上销售用 Java 编写的软件。Apple、HP、IBM、Microsoft、Novell、SGI、SCO 等公司均计划将 Java 并入各自开发的操作系统,而负责开发并推广 Java 技术的 SunSoft 公司(这是 SUN 下属的一个子公司),将通过颁发许可证的办法来允许各家公司把 Java 虚拟机和 Java 的 Applets 类库嵌入到他们开发的操作系统中,这样各类开发人员就能更容易地选择多种平台来使用 Java 语言编程,用户也就可以脱离 Web 浏览器来运行 Java 应用程序,这无疑是很受广大用户欢迎的,这也为 Java 语言的应用开拓了极为广阔的前景。当然,各类 Java 操作系统之间的兼容性必须得到重视,好在 JavaSoft 公司已保证将监督这种兼容性。

有人预言 Java 将是网络上的“世界语”,今后所有的用其他语言编写的软件统统都要用 Java 语言来改写。工业界也有不少人预言“Java 语言的出现,将会引起一场软件革命”,这

是因为传统的软件往往都是与具体的实现环境有关，换了一个环境就需要作一番改动，耗时费力，而 Java 语言能在执行码（二进制码）上兼容，这样，以前所开发的软件就能运行在不同环境的计算机上，只要所用的计算机能提供 Java 语言解释器即可。

Java 语言将对未来软件的开发产生影响，可从如下几个方面考虑。

（1）软件的需求分析：Java 语言可动态地、可视化地描述用户的需求，给设计者提出更加直观的要求。用户的需求是各种各样的，受地区、行业、部门、爱好的影响，而这些都可以用 Java 语言来描述。

（2）软件的开发方法：由于 Java 语言面向目标的特性，因此完全可以用 O-O 的技术与方法来开发 Java 应用程序，这是符合最新的软件开发规范要求的。

（3）动画效果：Java 语言的动画效果比 GUI 技术更加逼真，尤其是它利用 WWW 提供的巨大动画资源空间，可以共享全世界的动态画面的资源。

（4）软件产品：用 Java 语言开发的软件具有可视化、可听化、可操作化的特点。

（5）其他：使用 Java 语言对开发效益、开发价值都有比较明显的影响。

1.1.2 什么是 Java

简要说来，Java 是由 Sun 公司开发的一种面向对象编程语言，其最初以 C++ 为原型，具有小巧、便捷、强健和面向对象等特点。这些特点使之成为许多平台和操作系统上都可以访问的 WWW 所“选择”的语言。

Java 具有两种开发类型，一是 Java 小程序，二是 Java 应用程序。

Java 小程序是动态、交互式、安全、跨平台的网络应用程序，因而可以广泛地用来扩展用户在 Web 上完成事务的类型。当一个 Java 小程序创建并编译之后，便可将它嵌入到一个 HTML（Hyper Text Markup Language）Web 页中，并可在 Web 站点上发布该 Web 页。当用户浏览该 Web 页时，浏览器将下载嵌入到该 Web 页的小应用程序的二进制码到用户的计算机上执行。

值得注意的是：

- （1）Java 小程序不能对本地文件系统进行读写；
- （2）除了下载它的主机外，Java 小程序不能与任何其他主机建立连接；
- （3）小程序不能调用用户工作站上的其他可执行代码。

Java 应用程序是独立的、脱离浏览器、可自行运行的二进制程序，它和用其他语言编写的应用程序一样，只是基于的语言是 Java。

1.2 Java 的特性

Java 是一个广泛使用的网络编程语言，它是一种新的计算概念。首先，作为一种程序设计语言，它简单、面向对象、不依赖于机器的结构、具有可移植性、鲁棒性、安全性，并且提供了并发机制，具有很高的性能。其次，它最大限度地利用了网络，Java 的小应用程序（Applet）可在网络上传输而不受 CPU 和环境的限制。另外，Java 还提供了丰富的类库，使程序设计者可以很方便地建立自己的系统。Java 与 C、C++ 语言相比更具有优越性。

1.2.1 Java 的特点

1. 简单性

Java 语言是一种面向对象的语言，它通过提供最基本的方法来完成指定的任务，只需理解一些基本的概念，就可以用它编写出适合于不同情况的应用程序。Java 不使用头文件，摒弃了 C 和 C++ 语言中令人头疼的预处理程序，它略去了运算符重载、多重继承等模糊的概念，取消了 C 语言中的 Struct 和 Union 等结构，不再支持 Goto 语句，并且通过实现自动垃圾收集大大简化了程序设计者的内存管理工作。Java 语言虽然是一门全新的语言，但他吸收了很多 C 和 C++ 语言中的精华，所以对于熟悉 C 和 C++ 语言的程序员来说，对于 Java 是不陌生的，并且容易上手，步步深入。

2. 面向对象

Java 语言的设计集中于对象及其接口，它提供了简单的类机制以及动态的接口模型，高度抽象，除了一些简单的数据类型，比如整形、布尔型以外全部采用纯面向对象编程的思想和方法，因此程序员可集中精力在数据和数据的操作上，而不必按照过程的观点去思考。Java 还带有一套扩展的类包 (Package)，在程序中可以轻松调用，简化了程序员的工作。

3. 分布性

Java 是面向网络的语言。通过它提供的类库可以处理 TCP/IP 协议，用户可以通过 URL 地址在网络上很方便地访问其他对象。

4. 鲁棒性

通过集成的面向对象的例外处理机制，在编译时，Java 语言提示出可能出现但未被处理的例外，帮助程序员正确地进行选择以防止系统的崩溃。

5. 安全性

Java 语言不支持指针，一切对内存的访问都必须通过对象的实例变量来实现，这样就防止程序员使用“特洛伊”木马等欺骗手段访问对象的私有成员，同时也避免了指针操作中容易产生的错误。

6. 体系结构中立

Java 解释器生成与体系结构无关的字节码指令，只要安装了 Java 运行系统，Java 程序就可在任意的处理器上运行。这些字节码指令对应于 Java 虚拟机中的表示，Java 解释器得到字节码后，对它进行转换，使之能够在不同的平台运行。

7. 可移植性

与平台无关的特性使 Java 程序可以方便地被移植到网络上的不同计算机上。同时，Java 的类库也实现了与不同平台的接口，使这些类库可以移植。另外，Java 编译器是由 Java 语言

实现的, Java 运行时系统由标准 C 语言实现, 这使得 Java 系统本身也具有可移植性。

8. 解释执行

Java 解释器直接对 Java 字节码进行解释执行。字节码本身携带了许多编译信息, 使得连接过程更加简单。

9. 高性能

和其他解释执行的语言如 BASIC 不同, Java 字节码的设计使之能很容易地直接转换成对应于特定 CPU 的机器码, 从而得到较高的性能。

10. 多线程

多线程机制使应用程序能够并行执行, 而且同步机制保证了对共享数据的正确操作。通过使用多线程, 程序设计者可以分别用不同的线程完成特定的行为, 而不需要采用全局的事件循环机制, 这样就很容易地实现了网络上的实时交互行为。

11. 动态性

Java 的设计使它适合于一个不断发展的环境。在类库中可以自由地加入新的方法和实例变量而不会影响用户程序的执行。并且 Java 通过接口来支持多重继承, 使之比类继承具有更灵活的方式和扩展性。

1.2.2 丰富的类库

Java 语言提供了大量的类, 以满足网络化、多线程、面向对象系统的需要。

(1) Java 语言包提供的支持包括字符串处理、多线程处理、例外处理、数学函数处理等。用户可以用它简单地实现 Java 程序的运行平台。

(2) 实用程序包提供的支持包括哈希表、堆栈、可变数组、时间和日期等。

(3) 输入输出包用统一的“流”模型来实现所有格式的 I/O, 包括文件系统、网络、输入。低级网络包用于实现 Socket 编程。

(4) 抽象图形用户接口包实现了不同平台的计算机之间的图形用户接口部件, 包括窗口、菜单、滚动条、对话框等, 使得 Java 可以移植到不同平台的计算机上。

(5) 网络包支持 Internet 的 TCP/IP 协议, 提供了与 Internet 的接口。它支持 URL 连接、WWW 的即时访问, 并且简化了用户/服务器模型的程序设计。

1.2.3 Java 和 C、C++

对于变量声明、参数传递、操作符、流控制等, Java 使用了和 C、C++ 相同的传统, 使得熟悉 C、C++ 的程序员能很方便地进行编程。同时, Java 为了实现其简单、鲁棒、安全等特性, 也摒弃了 C 和 C++ 中许多不合理的内容。

(1) 全局变量: Java 程序中, 不能在所有类之外定义全局变量, 只能通过在一个类中定义公用、静态的变量来实现一个全局变量。Java 对全局变量进行了更好的封装, 而在 C 和

C++中，依赖于不加封装的全局变量常常造成系统的崩溃。

(2) Goto 语句: Java 不支持 C、C++中的 Goto 语句，而是通过例外处理语句 try、catch、final 等来代替 C、C++中用 Goto 语句处理遇到错误时跳转的情况，使程序可读性更强且更结构化。

(3) 指针: 指针是 C、C++中最灵活，也是最容易产生错误的数据类型。由指针所进行的内存地址操作常会造成不可预知的错误，同时通过指针对某个内存地址进行显式类型转换后，可以访问一个 C++中的私有成员，从而降低安全性，造成系统的崩溃。Java 语言不支持指针的操作，因而克服了 C、C++固有的缺点。

(4) 内存管理: 在 C 语言中，程序员通过库函数 malloc()和 free()来分配和释放内存，C++中则通过运算符 new 和 delete 来分配和释放内存。再次释放已释放的内存块或未被分配的内存块，会造成系统的崩溃。而 Java 程序可创建并动态维护数据结构需要的内存申请，还可以自动完成内存垃圾的收集工作。

(5) 头文件: C 和 C++程序中使用头文件来声明类、全局变量、库函数等，在大的系统中维护这些头文件是非常困难的，Java 不支持头文件，类成员的类型和访问权限都封装在类中，运行时对访问进行控制，防止对私有变量的访问。

(6) 结构和联合: C 语言中的 Struct 和 Union 等结构，由于其成员均为公有，因而带来了诸多安全性的问题，并且由于其设计思想有悖于面向对象的设计思想，所以 Java 中取消了结构 Struct 和联合 Union，所有的内容都封装在类中。

(7) 预处理: C、C++中用宏定义来实现的代码给程序的可读性带来了困难。在 Java 中，不支持宏，它通过关键字 final 来声明一个常量，以实现宏定义中广泛使用的常量定义。

(8) 数据类型的支持: 在 C、C++中，对于不同的平台，编译器对于简单数据类型如 int、float 等分别分配不同长度的字节数，例如 int 在 IBM PC 中为 16 位，在 VAX-11 中为 32 位，这导致了代码的不可移植性。但在 Java 中，对于这些数据类型总是分配固定长度的位数，如对 int 型，它总占 32 位，这就保证了 Java 的平台无关性。

(9) 类型转换: 在 C、C++中，可以通过指针进行任意的类型转换，常常带来不安全性，而 Java 中，运行时系统对对象的处理要进行类型相容性检查，以防止不安全的转换。

1.2.4 应用前景

Java 语言有着广泛的应用前景，大体上可以从以下几个方面来考虑其应用。

- (1) 所有面向对象的应用开发，包括面向对象的事件描述、处理、综合等。
- (2) 计算过程的可视化、可操作化软件的开发。
- (3) 动态画面的设计，包括图形图像的调用。
- (4) 交互操作的设计（选择交互、定向交互、控制流程等）。
- (5) Internet 系统管理功能模块的设计，包括 Web 页面的动态设计、管理和交互操作设计等。
- (6) Intranet（企业内部网）上的软件开发（直接面向企业内部用户的软件）。
- (7) 与各类数据库连接查询的 SQL 语句实现。
- (8) 其他应用类型的程序。

1.3 Java 体系结构

Java 体系结构的优越性充分表现在 Java 平台无关性这一特点上,而要做到平台无关性关键是字节码的使用,因此在 Java 体系中产生了一个重要的机制就是 Java 虚拟机。

1.3.1 Java 虚拟机

Java 语言可以编译,也可以被解释。通过编译器,可以把 Java 翻译成一种中间代码——字节码,这种字节码是可以被 Java 解释器解释的、独立于平台的代码。通过解释器,每一条 Java 字节码被分析,然后在计算机上执行。

Java 虚拟机 (Java VM) 这个概念的引入,就是在机器和编译程序之间加入了一层抽象虚拟的中间件。这个虚拟机在任何平台上都提供给编译程序同样的接口。编译程序只面向虚拟机,编译程序将 Java 代码编译成虚拟机代码,然后由解释器将虚拟机代码解释成特定系统的机器码执行。也就是说 Java 编译器是平台无关的,而 Java 虚拟机是平台无关的关键。每一种 Java 解释器,不管是任何 Java 开发工具,还是 Web 浏览器,都是 Java 虚拟机的实例。

可以说,Java 虚拟机是 Java 语言的基础,Java 虚拟机并不识别 Java 程序设计语言,它只识别 Java 字节码。Java 虚拟机是一个抽象的计算机,和实际的计算机一样,它有自己的指令集,并使用不同的存储区域。Java 虚拟机负责执行指令,还要管理数据、内存和寄存器。Java 虚拟机负责将字节码翻译成特定机器的机器码。

Java 虚拟由各种抽象的逻辑组件组成,包括:指令集、寄存器组、32 位栈、无用单元收集堆和方法区域,这几部分不依赖于任何实现技术和机器平台。Java 虚拟机的具体实现方式可以不同,可以软件实现,也可以硬件实现,依赖于不同的软硬件环境。

Java 虚拟机的设计原则,是尽可能的简单小巧,尽可能在所有计算机上都安装,这样使得 Java 能够在各种平台上都可以顺利执行。

1.3.2 Java 体系

Java 程序就是使用 Java 编程语言编写的各种应用程序。Java 程序运行的硬件或软件环境,称为平台。Java 平台包括 2 个部分:Java 虚拟机和 Java 应用程序接口。Java API 即 Java 应用程序接口,用于 Java 程序的使用和 Java 虚拟机的接口。Java 虚拟机用于将 Java 字节码翻译成特定机器的机器码。基于硬件的平台,即 Java 程序运行的特定的硬件环境。因此,Java 的体系结构如图 1-1 所示。

Java 程序		Java 平台
Java API		
Java 虚拟机		
基于硬件的平台		

图 1-1 Java 体系

由于 Java 虚拟机的实现，将具体的硬件环境和 Java 开发环境分割开来。同样的 Java 程序，使用不同 Java 解释器（即 Java 虚拟机）解释，就可以是实现平台的无关性和可移植性。

1.4 本章小结

本章主要介绍了关于 Java 的产生、发展和影响，以及 Java 的特点和优点。Java 独有的虚拟机机制，独特的体系结构更是使 Java 在网络世界发挥得淋漓尽致的基础。

第 2 章 JBuilder 6 编程环境

本章详细介绍了 JBuilder 6 的安装方法、集成环境的组成和主菜单、工具栏的使用方法，此外还简要介绍了如何使用该集成环境来开发简单的 Java Applet 程序。通过本章的学习，读者可以对 JBuilder 6 的开发环境有一个全面的认识。

2.1 JBuilder 6 开发环境概述

平台无关性是 Java 的一个重要特性，正是这个特性使得 Java 非常适用于以网络为中心的计算环境。因此，Java 一经推出就受到全世界的欢迎。随着 Java 语言地迅速发展，Java 开发工具也越来越多，如何选取最适合自己的开发环境，成为快速掌握 Java 语言的一个关键。

如果用户既要求易于进行可视化布局，又要求有较大的灵活性，JBuilder 就是最好的选择。JBuilder 允许程序员使用绝对坐标设计结构，然后再将其很方便地转换成网格，这使得管理网格的问题变得相当容易。

1. 结构易于创建

JBuilder 提供创建平台独立的 GUI 工具的方式与众不同，它默认使用一种叫做 XYLayout 的布局结构。这个特性允许程序员以所见即所得的方式将组件放到结构中。

但是 JBuilder 在创建所见即所得的设计和网格布局之间的连接时与其他的所见即所得设计器是不一样的。JBuilder 允许用户使用 XYLayout 设计一个带组件的结构，然后再将其转换到网格布局中而仍然保持绝大部分你想要的原有设计。

2. 结构的可维护性

在 JBuilder 中设置网格限制相当容易，对任何给定的组件用户都可以弹出一个网格限制对话框，在对话框中可以设置一些网格限制参数，如垂直或水平挤压、填充等。除此之外，用户也可以直接到网格限制属性对话框中进行设置。由于 JBuilder 的高版本的产品提供了 XYLayout 这样灵活的布局管理器，使得用户添加组件变得更加容易。

3. 可视化编程

JBuilder 提供了一个交互向导 (Interactive Wizard) 作为可视化编程工具。其目的是能够自动完成组件之间的一些简单的交互操作，并且 JBuilder 6 提供了丰富的组件库，使得初学者学习 Java 时能达到事半功倍的效果。

用户可以在自己编写的代码中将光标放到某个类上，单击鼠标右键并且选择“浏览光标处的符号”来跳转到类浏览器上。

2.2 JBuilder 6 的安装

这里介绍在 Java 可视化编程中最灵活的开发工具 JBuilder 6 的安装过程。

2.2.1 安装环境

JBuilder 6 Enterprise 是一个很复杂的软件，因此要求的机器配置也较高。以下是安装和运行 JBuilder 6 的最小系统要求：

- (1) Microsoft Windows XP、Microsoft 2000 (SP2) 或 NT 4.0 (SP6)；
- (2) Intel PentiumII 233MHz 或兼容处理器；
- (3) 最小内存 128MB，推荐使用内存容量 256MB；
- (4) 硬盘空间 700MB（根据安装的不同版本而定）；
- (5) CD_ROM；
- (6) SVGA 或更高阶显示器（800×600，256 色）；
- (7) 鼠标。

注意：JBuilder 6 包含 JDK 1.3.1 版本，建议使用该版本的 JDK 替代以前的版本。

2.2.2 安装步骤

1. 启动安装程序

首先将 JBuilder 安装光盘放入 CD-ROM 中，如果该光盘的安装程序没有自动运行，则在资源浏览器中打开该光盘，双击“install_windows.exe”文件，弹出安装启动界面，如图 2-1 所示。

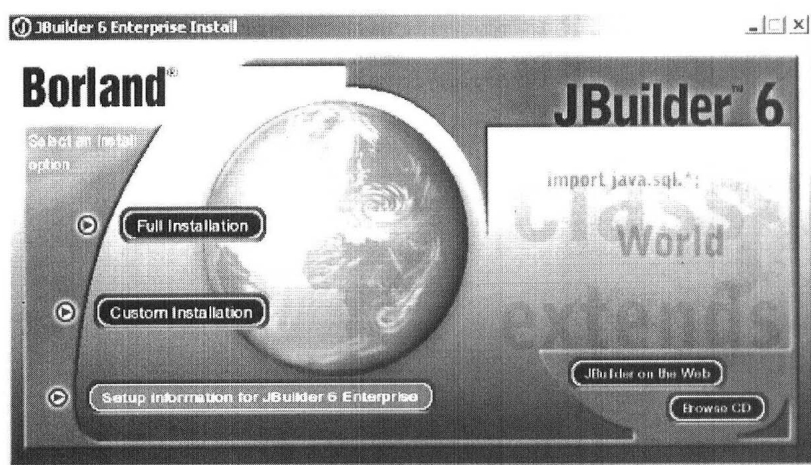


图 2-1 安装 JBuilder 启动界面