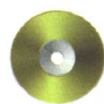




Borland
INPRISE

指定培训教材



随书赠送
Borland Development Tools
(Evaluation kit)



JBUILDER 3.0/4.0

标准教程

北京宝兰-英博思信息技术有限公司

Beijing Borland & Inprise IT Ltd., Co.

编著

清华大学出版社

<http://www.tup.tsinghua.edu.cn>



Borland _ INPRISE 指定培训教材

JBuilder3.0/4.0 标准教程

北京宝兰-英博思信息技术有限公司 编著
Beijing Borland & Inprise IT Ltd., Co.

清 华 大 学 出 版 社

本书配有光盘，需要者请到网络光盘实验室拷贝

(京)新登字 158 号

内 容 简 介

本书是 Borland-Inprise 公司的指定培训教材,书中系统而全面地介绍了这一强大而灵活的可视化 Java 编程语言开发工具。全书分为 32 章,针对 JBuilder3.0 开发环境的多领域应用进行了深入细致的讲解,包括使用 CUI 环境来创建基于关系数据库的客户机/服务器应用程序,如何创建类和 JavaBeans,以便提高当前应用程序或小程序的开发进度和改善工程性能。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: JBuilder3.0/4.0 标准教程

作 者: 北京宝兰-英博思信息技术有限公司

出 版 者: 清华大学出版社(北京清华大学校内,邮编:100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑:丁 岭

印 刷 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 24.5 字数: 610 千字

版 次: 2000 年 12 月第 1 版 2001 年 8 月第 2 次印刷

书 号: ISBN 7-900625-71-2

印 数: 4001~5000

定 价: 50.00 元

前 言

随着计算机通信和网络技术的发展,Java 这一门新兴的语言也在不断地发展。随着 Sun 向 Java 中加入新的功能,新版本也随之发布。目前的 JBuilder3.0 支持最新的 JDK 1.2,然而 JBuilder3.0 也可以配置成支持更新或较旧版本的 JDK。

JBuilder3.0 把 Java 编程语言的威力和图形用户界面的快速应用程序开发环境的易用性有机地结合在一起。全书共分 32 章、4 个附录,从 JBuilder3.0 的开发环境、用户界面设计、菜单设计、Java 编程、数据库操作等到软件的分发,进行了全面、系统的介绍。重点地讨论了 JBuilder3.0 的强大功能:

- (1) 利用 JBuilder3.0 中的向导如何创建应用程序、类和工程;
- (2) 设计器与双向开发工具的使用是如何使图形界面的设计变得十分容易;
- (3) JBuilder3.0 中如何使用 GUI 环境来创建基于关系数据库的客户机/服务器应用程序;
- (4) 如何方便地创建类和 JavaBeans,以便提高当前应用程序或小程序的开发进度,并改善以后工程的性能。

由于时间和水平所限,本书的编写难免存在不足之处,敬请读者批评指正。

本书主要由北京宝兰-英博思信息技术有限公司组编写,另外参加本书编写工作的同志还有赵洪利、朱诗兵、丁红勇、杜刚、刘友良、蒋太杰。

编 者

2000.8

目 录

第 1 章 Java 和 JBuilder 简介	1
1.1 Java	1
1.2 JBbuilder 简介	4
1.3 主窗口	5
1.4 应用浏览器(AppBrowser)	6
1.5 集成开发环境	9
1.6 Project 页	13
1.7 Opened 页	14
1.8 Directory 页	15
1.9 导航按钮	16
1.10 类体系浏览器与代码挖掘	17
1.11 使用帮助系统	20
1.12 从检视器中取得帮助	23
1.13 小结	24
第 2 章 使用工程	26
2.1 工程简介	26
2.2 使用工程向导创建工程	26
2.3 使用应用向导	29
2.4 向工程中加入文件	33
2.5 设置工程选项	34
2.6 小结	39
第 3 章 在用户界面设计器中使用组件	40
3.1 基于组件的开发	40
3.2 用户界面设计器	41
3.3 向应用中加入添加组件	43
3.4 检视器	45
3.5 组件树	48
3.6 容器	50
3.7 修改组件名称	52
3.8 同时选中多个组件	53

3.9 创建事件处理程序	55
3.10 小结	56
第4章 布局管理器	58
4.1 概述	58
4.2 容器与布局管理器	58
4.3 Border Layout	60
4.4 GridLayout	62
4.5 GridBagLayout	63
4.6 其他布局管理器	66
4.7 小结	67
第5章 组件概论	68
5.1 Jbuilder 组件	68
5.2 小结	74
第6章 菜单设计器	75
6.1 概述	75
6.2 使用菜单设计器	77
6.3 设计主菜单	79
6.4 设计弹出式菜单	86
6.5 JMenuBar 和 JPopupMenu	88
6.6 小结	94
第7章 Java 语言基础	95
7.1 Java 语法	95
7.2 Java 的数据类型	105
7.3 流程控制结构	111
7.4 小结	116
第8章 Java 面向对象编程	117
8.1 OOP (Object Oriented Programming) 简介	117
8.2 类	117
8.3 Java 包	128
8.4 小结	130
第9章 Java 的面向对象编程的高级技术	132
9.1 多态性	132

9.2 小结	142
第 10 章 使用 JBuilder 调试器	143
10.1 概述	143
10.2 调试环境	144
10.3 控制程序运行	147
10.4 使用断点	150
10.5 检查程序变量的值	155
10.6 定制调试器	159
10.7 小结	161
第 11 章 事件处理	162
11.1 概述	162
11.2 标准事件处理	163
11.3 Listener 接口	164
11.4 JBuilder 的 Adapter 类	166
11.5 匿名事件处理	168
11.6 小结	169
第 12 章 异常处理	170
12.1 概述	170
12.2 异常处理模型	170
12.3 Throwable 类	173
12.4 Java 内建的异常	174
12.5 案例分析:简易计算器	174
12.6 小结	175
第 13 章 JBuilder 的数据模型	176
13.1 在 JBuilder 中使用关系数据库简介	176
13.2 JBuilder 里的数据存取组件	181
13.3 小结	196
第 14 章 InterClient	197
14.1 概述	197
14.2 InterClient	197
14.3 小结	198

第 15 章 数据库工具	199
15.1 JDBC 浏览器	199
15.2 SQL 生成器	203
15.3 JDBC 监视器	207
15.4 小结	208
第 16 章 使用数据感知控件	209
16.1 开始使用数据感知控件	209
16.2 高级数据感知控件	212
16.3 小结	215
第 17 章 主从数据库窗体	216
17.1 主从关系	216
17.2 建立一个主从表用户界面	217
17.3 小结	222
第 18 章 在程序中操纵 QueryDataset	223
18.1 使用 Java 与数据库打交道	223
18.2 小结	230
第 19 章 数据模块	231
19.1 数据模块简介	231
19.2 建立数据模块	231
19.3 数据分割	236
19.4 小结	239
第 20 章 过滤、排序和查找	240
20.1 过滤	240
20.2 排序	244
20.3 查找	246
20.4 小结	248
第 21 章 列组件	249
21.1 列组件	249
21.2 选择列表	254
21.3 计算列	260
21.4 小结	262

第 22 章 带参数的查询与 DataSetViews	263
22.1 带参数的查询	263
22.2 DataSetView 组件	274
22.3 执行查询而不返回结果集	276
22.4 小结	277
第 23 章 存储过程	278
23.1 存储过程的定义	278
23.2 小结	283
第 24 章 数据集解决主案	284
24.1 使用 QueryResolver	284
24.2 使用 QueryResolver 的高级方法	287
24.3 小结	289
第 25 章 在运行时创建组件和数据驱动的用户界面	290
25.1 在运行时创建组件	290
25.2 案例分析:数据驱动的组件创建	292
25.3 小结	300
第 26 章 高级异常处理技术	301
26.1 建立定制的异常类	301
26.2 使用异常处理来保护资源	302
26.3 DataSetException 类	303
26.4 重载控件对 DataSetException 的默认处理	304
26.5 小结	305
第 27 章 JavaBeans 和 BeansExpress	306
27.1 概述	306
27.2 JavaBeans 组件模型	307
27.3 在 JBuilder 中创建 JavaBean	309
27.4 添加属性	311
27.5 向组件选项板中添加 JavaBean	313
27.6 小结	315
第 28 章 用户界面技术	317
28.1 桌面窗格(Desktop Panes)和内部窗体(JInternal Frames)简介	317
28.2 创建一个 MDI 风格的应用	317

28.3 小结	320
第 29 章 在 JBuilder3.0 中使用 DataStore	321
29.1 什么是 DataStore	321
29.2 什么时候使用 DataStores	321
29.3 建立一个脱机或公文包型的定单应用	321
29.4 JBuilder3.0 中与 DataStore 相关的特性	231
29.5 小结	324
第 30 章 dbSwing 高级特性	325
30.1 com.borland.dbSwing 包	325
30.2 dbSwing 组件的数据感知	326
30.3 使用文本	327
30.4 dbSwing、Swing 与 JBCL 的比较	327
30.5 小结	328
第 31 章 在 JBuilder 中创建 Applet	329
31.1 Applet 简介	329
31.2 Applet 的运行机制	330
31.3 在 JBuilder 中创建 Applet	331
31.4 建立数据库 Applet	334
31.5 添加 Applet 参数	335
31.6 小结	338
第 32 章 软件的分发	339
32.1 分发策略	339
32.2 命令行 Java	339
32.3 分发软件	342
32.4 小结	345
附录 A 理解 Java Object Hierarchy	346
A.1 引言	346
A.2 小结	347
附录 B Java 类库	348
B.1 引言	348
B.2 语言包	348
B.3 工具包	352

B.4 输入/输出包	355
B.5 JBCL 包	359
B.6 Java Generic Library (JGL)	361
B.7 小结	362
 附录 C 线程技术	 363
C.1 概述	363
C.2 创建一个线程	363
C.3 实现可运行接口	370
C.4 线程应用程序接口(API)	371
C.5 线程的生命周期	373
C.6 用代码保障线程安全	373
C.7 小结	374
 附录 D 串行化	 375
D.1 概述	375
D.2 为何要串行化	375
D.3 JDK1.1 中的串行化	375
D.4 运用输出流	377
D.5 运用输入流	378
D.6 读写对象流	379
D.7 小结	380

第 1 章 Java 和 JBuilder 简介

本章将讲述下列问题：

- ☐ Java 编程语言有何特点？
- ☐ 什么是 Java 虚拟机？
- ☐ 什么是 JBuilder？
- ☐ JBuilder 的主窗口位于何处？
- ☐ 什么是主窗口之下的大窗口？
- ☐ 什么是导航窗格、内容窗格和结构窗格？它们位于何处？
- ☐ 结构窗格上的符号各代表什么含义？
- ☐ 怎样打开用户界面设计器？
- ☐ 怎样定制源码编辑器，包括改变字体、键盘映射和语法高亮显示的设置？
- ☐ 怎样使用 Project 页、Opened 页与 Directory 页？
- ☐ 导航按钮有何用处？
- ☐ 如何通过“代码挖掘”查看一个类的定义？
- ☐ 联机帮助有哪些内容？
- ☐ 怎样使用联机帮助？
- ☐ 怎样在源码编辑器和对象检视器 (Inspector) 中使用帮助？
- ☐ 怎样定制结构窗格的显示方式？

1.1 Java

围绕 Java 这个词的使用和 Java 的含义存在着很多混淆的概念。或许这是因为 Java 实际上包含两种不同的概念：Java 虚拟机和 Java 编程语言。在本教程中，我们将使用 Java 语言创建 Java 应用程序(application)和 Java 小程序(applet)。“Java 语言基础”和“Java 面向对象编程”这两章将深入地讲解 Java 编程语言。不过，在介绍 Java 编程语言的概念之前，我们必须先理解什么是 Java 虚拟机。

Java 虚拟机

Java 虚拟机(JVM)是一个抽象概念，它给 Java 提供了高度开放的平台无关性。你可以这样来理解 Java 虚拟机：它是一台计算机，使用一种特殊机器语言来运行 Java 程序。它是基于堆栈的，字长为 32 位。这个机器指令集称为字节码，包括了把数值压栈、在栈内进行处

理和从栈里弹出之类的指令。它还需要相当于微处理器的寄存器的特殊内存区域。这种机制的优越性是大部分机器都可以用来模拟 JVM。

字节码一般是通过编译由 Java 语言编写的应用程序而产生,但是也有针对其他高级语言的编译器,像汇编语言。图 1.1 说明了高级语言代码、编译器、Java 虚拟机之间的联系。

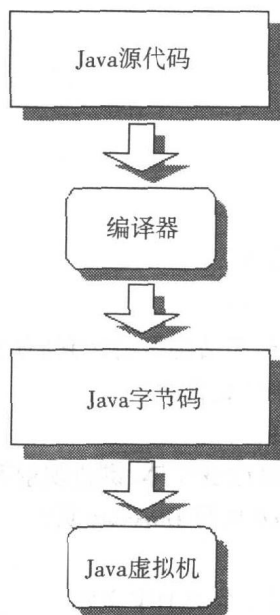


图 1.1 Java 虚拟机和 Java 语言

Java 虚拟机包括 4 个主要功能:

- 字节码装载
- 垃圾收集
- 安全性
- 提供与本地代码的接口

装载字节码

当 Java 程序要调用其他类或库的代码时,JVM 的任务就是从磁盘或网络上载入所需要的字节码。如果另一个类也调用了这个方法,JVM 会知道这个类的代码已经载入。JVM 还会有选择地卸出不使用的代码以节省内存资源。

垃圾收集

一个长期困扰程序员的问题就是内存的分配和释放。程序员经常不经意地产生内存泄漏,这通常是因为程序员分配了一块内存之后却忘记恢复或改变了内存的指针。Java 对内存的处理与其他语言稍有不同,JVM 会自动跟踪每个对象的引用。当某个对象不再被引用时,JVM 自动释放这个对象使用的所有内存。

安全机制

最让人感兴趣的可能还是 Java 内建的安全机制。当字节码在载入 JVM 时,规则检验机制会确定代码是否安全。具体来讲,规则检验机制确保以下内容:

- 代码不会试图访问未初始化的内存
- 禁止非法的类型转换。例如,JVM 不允许整数类型被当作指针类型
- 应用程序不会导致堆栈越界

规则检验机制不仅使 Java 程序更安全,还使它们更健壮而不易崩溃。通常,如果一个程序违背了上述任一准则,则该程序违背了安全机制,或者说包含了 Bugs。

Java 安全机制还能阻止不可信的程序访问类似硬盘这样的本地存储器,并且可以阻止不可信的程序访问网络。我们将在第 32 章“分发你的软件”中详细介绍这些内容。

访问本地代码

很多操作系统都有一些功能没有纳入到 Java 标准中。例如,Java 不能支持如串口、声卡及一些特定 I/O 设备等。但 JVM 可以调用本地代码(本机微处理器的机器码)。记住,对本地代码的使用应尽量少。应用程序使用了本地代码后便失去了平台无关性。

Java 语言

本教程的大部分内容旨在提高对 Java 编程语言或者“Java”的理解。Java 是一种面向对象的编程语言。Java 语言最早是由 Sun Microsystem 公司的一个进行电视顶置盒和嵌入式系统开发的小组发明的。后来,Sun 内部很多人认识到将 Java 用于开发 Internet 上的分布式程序的潜力。

一眼看上去,很容易把 Java 程序跟 C++ 程序弄混,因为两者的语法和控制结构非常相似(请参照“Java 语言基础”一章)。这种相似性使 C++ 程序员学习 Java 语言很容易上手。但请不要误会,Java 和 C++ 其实有很多不同。总体来说,Java 减少了 C++ 很多复杂的易导致错误的方面,如多重继承、预处理宏和运算符重载。Java 的创造者希望通过创建一门简单的语言来鼓励大家编写简单易懂的程序。

以下是 Java 语言的主要特点:

- 纯面向对象,即不能在类外面定义数据和函数。
- 所有对象都派生自同一个基类——Object,并共享它所有的功能。
- 通过接口,支持有限形式的多重继承。接口和面向对象编程会在“面向对象的 Java 编程”和“面向对象编程高级技术”这两章深入讨论。
- Java 的基本数据类型(比如浮点数和整数)是定义在一定的范围内,不依赖具体平台的实现。
- 程序员永远不需要显式地释放一块和对象相关联的内存,收集无用单元(前面讨论过,由 JVM 实现)是 Java 内建的机制。
- Java 具有一个巨大的标准代码库,以完成像网络通信、显示图形用户界面和管理数

据结构这样的功能。

- Java 不允许对指针进行算术运算, 具有很高的安全性。

Java 的版本

Java 是一门新兴且不断发展的语言。随着 Sun 向 Java 中加入新的功能, 新版本也随之发布。Java 版本编号的依据是最初支持它的 JDK(Java Development Kit)版本。版本以 X.Y.Z 的形式编号。其中 X 是主版本号, Y 是具有新特性的次版本号, Z 标识错误的修正和补丁。所以如果有人说他的应用程序是为 JDK1.2 开发的, 那么当你运行这个应用时, 你的 Java 环境(包括 Java 虚拟机与 Java 类库)需要支持 Java 1.0 版的第三个补充版本的 Java 标准。目前的 JBuilder3.0 支持最新的 JDK1.2。然而 JBuilder3.0 也可以配置成支持更新或较旧版本的 JDK。

Sun 的 JavaSoft 部门负责维护一个关于 Java 最新消息的网页, 网址是: <http://java.sun.com>。

1.2 JBuilder 简介

如果你选择 JBuilder 作为开发 Java 程序的环境, 显而易见它是最强大、最易于使用的 Java 快速应用程序开发(RAD)工具。

JBuilder 是最强大、最易于使用的 Java 快速开发工具。由于历史原因, 许多软件开发人员不愿意使用快速开发(RAD)工具, 因为他们总感觉这样的开发环境和语言不能满足他们所需要的灵活性和可控性。而 JBuilder 则把 Java 编程语言的威力和图形用户界面(GUI)的 RAD 环境的易用性有机地结合在一起。

本教程将讨论 JBuilder 的以下强大功能:

- JBuilder 中的向导可以使创建应用程序、类和工程的过程变得非常简单。
- 设计器与双向开发工具的使用使图形界面的设计变得多么容易。
- JBuilder 中如何使用 GUI 环境来创建基于关系数据库的客户机/服务器应用程序。
- 如何方便地创建类和 JavaBeans, 以便提高当前应用程序或小程序的开发进度, 并提高以后的工程的性能。

在讨论这些主题之前, 有必要介绍一下 JBuilder 的开发环境。图 1.2 显示的是 JBuilder 安装之后第一次运行的画面。

请注意, 一个题为 Welcome 的工程最先被打开。这个工程提供了一条熟悉 JBuilder3.0 的很好的途径。你只需选择“Begin your tour”按钮, 然后它会引导你了解 JBuilder3.0 的基本要素。Welcome 工程还提供了 hints、tips 和 shortcuts 的特性。

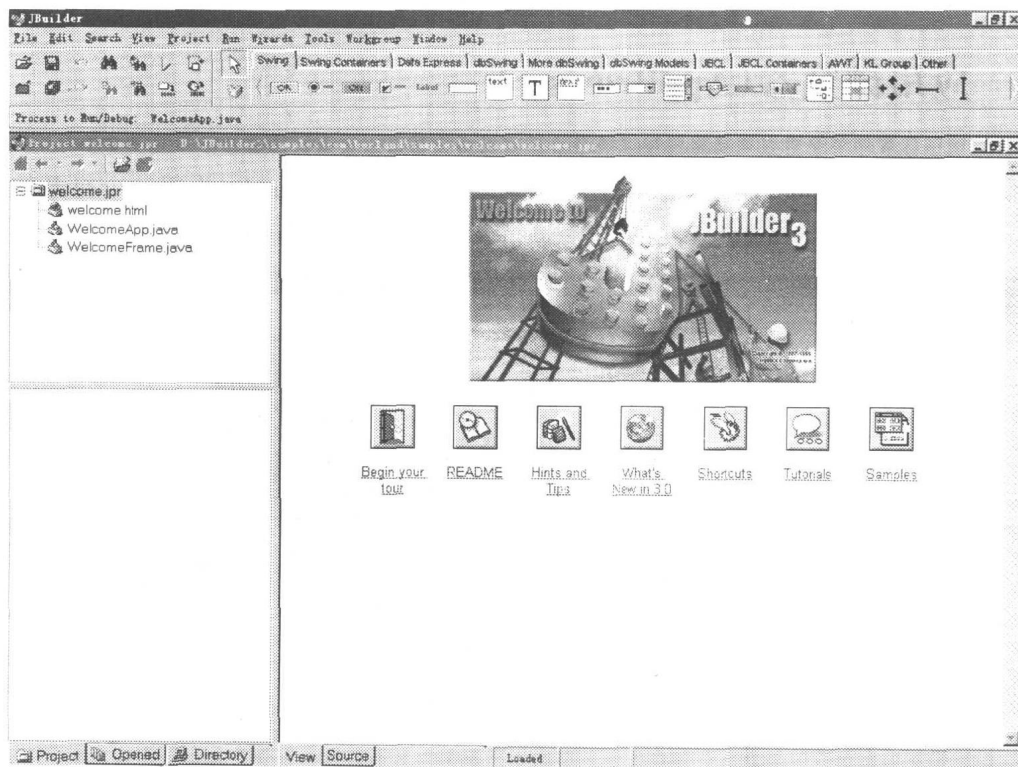


图 1.2 JBuilder 启动画面

1.3 主窗口

图 1.3 显示的 JBuilder 主窗口是开发过程的控制中心。主窗口提供了一个简单的方法来访问主菜单和工具条。主菜单和工具条在 JBuilder 开发过程中起着重要的作用。还有几个用来激活常用菜单项的快捷键。例如,如果同时按下 Ctrl 和 S,将激活 Save 菜单项。这些快捷键的定义在对应菜单项上都有说明。

主窗口还包括了组件选项板和状态条。状态条位于主窗口的底部,它显示了来自 JBuilder 的最近消息。对菜单项、工具条和组件的讨论贯穿了本章和全书。

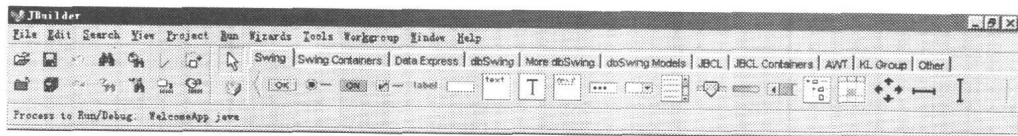


图 1.3 JBuilder 的主窗口

1.4 应用浏览器(AppBrowser)

除主窗口外,JBuilder 还显示另一个窗口——AppBrowser。JBuilder 为每一个打开的工程显示一个 AppBrowser。图 1.4 显示的是 Welcome 工程的 AppBrowser 窗口。

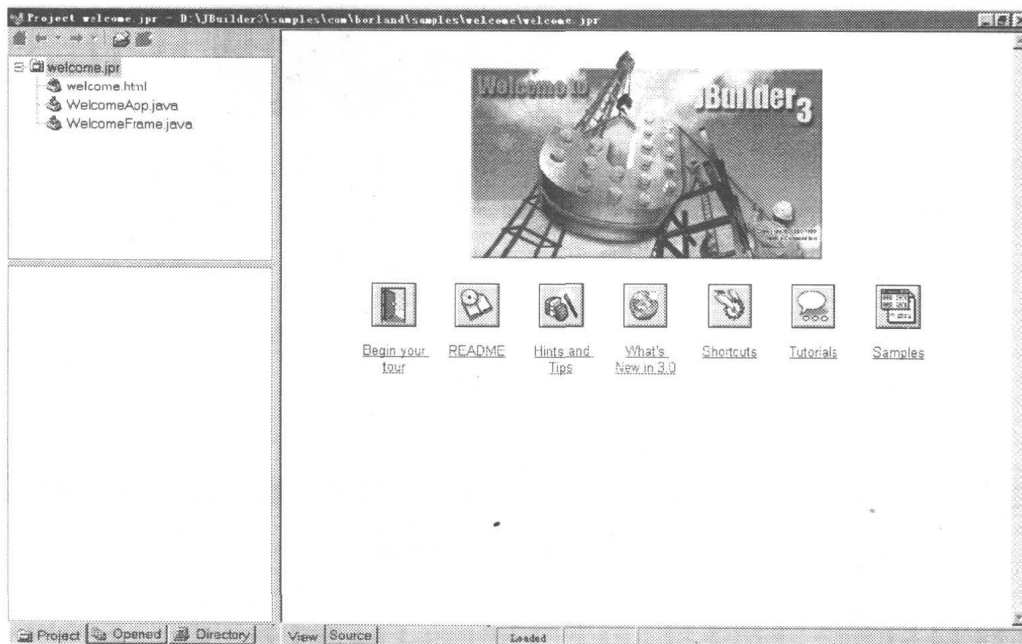


图 1.4 AppBrowser

AppBrowser 包括三个窗格,窗格的内容依赖于当前打开的文件与窗格的当前模式。如图 1.5 所示,这三个窗格分别是:

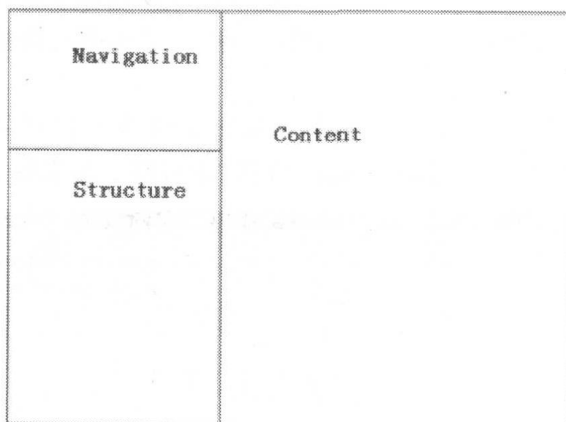


图 1.5 JBuilder 的三个主窗格