

JBuilder 6

实例入门

中青电脑艺术部 / 策划
袁 阳 / 主编 刘 勇 / 编著

- ◇ 面向初中级读者，从基础知识讲起。
- ◇ 每章一例，以实例贯穿全章。
- ◇ 采用 Step by Step 的讲解方式，结合实例的设计思路进行创意与扩展。
- ◇ 覆盖面广，涵盖 JBuilder 编程的方方面面。



随书附赠光盘，内含本书所用到的素材及实例的源程序，读者可对其参考引用。



海洋出版社



中国青年出版社

实例通解系列丛书

JBuilder 6

实例入门

中青电脑艺术部 / 策划
袁 阳 / 主编 刘 勇 / 编著



海洋出版社



中国青年出版社
CHINA YOUTH PRESS

(京)新登字 083 号

本书由海洋出版社与中国青年出版社合作出版。未经出版者书面许可,任何单位和个人不得以任何形式复制或传播本书的部分或全部内容。

图书在版编目(CIP)数据

JBuilder 6 实例入门 / 刘勇编著. — 北京: 海洋出版社, 2002.6

ISBN 7-5027-5815-1

I. J... II. 刘... III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2001)第 040657 号

总 策 划: 胡守文

王修文

郭 光

责任编辑: 钱晓彬

曹 建

黄 谊

责任校对: 王志红

丛 书 名: 实例通解系列丛书

书 名: JBuilder 6 实例入门

主 编: 袁 阳

编 著: 刘 勇

出版发行: 海洋出版社

地址: 北京市海淀区大慧寺 8 号 邮政编码: 100081

中国青年出版社

地址: 北京市东四 12 条 21 号 邮政编码: 100708

电话: (010) 84015588 传真: (010) 64053266

印 刷: 山东高唐印刷有限责任公司

开 本: 787 × 1092 1/16 **印张:** 21

版 次: 2002 年 8 月北京第 1 版

印 次: 2002 年 8 月第 1 次印刷

印 数: 1-5000

书 号: ISBN 7-5027-5815-1/TP · 589

定 价: 32.00 元 (附赠 1CD)

前言

在 JBuilder 发布之前,很多朋友问过我这样一个问题:要想学习 Java 编程的话,需要使用什么样的工具?这个问题在最初学习 Java 的时候也困扰了我很久。要知道在可视化编程满天飞的今天,让人使用 JDK 来开发一个应用程序几乎是不可想象的。因为使用 JDK 开发 Java 应用时,我们必须首先使用文本编辑器等工具完成代码的编写,然后使用 JDK 的命令行来编译和运行。如果程序没有任何错误,倒也还可以,但是关键在于,一旦编写的应用程序比较大,而且涉及到很多的类文件,要想保证代码一次编译成功几乎是不可能的。我们必须在 JDK 中找到程序出错的地方,然后到代码中进行修改,再编译,再修改。不用说,这样的过程需要程序开发人员具有极好的耐心才行。

但是 JBuilder 的出现,为我们提供了一个非常方便的开发各种 Java 应用的工具。使用 JBuilder,我们可以快速地开发和调试一个庞大的 Java 应用程序,并且完成程序的包装和发布。它是一组强大的编程工具的集合,用它可以编写出高品质的、易于升级的、与平台无关的 Java 应用程序。而且 JBuilder 本身就完全是用 Java 编写的,任何 Java 程序都可以在 JBuilder 中运行和调试。有了它,我们就可以完全告别 JDK 的行命令时代了。

“工欲善其事,必先利其器”,有了 JBuilder 这样一个非常好用的“器”,下面的工作就是如何使用它完成我们的“事”。本书用了十几个实例,来向读者展示如何才能把 Java 语言的强大功能同 JBuilder 的易用性、灵活性结合在一起。

目前市面上已经有不少介绍如何使用 JBuilder 的图书,不可否认这些书籍中有一些精品,但是其中一些的内容实在让人不敢恭维。有的书除了第 1 章中介绍了一些 JBuilder 开发环境中的菜单和命令外,在后面的章节中就再也看不到 JBuilder 的字样了,完全是在介绍 Java 语言。也有一些连篇累牍地介绍 JBuilder 中的新特性,如何使用 DataExpress 开发数据库应用,如何开发 XML 应用等等,但对于一个 Java 初学者,即使看完整本书,也学不到基本的 Java 应用程序的开发。

本书编写的基本思想就是力求把 Java 和 JBuilder 二者的优点结合起来向大家介绍,尽量使本书能够同时满足 Java 初学者,和那些具备 Java 程序开发基础知识,并开发过一定项目的程序员对于 JBuilder 的需要。对实例的选择更是以此为出发点,本书中所有实例都带有一定的 GUI 界面,放弃了一般 Java 入门图书中经常用到的控制台程序。但是实例的内容还是包括了基本的 Java 应用、数据库开发以及 Java 网络应用程序等。

本书中的实例大部分都是作者精心编写完成的,是作者在长期 Java 开发中的经验总结。有些实例的功能可以直接作为其他复杂应用程序中的单独模块来使用,比如在第 4 章中介绍的如何在应用程序中实现操作的撤销和重做 (Undo/Redo),在第 11 章中介绍的如何获得数据库结构信息,还有第 14 章的帮助系统的制作更是如此。这些程序的实用性都很强,相信经常从事工程开发的读者一定深有体会。

考虑到本书主要以初、中级读者为对象,所以在实例中并没有涉及到 JBuilder 中 XML 开发、EJB 开发以及其他分布式应用开发等方面的内容。上述这些内容无疑是 JBuilder 中最

147-65/11

为吸引人的地方,但是它们需要读者具有一定深度和广度的知识结构。本着宁缺勿滥的宗旨,作者在内容的选择上只好割爱了。希望有余力的读者,在学好本书的基础上,继续深入研究。

作者在上述内容的选择和介绍中,力求能够体现面向对象思想在程序开发中的应用,就是希望读者不要拘泥于某个应用程序的实现方法,而是从根本上提高程序的设计能力。如果读完本书之后,读者能够体会到面向对象思想应用的好处,那么我们的目的也达到了。

在本书的撰写过程中,得到了爱维图书的大力支持,袁阳先生为本书的结构做了总体策划,并在写作上给了我们许多宝贵的建议,张静茹小姐和梁红霞小姐对本书稿进行了审校和编排,在此深表感谢。

由于作者水平有限,难免留下了不足和遗憾。诚恳地希望读者把您的意见、建议和批评告诉作者,我们的电子信箱是: IVBOOK@LV99.net。

编者
2002年5月

1. 源代码

本书中所有实例的源代码位于光盘的“JBuilder 实例入门”文件夹中，并按照章节序号，分别放置在对应的文件夹下。

所有的源文件都是已经组织好的 JBuilder 工程，读者必须通过 .jpx 文件来打开整个工程，在其中编辑和运行实例。

2. 实例数据库

在本书中一共用到了两个 Access 数据库文件，这两个文件保存在“JBuilder 实例入门”文件夹中。其中在第 10~第 13 章中，构建数据库应用实例所需的 Access 数据库文件为 Biblio.mdb；第 18 章用到的数据库文件为 RegForm_MSAccess.mdb。读者使用时，需要将这些文件拷贝到自己的电脑硬盘中。



光盘使用说明

书附光盘中，包含了所有实例的源代码以及应用到的数据库。

目 录

第1章 制作欢迎屏幕

1.1 基础知识	3
1.1.1 面向对象程序设计的基本概念	3
1.1.2 Java 程序的基本结构	4
1.1.3 Java 中 GUI 程序设计	5
1.2 编程思路	7
1.3 操作步骤	7
1.4 创意与超越	12
1.5 本章小结	13

第2章 布局管理器

2.1 基础知识	16
2.1.1 Swing 中的容器和组件	16
2.1.2 FlowLayout 布局管理器	17
2.1.3 BorderLayout 布局管理器	17
2.1.4 CardLayout 布局管理器	17
2.1.5 GridLayout 布局管理器	17
2.1.6 GridBagLayout 布局管理器	18
2.2 编程思路	19
2.3 操作步骤	19
2.4 创意与超越	26
2.5 本章小结	27

第3章 文本编辑器

3.1 基础知识	30
3.1.1 流的基本概念	30
3.1.2 常用的字节流	30
3.1.3 常用的字符流	32
3.1.4 File,FileReader 和 FileWriter 类	32
3.1.5 本实例中使用到的几个组件	32
3.2 编程思路	34
3.3 操作步骤	34
3.4 创意与超越	43
3.5 本章小结	44

第4章 为文本编辑器添加编辑功能

4.1 基础知识	46
----------	----

4.1.1 Swing 中的事件处理	46
4.1.2 剪贴板功能的实现	48
4.1.3 Java 中 Undo 功能的实现	49
4.2 编程思路	49
4.3 操作步骤	49
4.4 创意与超越	53
4.5 本章小结	54

第5章 画图

5.1 基础知识	56
5.1.1 抽象类	56
5.1.2 矢量 Vector 对象	57
5.1.3 Cloneable 接口	58
5.1.4 鼠标事件和事件监听器	59
5.1.5 Java 中图形绘制的实现	61
5.2 编程思路	62
5.3 操作步骤	63
5.4 创意与超越	70
5.5 本章小结	71

第6章 实现一个图片浏览器

6.1 基础知识	74
6.1.1 模型—视图—控制体系 (MVC)	74
6.1.2 树	75
6.1.3 图像的缩放显示	76
6.2 编程思路	76
6.3 操作步骤	77
6.4 创意与超越	84
6.5 本章小结	85

第7章 使用 Applet 制作的时钟

7.1 基础知识	87
7.1.1 Applet——小应用程序的基本概念	87
7.1.2 Applet 应用程序的基本生命周期	88
7.1.3 线程的基本概念	89

7.2 编程思路.....	90	11.1.6 系统表.....	156
7.3 操作步骤.....	91	11.1.7 JDBC 中元数据的提取.....	156
7.4 创意与超越.....	98	11.2 编程思路.....	156
7.5 本章小结.....	99	11.3 操作步骤.....	156
第8章 使用JMF制作的媒体播放器		11.4 创意与超越.....	169
8.1 基础知识.....	102	11.5 本章小结.....	169
8.1.1 JMF的功能和结构.....	102	第12章 制作主从数据库窗体	
8.1.2 JMF所支持的文件格式.....	104	12.1 基础知识.....	172
8.2 编程思路.....	104	12.1.1 数据库中主键和外键的概念.....	172
8.3 操作步骤.....	104	12.1.2 DataExpress 中的主从关系.....	173
8.4 创意与超越.....	111	12.1.3 DataExpress 中如何定义 主从关系.....	174
8.5 本章小结.....	111	12.1.4 提取从数据.....	174
第9章 编写一个JavaBean 组件		12.2 编程思路.....	176
9.1 基础知识.....	113	12.3 操作步骤.....	176
9.1.1 JavaBean 的基本概念.....	114	12.4 创意与超越.....	181
9.1.2 JavaBean 的属性.....	115	12.5 本章小结.....	181
9.1.3 JavaBean 的自省.....	117	第13章 实现数据库的备份和恢复	
9.2 编程思路.....	117	13.1 基础知识.....	185
9.3 操作步骤.....	118	13.1.1 数据的备份和恢复.....	186
9.4 创意与超越.....	131	13.1.2 DataExpress 中使用文本文件 存储数据.....	187
9.5 本章小结.....	131	13.2 编程思路.....	188
第10章 数据库浏览器的制作		13.3 操作步骤.....	188
10.1 基础知识.....	135	13.4 创意与超越.....	195
10.1.1 JDBC 的基本概念.....	135	13.5 本章小结.....	196
10.1.2 JDBC 的组成部分.....	136	第14章 制作一个帮助系统	
10.1.3 四种 JDBC 驱动程序.....	137	14.1 基础知识.....	199
10.1.4 JBuilder 的 DataExpress 结构.....	138	14.1.1 TableDataSet 组件.....	199
10.2 编程思路.....	139	14.1.2 数据感知组件.....	200
10.3 操作步骤.....	140	14.2 编程思路.....	201
10.4 创意与超越.....	148	14.3 操作步骤.....	201
10.5 本章小结.....	149	14.4 创意与超越.....	207
第11章 数据库结构浏览器		14.5 本章小结.....	208
11.1 基础知识.....	153	第15章 网络中的客户/服务器程序	
11.1.1 关系数据库基础：实体——关系 模型.....	153	15.1 基础知识.....	213
11.1.2 表.....	154	15.1.1 TCP/IP 协议的基本概念.....	213
11.1.3 视图.....	154	15.1.2 TCP 套接字.....	214
11.1.4 索引.....	155	15.1.3 Java 中的 TCP 套接字类.....	215
11.1.5 同义词.....	155		

15.2 编程思路	215	A.3 JBuilder6 编程环境介绍和使用	282
15.3 操作步骤	216	A.4 定制 JBuilder 编辑环境	284
15.4 创意与超越	227	A.5 使用 JBuilder6 创建一个工程	285
15.5 本章小结	227	A.6 编辑源代码	288
第 16 章 一个时钟服务程序		A.6.1 编译程序	289
16.1 基础知识	231	A.6.2 调试程序	290
16.1.1 UDP 协议	231	A.6.3 使用帮助系统	291
16.1.2 Java 中的 UDP 实现	232	A.6.4 使用向导	292
16.2 编程思路	232	附录 B Java 语言说明	
16.3 操作步骤	232	B.1 程序结构	293
16.4 创意与超越	239	B.2 语法问题	293
16.5 本章小结	240	B.2.1 注释	293
第 17 章 使用 Servlet 收发 E-mail		B.2.2 标识符	293
17.1 基础知识	244	B.2.3 自变量	294
17.1.1 动态网页技术——Servlet	244	B.2.4 运算符及特殊分隔符	295
17.1.2 Servlet 的特点	245	B.3 数据类型	295
17.1.3 Servlet 的基本实现结构	246	B.3.1 数值类型	295
17.1.4 Servlet 的基本生命周期	247	B.3.2 布尔类型	297
17.1.5 Java Mail API 的介绍	248	B.3.3 数组	297
17.2 编程思路	249	B.3.4 类	298
17.3 操作步骤	249	B.4 运算符和表达式	300
17.4 创意与超越	259	B.4.1 算术运算符	301
17.5 本章小结	260	B.4.2 关系运算符	301
第 18 章 利用 JSP 制作一个网上注册程序		B.4.3 布尔逻辑运算符	302
18.1 基础知识	266	B.4.4 位运算符	302
18.1.1 什么是 JSP 及其特点	266	B.4.5 条件运算符	303
18.1.2 JSP 页面的执行过程	267	B.4.6 赋值运算符	303
18.1.3 JSP 的两种执行方式	267	B.4.7 表达式	303
18.1.4 JSP 基本的语法元素	268	B.5 控制语句	304
18.1.5 指令元素	269	B.5.1 条件语句 If...Else	304
18.1.6 脚本元素	270	B.5.2 多分支语句 Switch	304
18.1.7 动作元素	270	B.5.3 返回语句 Return	305
18.2 编程思路	271	B.5.4 While 语句	305
18.3 操作步骤	271	B.5.5 Do...While 语句	306
18.4 创意与超越	279	B.5.6 For 语句	306
18.5 本章小结	279	B.5.7 Continue 语句	306
附录 A JBuilder 6 的安装和使用		附录 C Java2 中的关键字	
A.1 JBuilder6 的安装要求	280	附录 D Java 中的异常处理	
A.2 JBuilder6 的安装过程	280	附录 E Java 中的常用类	

第 1 章 制作欢迎屏幕

知识点:

- Java 应用程序的基本结构。
- 事件处理的基本内容。
- 使用 Project 向导来建立新的工程。
- 使用 Class 向导来建立新的类文件。
- 初步设置工程编译、运行环境。

设计步骤:

1. 使用向导创建新工程。
2. 创建类文件。
3. 在窗体中添加组件。
4. 添加事件处理代码。
5. 处理主函数。
6. 编译工程并运行。

实例效果:



程序功能:

本程序实现了制作欢迎屏幕的功能。在程序运行时会显示一幅美丽的风景画，单击鼠标左键后图像消失，程序结束。

Java 语言自诞生之日起,就以其简单性、平台无关性等区别于其他传统程序设计语言的特性,吸引着广大程序开发人员的注意力。随着 Internet 的发展,Java 几乎导致了一场软件行业的变革。根据权威研究报告显示,2005 年全球超过 75% 的应用系统将会以 Java 为主要平台,未来 Java 开发人员数量增长幅度更为惊人,预测 2005 年将有 250 万人投入 Java 软件开发领域。

要想熟练地掌握 Java,一个好的开发环境是必不可少的。Borland 公司的 JBuilder 就是目前可选的 Java 强力开发工具之一。继 3.0、4.0、5.0 版本之后,2001 年 11 月 27 日 Borland 又宣布了新的 JBuilder 6,并于 2001 年 12 月上市。JBuilder 是针对建构企业级 Java 应用程序的跨平台开发环境,在 JBuilder 6 Enterprise 版中提供双向、视觉化 EJB 2.0 设计工具,让 EJB 组件开发过程变得更简易,协助企业快速导入 J2EE 平台。通过 UML 视觉化展现程序代码、关联异动(Refactoring)功能、单元测试(Unit Testing)向导、Javadoc 文件产生向导、支持多种程序代码版本控制系统等新功能。

从本章起,我们就为大家介绍如何利用 JBuilder 6 开发各种 Java 应用程序。

“千里之行,始于足下”。对于前面介绍的 JBuilder 6 提供的新特性,都是面向高级企业级应用开发的,对开发人员的要求也很高,没有一定的实践开发经验和知识积累,是很难掌握的。因此我们先从基本的 Java 应用程序设计开始,逐步向大家展示 JBuilder 的强大功能。首先,为大家介绍如何在 JBuilder 6 中开发一个欢迎屏幕。

无论学习使用何种高级程序语言,几乎每一个开发者都在开始阶段编写过这样的应用程序:Hello World。Hello World 程序非常简单,只是向屏幕输出一个字符串。本章要为大家介绍的也是一个与此类似的 Java 应用程序,只不过它向屏幕输出的不是一个字符串,而是一幅图片,我们称之为欢迎屏幕。开始运行 JBuilder 6 的时候,大家首先看到的就是如图 1-1 所示的一幅画面。

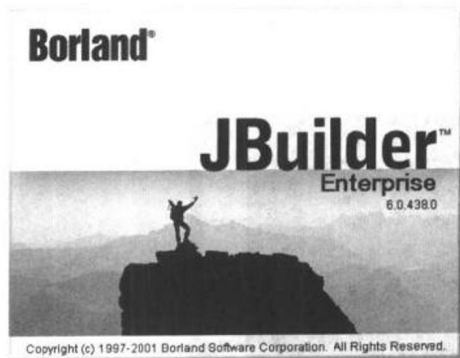


图 1-1 JBuilder 6 的欢迎屏幕

这幅画面就叫做欢迎屏幕,目前 Windows 中几乎所有常用的应用软件都有一个欢迎屏幕。为我们开发的应用程序制作这样的欢迎屏幕,不仅是宣传产品的方法之一,而且在程序长时间的初始化和启动过程中,欢迎屏幕还可以用来表示该应用程序正在准备过程中,不至于让用户面对单调的桌面,真正体现了应用程序人性化的一面。

本章中我们使用 JBuilder 6 开发了一个最简单的欢迎屏幕,如图 1-2 所示,即在程序运行时显示一幅美丽的风景画,单击鼠标后图像消失,程序结束。



图 1-2 欢迎屏幕的运行效果

下面，我们就为大家介绍这个实例的设计过程。

1.1 基础知识

在具体完成程序的实现方法之前，我们为大家介绍一些有关面向对象程序设计的基本概念以及 Java 程序的基本知识和结构。

1.1.1 面向对象程序设计的基本概念

面向对象技术是目前流行的系统设计开发技术，Java 就是一种纯粹的面向对象的程序设计语言，因此只有在了解面向对象的基本思想后才能深入理解 Java 中的许多概念。如果说一个程序设计语言是面向对象的语言，那么它必须具备以下几个特性：

- 封装性：隐藏实现细节。
- 多态性：将同一个消息发送给不同的对象，并使各个对象以预定的方式对消息作出响应的能力。
- 继承性：拓展现存的类，以生成专有类用来继承原来类的状态和行为的能力。
- 动态绑定性：在编程时不必知道对象的具体类型就能向它们发送消息的能力。

下面为大家介绍面向对象程序设计中最为基本的两个概念：对象和类。

对象是代表现实世界中的实际事物的软件抽象，比如说银行账号、计算机用户、用户界面上的按钮、窗口菜单等等。对象是由它们各自的状态和行为定义的，例如，一个银行账号拥有一种状态，诸如当前的收支状况，账号的所有人，允许的最小交易额等等；而它的行为则包括提取、存入、收支平衡等。

一个对象的状态只由对象自己内部的变量来确定，通常将这些变量称为对象的数据域或者对象的成员变量。一个对象的行为是由它上面的操作定义的，一般情况下这些操作被叫做方法。方法可以改变一个对象的状态、创建新对象、实现某些特定的功能等。

根据封装性，对象可以根据需要来设置其他对象对自己成员变量和方法的访问权限。一般成员变量和方法的访问权限可以分为三类：**Public**、**Private**、**Protected**。其中 **Public** 表示该成员变量或方法可以被任何外部对象来使用和访问；**Private** 表示该成员变量或方法不能被任何外部对象来使用和访问，即对外部是不可见的；**Protected** 表示该成员变量或方法只能在以本类作为父类的继承类中访问，对于其余外部对象是不可见的。

类是一个实体，它可以理解为对象的抽象，它定义了一个对象的运行方式以及在对象被

创建或者说实例化的时候所包含的数据（成员变量的取值）。类的作用如同一个模板，一个或者多个对象可以依照这个模板来创建。

下面我们通过一个简单的 Java 程序来看一下 Java 中是如何实现对象和类的。

1.1.2 Java 程序的基本结构

Java 程序设计是采用面向对象的设计思想，即所有的程序功能都是基于类的基础之上的。程序运行时，必须将所需的类实例化为对象，调用对象的方法和属性完成这种操作。

下面是一个基本的 Java 程序。

```
import java.lang.*;
public class HelloWorldApp{
public static void main (String args[]){
System.out.println("Hello World!");
}
}
```

上面的程序首先利用保留字 **Import** 引入 Java 中已经事先定义好的某些常用类，如 **Java.lang** 包中定义了完成有关程序输入输出的类。然后，用保留字 **Class** 来声明一个新的类，其类名为 **HelloWorldApp**，它是一个公共类（**Public**）。整个类定义由大括号“{}”括起来。

在该类中定义了一个 **Main()** 方法，其中 **Public** 表示访问权限，指明所有的类都可以使用这一方法；**Static** 指明该方法是一个静态方法，它可以通过类名直接调用；**Void** 则指明 **Main()** 方法不返回任何值。

对于一个应用程序来说，**Main()** 方法是必需的，而且必须按照如上的格式来定义。Java 解释器在没有生成任何实例的情况下，以 **Main()** 作为入口来执行程序。

Java 程序中可以定义多个类，每个类中可以定义多个方法，但是最多只能有一个公共类，**Main()** 方法也只能有一个。作为程序的入口，**Main()** 方法定义中，括号“()”中的 **String Args[]** 是传递给 **Main()** 方法的参数，参数名为 **Args**，它是类 **String** 的一个实例。参数可以是 0 个或多个，每个参数用“类名、参数名”来指定，多个参数之间用逗号分隔。

Java 程序是由类构成的。对于一个应用程序来说，必须有一个类中定义 **Main()** 的方法。在类的定义中，应包含类变量的声明和类中方法的实现。

以上的 Java 代码只是完成了一个简单的可运行的 Java 应用程序，其中并没有涉及类的实例化，以及类成员变量和方法的使用。我们可以在上面的类 **HelloWorldApp** 中添加一个成员变量 **Name** 和一个方法 **Show()**，并且在 **Main()** 函数中调用该方法，代码如下所示：

```
import java.lang.*;
public class HelloWorldApp{
    public String name;
    public void show(){
        System.out.println("Hello World!I am "+name);
    }
    public static void main(String args[]){
        HelloWorldApp app=new HelloWorldApp();
        app.name="JbuilderUser";
        app.show();
    }
}
```

上面的代码中，我们在 Main()函数中创建（实例化）了一个 HelloWorldApp 对象 App，设置该对象的成员变量 Name 为字符串“JBuilderUser”，然后调用该对象的 Show()方法来执行输出操作。由于 Show()方法没有声明为静态方法（Static），因此必须实例化一个对象才能利用该对象来执行该方法。

1.1.3 Java 中 GUI 程序设计

Java 也可以像 VB、VC 等可视化设计程序一样完成带有用户交互界面的应用程序的设计。而且，使用 Java 开发的可视化应用程序可以脱离特定的操作系统，这就为程序的移植带来了极大的方便。Java 中的可视化设计功能主要是依靠 Awt 和 Swing 组件来完成的。

从 Java 语言的出现到现在，图形和用户界面功能已取得了飞跃发展。Java2 平台包含一个复杂的跨平台的用户界面体系结构。它的组成包括众多的高级组件，一个先进的、丰富的、独立于设备的图形系统和许多多媒体扩展。

在 Java 2 平台发行之初，AWT 是 Java 平台实现图形功能的基础。例如 Swing 等各种技术都是作为可选的扩展被引入的。在 Java 2 平台上，这些扩展中的大部分都可以在平台的核心中找到它们的一席之地，成为 Java 基础类（Java Foundation Classes, JFC）的一部分。JFC 指的是包含在 Java 2 平台内的一整套图形和用户界面技术，包括 AWT 和 Swing 等。

Swing 是一个带有一套丰富组件的 GUI 工具包。它组成了 JFC 用户界面功能的核心部分。它不仅是 AWT 所提供的组件的替代品，而且在这些组件的基础上有了很大进步。

在 JDK 首次发行时，由于本机平台的集成环境具有优先权，于是 AWT 提供了一些组件，这些组件使用各平台的本机组件（在 Java 编程术语中，这些被称为重量级组件）来实现。例如，在 UNIX 平台上，Java.awt.Button 类是用窗口小部件 Motif PushButton 实现的。

相同的 Java 应用程序在每个平台上都有不同的外观，虽然外观不同但是在功能上是等同的。当然，这也是问题开始的地方。在简单的界面里，这种等同是真的。但随着复杂性的增加，情况就不一样了。这是因为组件有所不同，而且在某些情况下，不管修正了多少错误，不管 AWT 的组件被重写了多少次，它们总会表现得有些不同。

语言本身已经变得不只是一种平台了，GUI 也正朝着在所有平台上都有相同的外观和行为了这种方向转变。为了实现这个目标，本机组件必须尽量丢弃。但是，有一些本机代码还是必需的。如果不包含 X System Window 的调用，您就不能在 UNIX 上显示一个窗口。

现在来看一下 Swing，它使用 AWT 的一个子集来实现这个目标。这个子集包括基本画图操作和 Java.awt 包中的某些类（包括 Container, Window, Panel, Dialog 和 Frame）。每一个 Swing 组件都有一个自己的功能模型和一个独立的外观（Look-and-feel），这可以预先设置在运行时更改。

Swing 为 Windows 平台和 Motif 平台分别提供一个 Metal（也可以称做 Java Look-And-Feel），同时为 Macintosh 平台提供一个 Java Look-And-Feel 作为额外选项。平台的 Look-And-Feel 不像 AWT 一样使用平台的本机组件，相反它们使用轻量级组件，有和本机组件一样的外观，这对于它的功能来说是好的，但在外观或行为上总有一些不同。所以复杂的界面永远不会和使用本机组件的界面完全相同。

由于 JBuilder 6 开发环境也是使用 Java 开发的一个应用程序，所以我们同样可以改变它的外观。通过在 JBuilder 6 开发环境中打开 Tools 菜单中的 IDE_Options，然后选择 Browser

选项卡，在 Look And Feel 的下拉列表中选取我们希望采用的程序外观，选择的过程如图 1-3 所示。图 1-4、图 1-5 和图 1-6 分别展示了每一个 Look-And-Feel 的界面样式。

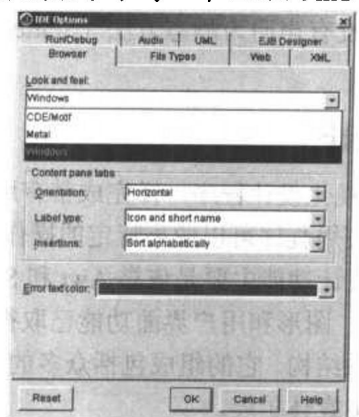


图 1-3 在 JBuilder 6 选取不同的外观

在本书中所有的组件采用的都是 Swing 组件，我们可以选择它们的外观。在本章的示例程序中我们用到了 JWindow 类。JWindow 是一个重量级 Swing 组件，它是对 Java.awt.window 类的扩展。JWindow 类的实例对象没有边框、标题栏和菜单栏，并且不能调整窗口的大小。

JWindow 类一般用于在无边框的区域中显示图形、组件。例如，本章示例程序所显示的欢迎屏幕就是在 JWindow 中显示的 JLabel 组件。



图 1-4 Windows 外观



图 1-5 Metal 外观



图 1-6 Motif 外观

1.2 编程思路

由于这是本书的第 1 个程序，因此，我们将功能尽量简化，程序的总体思路也比较简单，主要就是实现一个类对 JWindow 类进行的扩展，为 JWindow 类添加 JLabel 组件。在 JLabel 组件之上完成图形的显示，同时处理鼠标被单击的事件，关闭应用程序。

1.3 操作步骤

在本实例的介绍中，我们将为大家演示如何利用 JBuilder 6 创建一个新的工程。

步骤 1：使用向导创建新工程。

在 JBuilder 6 开发应用程序中，必须首先建立一个新的工程。所有的 Java 文件都位于这个特定的工程中，建立新工程的方法有两种。

一种是在 File 菜单中选择 New 选项或在工具栏中选择 New 按钮。然后，出现如图 1-7 所示的对象选择对话框，选择 Project 选项板的 Project 选项卡，单击 OK 按钮。

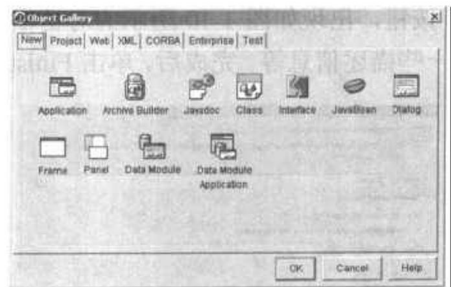


图 1-7 Object Gallery 对话框

另一种是在 File 菜单中直接选择 New Project 选项。此时，出现如图 1-8 所示的工程属性设置对话框，填入工程名、工程路径等。

工程的类型选择系统默认的为 Jpx 型，工程名为 WelcomeToJBUILDER 6。这个工程名的小写形式将作为封装工程所有类的包名。工程的路径 Directory 选择 E:/JBUILDERProgramme，如果该路径不存在，JBuilder 6 可以为我们建立相应的文件夹。这里需要注意的是，工程将在选定的 Root Path 下建立一个以工程名命名的文件夹，所有的工程文件分类（比如源文件放在 src

目录下等)放在这个文件夹下,填写完毕如图 1-8 所示。

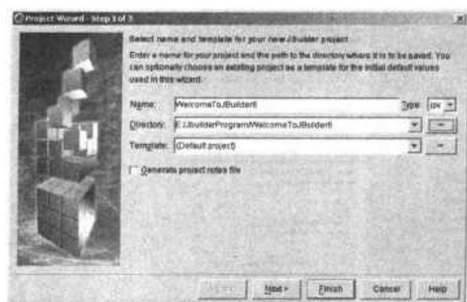


图 1-8 建立新工程向导

填写完毕后,单击 Next 按钮,会出现一个与工程有关的路径信息设置对话框,如图 1-9 所示。在其中我们可以看到,工程文件会按照各自的类型分别保存在不同的路径下。我们可以修改这些路径的名称,当然一般情况下,让 JBuilder 为我们默认设置即可。另外在这个对话框中,还可以设置工程需要加载的类库,这一内容,将在后面的章节中介绍。

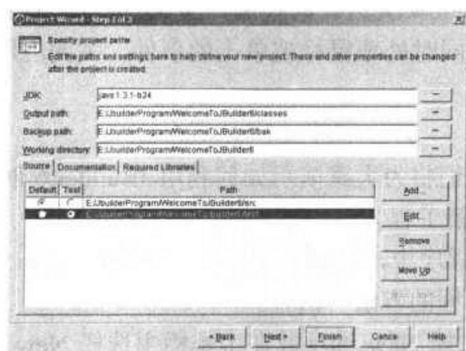


图 1-9 工程属性的确认

确认完成后,单击 Next 按钮,出现如图 1-10 所示的对话框,其中可以设置工程所采用的编码方式以及有关工程的一些描述信息等。完成后,单击 Finish 按钮就完成了工程的创建。

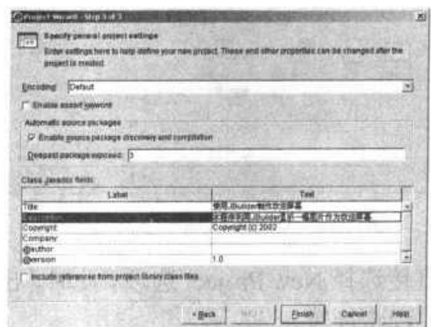


图 1-10 工程描述信息对话框

步骤 2: 创建类文件。

接下来就是为新的工程添加类文件。

在这里,我们为这个新的工程添加一个新的类文件,选择 File 菜单中的 New Class 项或