

How to Program JavaBeans



- 扩展 Java 的功能和互操作性
- 利用 Beans 构建可移植、可重用的 Web 构件
- 学习构建定制的 Beans, 使用已有的 Beans

使用 JavaBeans

[美] Michael Morrison 等著
郭越译

编程

随书所附光盘包含 JDK1.1.1 版、BDK1.0 版、《使用 Java 编程》一书的 HTML 格式版、Sun 公司的 Java Workshop Try-and-Buy1.0 版以及本书实例的源代码等



電子工業出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.com.cn>

28.15

使用JavaBeans编程

How to Program JavaBeans

[美] Michael Morrison, Randy Weems 著
Peter Coffee, Jack Leong

郭越译



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

内 容 简 介

随着JavaBeans开发工具包(BDK)的发布,交互构件的开发进入了追求易用性和灵活性的新领域。本书讨论了JavaBeans的基础知识和设计规则。全书共分四个部分,18章,分别介绍了:Java与构件的结合——JavaBeans,使用JavaBeans API,Beans的使用,JAR文件,用BeanBox测试Bean,以及Bean的作用和未来发展等。书中精心设计了5个Bean实例,所有实例的源代码均在随书所附的光盘上。

本书适合各类计算机编程人员和软件工程师,对大专院校师生也是很好的参考书。



Authorized translation from the English language edition published by Ziff-Davis press an imprint of Macmillan Computer Publishing U.S.A. Copyright© 1997.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the publisher.

SIMPLIFIED CHINESE language edition published by Publishing House of Electronics Industry. Copyright © 1998.

本书英文版由美国麦克米兰计算机出版公司的Ziff-Davis公司出版,已将中文版版权授予电子工业出版社。未经许可,不得以任何形式和手段复制或抄袭本书内容。版权所有,侵权必究。

原 书 名: How to Program JavaBeans

书 名: 使用JavaBeans编程

著 者: [美] Michael Morrison, Randy Weems, Peter Coffee, Jack Leung

译 者: 郭 越

责任编辑: 陆伯雄

印 刷 者: 天津新华印刷厂印刷

出版发行: 电子工业出版社出版、发行

北京市海淀区万寿路173信箱 邮编100036 发行部电话 68214070

URL: <http://www.phei.com.cn>

经 销: 各地新华书店经销

开 本: 787×1092 1/16 印张: 17.5 字数: 400千字

版 次: 1998年1月第1版 1998年1月第1次印刷

印 数: 6000册

书 号: ISBN 7-5053-4293-2

TP·1947

定 价: 45.00元

著作权合同登记号 图字: 01-97-1530

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换

版权所有·翻印必究

引言

对象、构件、GUI、快速应用开发、可视化编程，这些计算机界专门用语的不断出现开始让人觉得头晕目眩。可对于大多数软件开发人员来说，这些专用语并没有使他们的工作变得简单起来。构想出一个结构合理、完全自我约束的软件体系，让它以一种高度可视化的方式集成并用于构建应用程序可能是一件简单的事情，但实现起来却不那么容易。

尽管迄今为止还没有一项技术能够完全实现这一构想，但Sun公司的JavaBeans软件构件（software-component）技术已最为接近地靠近了目标。JavaBeans建立在已取得极大成功的Java编程语言和运行时间系统（runtime system）之上，它的出现实现了Sun公司的承诺：发布一项用于生成完全复用，独立于平台软件的技术。由于缺少软件构件支持是原Java难题中引人注目的一个部分，因此JavaBeans标志着Java又向前迈进了一大步。

本书从多个角度剖析了JavaBeans技术。你不仅可以从中学习支持JavaBeans技术的许多理论以及JavaBeans与核心Java API的关系，还可以尝试构造自己的JavaBeans构件，即Bean。除此之外，你还将了解JavaBeans技术未来的一些发展方向，包括这些方向与Microsoft的ActiveX技术的关系。

本书不仅详细地介绍了JavaBeans技术，同时也是一本开发JavaBeans构件的实用指南。我们的目标是：当你读完本书时，你已经具备了开发自己的JavaBeans构件的知识，同时也具备了开发的自信心。我们希望你会和我们一样，发现JavaBeans是一项令人感到振奋的技术，并从中体会到无穷的乐趣。

——Michael Morrison

目 录

引言

第一部分 JavaBeans初探

第一章 JavaBeans: Java与构件的结合	3
编程人员为什么需要构件	3
构件技术满足了人们长期的渴望	4
开发者需要规则	5
使用不是自己亲手构造的构件, 安全吗?	5
JavaBeans的目标	6
目标1: 是一个可重用的构件, 而不是七巧板的一个拼块	6
目标2: 在设计中可视化地操作	7
目标3: 能与编程工具一起使用	8
Java软件平台	13
Java语言	13
Java虚拟机	14
小 结	17
第二章 用于 JavaBeans开发的API简介	18
选择起始点	18
好的映像	19
Field类	20
Method类	25
Constructor类	26
Array类	27
JavaBeans API	27
JavaBeans API接口	27
JavaBeans API类	29
小 结	32

第三章 Bean的使用	34
JavaBeans和可视化工具	34
把Bean加入到一个可视化工具中	35
在可视化工具中使用Bean	39
用老式方法使用JavaBeans	40
手工创建Bean	41
手工定制Bean	42
小 结	43
第四章 用JAR文件包装 Bean	44
JAR基础知识	44
JAR文件是跨平台的	45
JAR文件支持不同类型的文件	45
JAR文件是后向兼容的	45
JAR文件提供了一个开放的、可扩展标准	45
JAR文件和JavaBeans	45
声明文件 (manifest file)	46
JAR实用程序	48
检测现有的文档	48
生成新文档	51
小 结	52
第五章 用BeanBox测试Bean	53
BeanBox简介	53
运行BeanBox	54
解剖BeanBox	54
在BeanBox中使用Bean	57
用事件连接Bean	58
保存并装入Bean	60
小 结	60

第二部分 使用JavaBeans API

第六章 Bean事件：事件处理API	65
今晚的主要事件	65
新旧模式的比较	65
过程化程序设计	66
事件驱动程序设计	66
发生了什么事件？告诉我更多有关事件的内容	66
JavaBeans 事件：三种模式	67
基于继承的事件处理	67
基于委派(delegation-based)的事件处理	69
实用的委派	72
从一个Bean到另一个Bean	72
接合器(adaptor)：使工作变得轻松	75
内嵌类(inner class)：使工作更加简单	78
再谈基于继承的事件处理	81
小结三种模式	83
关于事件源	84
小 结	84
第七章 Bean的内部存在：属性API	85
改变状态	85
在你自己的Bean中利用属性	88
在Name(名字)中有什么?	88
Constrained(受限制的)和Bound(受约束的)属性	89
定制属性编辑器	94
属性编辑器(PropertyEditor)接口：进一步的考察	100
各负其责	100
小 结	103

第八章 保存Bean：持久性和连续性API	104
一个真正的Bean保存器	104
Serializable接口	105
Externalizable接口	110
对象版本	111
小 结	112
第九章 了解Bean：内省API	113
建立映像	113
利用一个小类	114
描述特征	114
一般特征	115
描述方法和变元	117
描述事件	118
描述属性	123
描述Bean自己	125
BeanInfo接口	125
命名BeanInfo	126
小 结	128
第十章 作为工具的Bean：应用程序生成器API	129
我身处何地？	129
使用Bean定制器	130
定制的Arrow Bean	131
小 结	140
 第三部分 构建自己的Bean	
第十一章 图像按钮Bean	143
设计图像按钮Bean	143
属性	144
方法	144

事件	145
构造图像按钮Bean	146
属性和成员变量	147
构造器	147
访问方法	148
公用方法	150
事件注册方法	152
事件处理方法	152
支持方法	155
附加的开销	156
使用图像按钮Bean	158
小 结	160
第十二章 针示测量仪Bean	161
设计针示测量仪Bean	161
属性	162
方法	163
事件	164
构造针示测量仪Bean	164
属性和成员变量	165
构造器	165
访问方法	166
公用方法	168
支持方法	171
附加的开销	173
使用针示测量仪Bean	174
小 结	177
第十三章 时钟Bean	178
设计时钟Bean	178
属性	179

方法	179
事件	180
构造时钟Bean	180
属性和成员变量	180
构造器	181
访问方法	182
公用方法	183
支持方法	185
附加的开销	188
使用时钟Bean	189
小 结	190
第十四章 图像增强器Bean	192
图像过滤基础知识	192
设计图像增强器Bean	194
属性	194
方法	195
事件	196
构造图像增强器Bean	196
属性和成员变量	196
构造器	197
访问方法	198
公用方法	200
支持方法	201
附加的开销	203
使用图像增强器Bean	208
小 结	210
第十五章 计时器Bean	211
设计计时器Bean	211
属性	212

方法	212
事件	213
构造计时器Bean	213
属性和成员变量	213
构造器	214
访问方法	215
公用方法	216
事件注册方法	217
事件处理方法	217
支持方法	218
附加的开销	218
使用计时器Bean	219
小 结	222

第四部分 发挥Bean的作用

第十六章 用Bean、RMI和CORBA进行分布计算	225
网络环境中的Bean——综述	225
准备工作	226
远程方法调用简介	229
编写服务器程序	229
编写客户程序	232
使用rmic	235
将它们组合起来	235
Java IDL简介	236
Java IDL包	236
编写IDL文件	237
使用idlgen	237
编写远程方法的实现类	238
编写服务器程序	239
编写客户程序	240

组合在一起	243
小 结	244
第十七章 用桥连接JavaBeans和ActiveX	245
ActiveX桥基础知识	245
ActiveX桥的连接属性	246
ActiveX桥的连接方法	247
ActiveX桥的连接事件	247
在Visual Basic中使用Bean.....	247
包装Bean.....	247
在Visual Basic中加入Bean	250
在Visual Basic中使用Bean	252
小 结	255
第十八章 JavaBeans的未来	256
发展空间	256
100%纯Java.....	256
定义100%纯Java	257
JavaOS	260
HotJava Views.....	263
Java电子贸易 (Java Electronic Commerce)	265
小 结	266
附录	267

第一部分 JavaBeans初探

第一章 JavaBeans: Java与构件的结合

第二章 用于JavaBeans开发的API简介

第三章 Beans的使用

第四章 用JAR文件包装Bean

第五章 用BeanBox测试Bean

第一章 JavaBeans: Java与构件的结合

你一定知道，当你坐在计算机前键入第一行程序代码时，最艰难的时候就要开始了。就像在稿纸的第一行写下“第一章”一样，你将面临一系列的挑战。

你将写出的作品是由人阅读，还是由机器以某种方式进行解释在这里似乎无关紧要。无论这两种方式中的哪一种，您都必须提出一个约定，在约定中确定一组需求，并提出有效的解决方案。

“无论哪一种方式”，这是一个极有分量的承诺。本书介绍的JavaBeans为你信守此承诺提供了一条较好的途径。JavaBeans技术通过简化包装和重用代码的处理，使一个好的解决方案能够用在任意需要之处，从而为程序用户提供了一个更快、更灵活，同时也更加可靠的解决方案。

第一章的内容包括支持JavaBeans的一些基本概念，其中有：

- ▶ 基于构件（component）开发的逻辑
- ▶ 使用买来的构件的危险因素
- ▶ JavaBeans的目标：重用性、可视化开发和工具
- ▶ Java软件平台：Java语言和Java虚拟机（Java Virtual Machine, JVM）

编程人员为什么需要构件

我们已经说过，使用代码的编程者与使用文字的写作者所面临的挑战几乎是同样的。但是，同使用代码编写机器可读作品的编程人员相比，用文字写作的人有着得天独厚的优势。

他们的优势就在于其作品的读者拥有各种各样的资源，例如提供了有价值的文字关系的字典和众所周知的文字体系等。因此即使是一页空白，作者也不要担心，他完全可以认为读者会从此页开始了解下文的内容。

用文字写作的作者不仅可以依赖于读者对单词标准含义的理解（或通过字典等资源查找单词含义），还可以依赖于读者对成千上万的语言符号的起源和相互关系的理解（或通过字典等资源查找）。当读者不理解某个单词的含义时，他自然而然地就可以按照常规去查找此单词，以确定它在文中的含义。

这里需要注意的是，用文字写作的人们不需要了解一些复杂的细节，例如，读者怎样使用字典等等。他们只要正确地拼写单词、并将单词用于人们通常理解的含义便可以了。也就是说，在将“构件（component）”用于多处时只有一个标准的界面。这不仅仅是

件简单定义的事，它意味着作者对读者了解得越多，他的写作就越省力。例如，如果读者熟悉Windows的话，自然明白“double click”这一短语的意思是指双击鼠标左按钮。相比那些不得不从头到尾详细说明的作者来说，了解读者的作者不用花很大的气力，就能在其作品中增加许多内容。

为什么不能赋予软件的编写者们同样的基准呢？软件的编写者应该拥有同样的能力去调用一些预构的单元。他们应该能依赖于一种标准的机制，在需要的时候将这些单元组合地使用。

这就是基于构件的软件技术的目标：让软件的编写者只用几句程序行调调用整个的已知和已测试的构件库。

基于构件应用的开发和利用使得软件用户只需支付一次费用，便能将其中的构件用于多个任务，从而从构件库中多次获益。软件用户可以用钱的形式支付费用，也可以以网络检索时间的形式支付费用。不管是哪种形式，构件都因节约了运行时间而受到欢迎。

构件技术能够降低用于相互通信的软件的成本，潜在地提高软件的质量。它能够分析周围的环境，并控制我们所构建的复杂系统。Sun公司的Java语言有助于构件实现以上的功能，它使基于Java的构件成为受人们欢迎的对象。

■ 构件技术满足了人们长期的渴望

构件的出现为我们提供了一条途径，以解决十年以前一位老资格的软件开发发言人（Frederick Brooks）在一篇论文中提出的问题：“一个软件项目常常会演变成为‘导致计划失算、预算猛增同时产品中又存在缺陷的魔鬼’”。他认为之所以这样是因为在设计或编写附加的程序时造成的软件的内在复杂性。

设想一个有着1000个部件的机器系统，系统中具有许多像紧固件这样的相同零件，它们的设计费用可以分配到上百次的系统使用中。因此，螺栓等机器零件经过设计和测试后可以便宜地复制出许多，并用于不同的地方。

但是，一个具有1000个组成部分的软件系统在设计时要比上述情况复杂得多，费用也高得多。其中的原因就在于这1000个组成部分中没有哪两个是相同的。当软件系统需要在多处使用一个函数时，需要有单独的模块完成从多处调用此函数的任务。

对于软件系统来说，的确有一些通用的函数，例如查找并打开文件等。这些函数中的一部分已经从单独的应用程序中迁移出来，进入Windows等操作系统中。对于制造业等应用领域，有一些通用的概念，例如客户和装运等是可以用在多个项目中的，但操作环境却做不到这一点。包装一些概念以便将它们的代码段用于相似的场合是相当费力的，况且编程人员也没有把所需遵守的规则当作一项约定俗成的传统。

构件技术通过定义框架结构并建立规则，将设计费用分配到了上百次的程序使用中，从而为软件开发者们带来了重用软件的经济效益。一个提供了基础功能的构件，例如格式化的数据项目等可以被开发人员用在公司的不同部门中。如果此构件已经公开销售，那么它还可以用于不同的业界。

第三方的构件开发者具有提供更完整、经过更全面测试、也更易于使用的构件的市场动力，他们提供的构件不同于机构内部开发者们所构建的那些符合某种特定需求的构件。这一市场动力可以实现前面我们提到的所有好处，即第三方开发者可以赚钱；机构内部的开发者可以将有限的资源集中用于开发需要专业化知识、特定任务的应用；而终端用户则可以获得一个更好的解决方案。一项强大的构件技术能够使一个软件开发平台在利润丰厚的软件业中比以前更具竞争性。

■ 开发者需要规则

我们在上文中提到的Frederick Brooks还曾指出：软件的编写是相当困难的。其原因是编程人员需要遵守任意可变的规则 and 标准，软件的某个部分还可能和其它部分相互影响。对其它学科来说，系统在最低一级时变得很相似。例如，一个工程师可以毫无风险地确定，钢条在折断之前曾经发生弯曲，但在软件中却没有诸如此类的内部通用规则。

构件技术规范为软件开发者提供了这样的内部通用规则，从而使软件交互作用的规律性大大增强。构件技术使系统某个部分的更新变得容易起来，并且不会对系统中其它部分产生不应有的边缘效应（side effect）。目前，精确的研究已将原工作码（previously working code）的“断裂（breakage）”确认为是导致软件有缺陷的原因。构件技术有助于解决此问题，并增加了用于提高软件功能和易用性的资源。

■ 使用不是自己亲手构造的构件，安全吗？

除了上述的优点之外，使用由外部中介机构编制的一些软件构件确实也有一些坏处。尤其是在网络环境中，使用不是自己编写的代码段自然而然地会引起人们的怀疑，使用者需要亲自对这段代码一行一行地进行审计，以查找其中的缺陷或错误（出于有意或其它原因的）。

警告：外部编写的构件可能有以下坏处：

- ▶ 破坏或修改数据（例如，将资金从一个帐户转移到另一个帐户）
 - ▶ 破坏专用权机制（例如，读取保险信息）
 - ▶ 拒绝合法用户的业务（例如，将处理器中塞满了高优先权的任务，减慢或停止了正常的操作）
 - ▶ 激怒合法用户（例如，使用客户系统的多媒体硬件发出了烦人的噪声）
-

在网络环境下，软件开发人员可能用一些工具从某些厂商的Web地址等远程地点下载其最新产品。Ken Thompson在一篇有关软件风险的经典论文中指出，如果编译器等关键工具来自外部资源时，利用它建立的应用可能会产生一些不该出现的情况，例如在访问控制例行程序中提供陷阱门（trapdoor）口令等。而且，如果是用编译器的不可信执行版本编译编译器本身时，这些不该出现的情况甚至可能会在那些专心推敲应用或编译器源代码的人还未察觉的情况下发生。