

JBuilder 应用开发丛书



沈耀 郝萍 李扬 等编著

JBuilder

高级开发篇

(JBuilder x / 2005版)

- ✓ JBuilder的UML可视化程序开发
- ✓ 用Jbuilder开发XML应用
- ✓ 用JBuilder开发J2EE应用
- ✓ 用JBuilder开发J2ME应用
- ✓ JBuilder对团队开发的支持



清华大学出版社

JBuilder 应用开发丛书

JBuilder 高级开发篇

(JBuilder X/2005 版)

沈 耀 郝 萍 李 扬 等编著

清华大学出版社

北 京

内 容 简 介

本书为清华大学出版社“JBuilder 应用开发丛书”之第二册。本书按 JBuilder X 与 JBuilder 2005 版编写, 主要内容包括: JBuilder X 的 UML 可视化程序开发; 用 JBuilder X 开发 XML 应用、J2EE 应用、J2ME 应用等; JBuilder 对团队开发的支持等。

本书内容翔实, 讲解全面、清晰、深入、细致。书中提供了详细的代码示例, 所有这些示例在配书光盘中有相应的源代码。

本书要求读者有一定的 Java 语言基础知识, 适用于所有希望掌握 JBuilder X/2005 下开发复杂应用, 如 J2EE 应用、J2ME 应用等的开发人员。本书也可作为利用 Java 开发 XML 应用的参考书。对于希望掌握 JBuilder X/2005 下团队开发知识的软件开发人员, 本书也具有重要的参考价值。

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术, 用户可通过在图案表面涂抹清水, 图案消失, 水干后图案复现; 或将表面膜揭下, 放在白纸上用彩笔涂抹, 图案在白纸上再现的方法识别真伪。

图书在版编目 (CIP) 数据

JBuilder 高级开发篇 (JBuilder X/2005 版) / 沈耀, 郝萍, 李扬等编著. —北京: 清华大学出版社, 2005.4
(JBuilder 应用开发丛书)

ISBN 7-302-10766-1

I. J… II. ①沈… ②郝… ③李… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 028276 号

出 版 者: 清华大学出版社
<http://www.tup.com.cn>
社 总 机: 010-62770175

地 址: 北京清华大学学研大厦
邮 编: 100084
客 户 服 务: 010-62776969

组稿编辑: 欧振旭

文稿编辑: 鲁秀敏

封面设计: 姜凌娜

版式设计: 俞小红

印 刷 者: 北京市清华园胶印厂

装 订 者: 三河市李旗庄少明装订厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 18.25 字数: 419 千字

版 次: 2005 年 4 月第 1 版 2005 年 4 月第 1 次印刷

书 号: ISBN 7-302-10766-1/TP·7167

印 数: 1~5000

定 价: 29.00 元(附光盘 1 张)

丛书简介

关于本丛书

“JBuilder 应用开发丛书”向读者全面、清晰、细致、深入地介绍了 JBuilder 开发工具及在 JBuilder 下开发各类应用程序的基础知识。

需要说明的是，本丛书写作时 JBuilder X 和 JBuilder 2005 均已发布。为了让两个版本的用户均能使用本丛书，编写时我们兼顾了两个版本的共同特点。但应注意，本丛书中涉及的图片是在 JBuilder X 中截取的，所以和 JBuilder 2005 的截图有一定差别，但差别不大，不影响阅读。从书中涉及的示例源程序在 JBuilder X 与 JBuilder 2005 下均能正常运行。

本丛书包括《JBuilder 基础开发篇（JBuilder X/2005 版）》和《JBuilder 高级开发篇（JBuilder X/2005 版）》。前者主要向读者介绍 JBuilder 下 Java 语言的一些基础性开发和相关知识；后者向读者深入介绍 JBuilder 所支持的大型应用的开发和管理，如 UML 图形、J2EE 应用、J2ME 应用及 JBuilder 对团队开发的支持等。

本丛书的定位是使读者能够对 JBuilder 有一个全面而深入的掌握，因而，读者可以在学习的过程中深刻体会到，本丛书对于 JBuilder 的介绍细致入微，因此它适合于任何期望对 JBuilder 有全面了解的读者。

此外，在介绍 JBuilder 的同时，本丛书也深入浅出地介绍了 Java 语言的相关知识和典型平台下应用开发的相关知识，如 Java 中的 Swing、多线程、数据库开发等，以及 J2EE、J2ME 应用的开发等。

本丛书的主要特点如下：

- 细致、全面地介绍了 JBuilder 开发工具；
- 尽可能地采用图形的方式，可视化地向读者介绍；
- 以充足的例子来说明各个知识点；
- 配书光盘提供了书中涉及的完整源代码；
- 在各个相关知识点处列出了主要的技巧、注意点以及说明等。

本丛书的组织结构

第 1 册 《JBuilder 基础开发篇（JBuilder X/2005 版）》

第 1 章 初识 JBuilder X

本章向读者展示了 JBuilder 的版本演化及 JBuilder X 的安装，并演示了如何创建第 1 个 Java 应用程序。

第 2 章 Java 提要

本章立足于 Java 语言, 向读者介绍了 Java 语言的相关知识, 包括类和对象、数据类型和表达式、操作和控制流、异常处理、输入/输出以及注释等。

第 3 章 JBuilder X 项目开发

本章主要介绍 JBuilder X 的集成开发环境、JBuilder X 的项目管理以及应用程序的运行和调试。

第 4 章 基础主题式开发

本章介绍了 JBuilder X 下的 Applet 开发、基于 Swing 的应用程序的开发和 Java 的网络编程。

第 5 章 开发数据库应用程序

本章针对 Java 中的数据库应用开发, 详细介绍了 JBuilder X 下如何开发数据库应用程序, 包括数据库应用的基础知识、安装 JDBC 和配置数据库、JBuilder 中的数据库管理工具、数据库查询与操作、数据模块的使用以及创建分布式数据库应用等。

第 6 章 Swing 的高级应用

本章详细介绍了 Swing 中相关组件的使用。

第 7 章 编写多线程应用

本章详细介绍了 Java 应用中的多线程编程, 同时介绍了多线程间的同步、通信, 以及对多线程应用的调试等。

第 8 章 JavaBean 和 BeansExpress

本章详细介绍了 JavaBean 的相关知识及如何运用 BeansExpress 创建 JavaBean。

第 2 册 《JBuilder 高级开发篇 (JBuilder X/2005 版)》

第 1 章 UML 可视化程序开发

本章向读者介绍了 UML 的基础知识, 并详细介绍了 JBuilder X 对 UML 的支持。

第 2 章 开发 XML 应用

本章详细介绍了 XML 的相关知识, 并介绍了 JBuilder 下 Java 语言如何操作 XML。

第 3 章 用 JBuilder 开发 J2EE 应用

本章详细介绍了 J2EE 的相关知识及如何利用 JBuilder X 开发相关应用, 包括 Web 应用, EJB 中的会话 Bean、实体 Bean 和消息驱动 Bean, Web 服务等。此外, 本章还介绍了 EJB 开发中的设计模式及如何调试企业级的应用程序。最后介绍了 CORBA 和 Java 对 CORBA 的支持, 并以实例说明了如何在 JBuilder X 中构建 CORBA 的服务器和客户端。

第 4 章 用 JBuilder 开发 J2ME 应用

本章介绍了在 JBuilder X 下如何配置和开发 J2ME 的应用, 包括 JBuilder X 下移动应用的开发环境、开发 MIDP 项目、部署移动应用等。

第 5 章 JBuilder 对团队开发的支持

本章介绍了 JBuilder X 对团队开发的支持, 并详细介绍了 StarTeam 和 CVS 系统。

丛书读者对象

本书的目的在于向读者全面、清晰、细致、深入地介绍 JBuilder 开发工具及在 JBuilder 下开发各类应用程序的基础知识, 希望读者在完成本书的阅读后, 能够对 JBuilder 有一个全面深入的掌握, 并能够熟练运用 JBuilder 的各项功能, 进行各类应用的开发。

本书涉及 JBuilder 的基础开发和高级开发两部分, 将它们在两本书, 即《JBuilder 基础开发篇 (JBuilder X/2005 版)》和《JBuilder 高级开发篇 (JBuilder X/2005 版)》中分别介绍。

对于初涉 Java 的程序员, 本书是一套极好的入门教材。这是因为, 丛书的《JBuilder 基础开发篇 (JBuilder X/2005 版)》向读者详细地介绍了 Java 语言的基础知识和常规应用的开发, 同时还详尽地介绍了如何运用 JBuilder X; 丛书的《JBuilder 高级开发篇 (JBuilder X/2005 版)》则从 Java 的大型应用方面介绍了 UML、XML 以及 J2EE、J2ME 和团队开发等。如此, 对于 Java 新手, 不仅能够学到 Java 的基本知识和 JBuilder X 的使用方法, 而且也能够掌握 Java 平台下的各类大型应用的开发。

对于 Java 有基本知识的读者, 本书能够帮助他们了解如何利用 JBuilder 开发各类 Java 应用。同时, 《JBuilder 高级开发篇 (JBuilder X/2005 版)》可以帮助读者掌握企业大型应用的开发。

对于长期使用 Java 语言的读者, 本书无疑可以帮助他们全面深入地了解 JBuilder 提供的智能化的、友好的开发方式。

尽管对使用本书的读者没有任何特殊的要求, 但是具有一门或多门程序设计语言的读者将更容易掌握本书的内容。

作者简介

本丛书由沈耀和郝萍组织编写，并由沈耀和李扬主笔，其他参与编写的人员还有陈琳、朱乐骏、杨建良、周颢、刘新华、崔卓等。在此，感谢他们的辛勤工作。以下是各位作者的简介。

沈耀 毕业于上海交通大学计算机软件专业，工学硕士。现就读于上海交通大学计算机科学与工程系，攻读博士学位。多年从事分布计算的研究，对 J2EE、CORBA、Web 服务及网格计算等有较为深入的探讨。

郝萍 毕业于上海交通大学计算机软件专业，工学硕士。现任职于 IBM 中国软件开发中心，从事计算机软件开发工作。具有多年 WebService 和电子商务方面的研究及 Java 项目经验。

李扬 毕业于上海交通大学计算机应用专业，工学硕士。现任职于 IBM 中国软件开发实验室（上海）。长期专业于 Java 软件设计开发，覆盖 J2SE、J2EE 和 J2ME。工作期间参与的项目多为 J2EE。业余时间喜欢研究各种 Java open source。

陈琳 在重庆邮电学院分别获得计算机学士学位和硕士学位。毕业后曾在中兴通信股份有限公司 CDMA 事业部就职，主要从事 J2EE 架构下 CDMA 设备电信管理系统的研发工作。现就读于上海交通大学，攻读博士学位，从事网格计算、Web Service 方面的研究。感兴趣的研究领域包括网格计算、BPEL、Web Service、CORBA 等。

朱乐骏 上海交通大学计算机科学与工程系硕士毕业。自 1998 年起开始研究 Java 语言，擅长 J2SE 和 J2EE 架构下的各种编程技术，特别是基于网络的分布式应用和企业级多层结构。另外，还对 XML 标记语言、Web 服务和语义 Web 有深入研究，并著有数篇学术论文。

杨建良 计算机硕士，对 Java、数据库和 J2EE 有浓厚兴趣，并具有 5 年以上的 JBuilder 开发经验。现从事计算机信息安全方面的工作，主要研究方向为数据库安全。

周颢 获得上海交通大学计算机系硕士学位。现任职于 Intel 中国软件拓展中心，主要的工作内容是设计和开发英特尔平台相关的解决方案，以及提供基于英特尔平台的软件性能优化和功能扩展。

刘新华 毕业于上海交通大学计算机系。有多年的系统集成与产品开发经验。现为 HP 中国软件研发中心的高级工程师。

崔卓 在吉林大学分别获得计算机科学学士学位和硕士学位。现任职于中兴通信股份有限公司 CDMA 事业部支撑软件部，主要从事嵌入式操作系统、单板软件支撑包的研发工作。对 Java 有浓厚兴趣，很早就开始使用 Java 开发各类应用程序。

目 录

第 1 章 UML 可视化程序开发.....	1
1.1 可视化建模与 UML 简介.....	2
1.1.1 可视化建模.....	2
1.1.2 UML 概述.....	3
1.1.3 UML 的结构.....	3
1.1.4 UML 构造块.....	4
1.1.5 UML 规则.....	8
1.1.6 UML 的公共机制.....	8
1.2 JBuilder X 对 UML 的支持.....	9
1.2.1 Java 和 UML.....	9
1.2.2 JBuilder 和 UML.....	10
1.2.3 UML 图的个性化设置.....	13
1.3 查看 UML 视图.....	18
1.3.1 UML 浏览器.....	18
1.3.2 UML 图形导航.....	22
1.4 小结.....	23
第 2 章 开发 XML 应用.....	25
2.1 XML 语法和规范.....	26
2.1.1 XML 概述.....	26
2.1.2 XML 相关标准.....	30
2.1.3 XML 编程模型.....	34
2.2 创建 XML 以及相关文件.....	36
2.2.1 手工生成 XML 及相关文件.....	36
2.2.2 用 JBuilder 查看、编辑 XML 文档.....	37
2.2.3 使用 Wizard 由 DTD 生成 XML 文件.....	39
2.2.4 使用 Wizard 由 XML 生成 DTD 文件.....	40
2.3 XML 相关编程.....	42
2.3.1 SAX Handler 的使用.....	42
2.3.2 Databinding 的使用.....	48
2.4 XML 数据库组件.....	53
2.4.1 XMLDBMSTable 和 XMLDBMSQuery.....	54
2.4.2 XTable 和 XQuery.....	63
2.5 小结.....	68

第 3 章 用 JBuilder 开发 J2EE 应用	71
3.1 J2EE 概述	72
3.2 与应用服务器集成开发 Web 应用	73
3.2.1 什么是 Web 应用	74
3.2.2 Servlet 介绍	74
3.2.3 使用 JBuilder 编写 Servlet	76
3.2.4 打包 Web 应用	81
3.2.5 与 Tomcat 集成开发	85
3.2.6 使用框架	87
3.3 EJB 概述	98
3.3.1 什么是 EJB	98
3.3.2 为什么要使用 EJB	99
3.3.3 Enterprise Bean 的分类	100
3.3.4 C/S 体系结构和多层体系结构	101
3.3.5 EJB 的运行环境	102
3.3.6 EJB 是怎样工作的	103
3.4 开发 Session Bean	104
3.4.1 Session Bean 详细描述	104
3.4.2 用 JBuilder X 编写 EJB	105
3.4.3 用 JBuilder X 编写客户端	113
3.4.4 在 Web 应用中使用 Session Bean	124
3.4.5 总结: Session Bean 的生存周期	125
3.5 开发 Entity Bean	126
3.5.1 两种不同的 Entity Bean	126
3.5.2 用 JBuilder X 编写 CMP Bean	127
3.5.3 用 JBuilder X 编写 BMP Bean	134
3.5.4 CMP Bean 的配置文件	135
3.5.5 安全	137
3.5.6 总结: Entity Bean 的生存周期	140
3.6 开发 MDB	141
3.6.1 消息服务	141
3.6.2 MDB 详细描述	141
3.6.3 用 JBuilder X 编写 MDB	142
3.6.4 MDB 的设计使用	143
3.6.5 总结: MDB 的生存周期	144
3.7 EJB 开发中的设计模式	144
3.7.1 Session Facade	145
3.7.2 Message Facade	145

3.7.3	EJB Command	146
3.7.4	Business Interface	147
3.7.5	Data Transfer Object	148
3.8	Web Service	149
3.8.1	为什么需要 Web Service	149
3.8.2	SOAP	150
3.8.3	WSDL	151
3.8.4	注册 Web Service	152
3.8.5	JBuilder 对 Web Service 的支持	152
3.8.6	Web Service 调试	157
3.8.7	总结	159
3.9	调试企业应用	160
3.9.1	远程调试介绍	160
3.9.2	构建调试环境	161
3.9.3	调试一个远程应用	162
3.9.4	总结	165
3.10	CORBA	165
3.10.1	CORBA 基础	165
3.10.2	Java 对 CORBA 的支持	166
3.10.3	设置 JBuilder 使用 CORBA	166
3.10.4	构建一个 CORBA 服务器	167
3.10.5	构建一个 CORBA 客户端	170
3.11	小结	171
第 4 章	用 JBuilder 开发 J2ME 应用	173
4.1	JBuilder X 的移动应用开发环境	174
4.1.1	设置 JDK	174
4.1.2	设计 MIDP 用户界面	179
4.1.3	使用屏幕管理器	180
4.1.4	对 MIDlet 类文件使用混淆器	182
4.1.5	构建掌上平台应用	185
4.2	创建和管理 MIDP 项目	187
4.2.1	新建一个 MIDP 项目	187
4.2.2	添加 MIDlet 文件	190
4.2.3	设置项目属性	191
4.3	MIDP 应用	193
4.3.1	编译 MIDP 应用	193
4.3.2	运行 MIDP 应用	194
4.3.3	调试 MIDP 应用	198

4.4	建立 MIDP 用户界面.....	198
4.4.1	MIDP 用户界面.....	199
4.4.2	MIDP 用户界面组件.....	200
4.4.3	创建 MIDP 用户界面.....	202
4.4.4	用户界面设计器.....	204
4.4.5	在设计器中设置属性.....	208
4.4.6	控制用户界面布局.....	209
4.4.7	使用画布 (Canvas)	210
4.4.8	MIDP 消息.....	212
4.4.9	在 MIDlet 中使用图像.....	213
4.4.10	添加 MIDP 组件到工具栏.....	214
4.5	MIDP 数据库编程.....	215
4.6	打包 MIDlet.....	216
4.6.1	从命令行生成文档.....	216
4.6.2	使用文档生成器.....	220
4.7	部署移动应用.....	231
4.7.1	MIDlet Suite 的生命周期.....	232
4.7.2	配置服务器识别 JAD 和 JAR 文件	232
4.7.3	使用 OTA Provisioning	234
4.8	小结	239
第 5 章	JBuilder 对团队开发的支持.....	241
5.1	软件项目的团队开发.....	242
5.2	团队的结构和职责.....	244
5.3	StarTeam.....	246
5.3.1	StarTeam 服务器权限设置.....	246
5.3.2	StarTeam 客户端设置.....	254
5.3.3	StarTeam 项目管理	256
5.3.4	StarTeam 视图管理.....	256
5.3.5	StarTeam 分支管理.....	256
5.3.6	StarTeam 标签管理.....	258
5.3.7	StarTeam 目录划分管理	259
5.4	CVS.....	261
5.4.1	CVS 服务器设置.....	261
5.4.2	CVS 客户端设置.....	265
5.4.3	CVS 环境初始化.....	266
5.4.4	CVS 的日常使用	267
5.4.5	CVS Branch: 项目多分支同步开发	269

5.4.6 通过 SSH 远程访问 CVS.....	270
5.4.7 CVSWEB: 提高文件浏览效率.....	271
5.5 小结	272
参考文献.....	274

第 1 章



DESIGN

UML 可视化程序开发

本章主要内容包括：

- ▣ 可视化建模与 UML 简介
- ▣ JBuilder X 对 UML 的支持
- ▣ 利用 UML 浏览器查看 UML 视图

面向对象的分析与设计 (OOA&D) 方法的发展在 20 世纪 80 年代末至 20 世纪 90 年代中期出现了一个高潮, UML 是这个高潮的产物。它不仅统一了 Booch、Rumbaugh 和 Jacobson 的表示方法, 而且对其作了进一步的发展, 并最终统一为大众所接受的标准建模语言。

Java 是一种纯面向对象的编程语言, 对于基于 Java 语言的应用程序来说, 利用 UML 进行的可视化分析与设计, 能够无缝地过渡到程序的开发过程。另一方面, 基于 Java 语言的代码, 也能够轻易地通过逆向工程, 生成 UML 的可视化图形。

JBuilder 提供了通过 Java 代码生成 UML 类图与包的依赖关系图的功能, 这一功能有助于应用程序的开发人员能够对程序更好地设计、理解以及解决开发过程中的一系列问题。

1.1 可视化建模与 UML 简介

建模是软件分析和设计中一个必不可少的阶段。UML 是一种标准的可视化建模语言。本节将简要介绍可视化建模以及 UML 的相关知识。

1.1.1 可视化建模

我们所做的事情, 很多都需要首先制定计划并建立相应的模型, 例如建造大楼, 在动工之前需要制定精确的图纸, 建立详细的大楼模型。对于开发应用程序来说, 也需要建立相应的模型。建立模型的过程简称建模。

建模对于软件项目来说非常重要。不成功的软件项目失败的原因可能各不相同。然而所有成功的软件项目都有一个共同点, 那就是采用了建模技术。

模型是对现实的简化, 它提供系统的蓝图。模型既可以包括详细的计划, 也可以包括从高层次给出的对系统的抽象。一个好的模型必然包括那些对后续构建系统起重大作用的必需元素, 而忽略那些与模型抽象层次不相关的次要元素。每个系统都可以从不同的角度对现实进行抽象, 从而建立不同的模型。模型可以是结构性的, 强调系统的组织特性, 也可以是行为性的, 强调系统的动态特性。

为什么要建模呢? 因为模型可以帮助人们了解当前的现实情况, 了解当前所拥有的系统, 同时允许人们对所要构建的系统作出详细的说明, 从而指导人们构建未来的系统, 而且模型对人们作出的决策进行了文档化。

可视化建模是以图形的方式描述所开发系统的过程。尽管可视化建模以图形描述为主, 但事实上, 任何一个复杂系统, 简单的图形描述是不够的。这主要是因为: 图形描述尽管形象直观, 但是有些情况下图形说明不够精确; 而且, 不是系统的所有方面都能通过图形进行建模的。因而, 一般情况下, 可视化建模同时需要文字进行说明。

1.1.2 UML 概述

UML (Unified Modeling Language, 统一建模语言) 是用于对系统进行可视化建模的语言。

UML 建模具有两个方面, 如下:


- 静态结构——描述什么类型的对象对于建模系统是重要的, 以及它们之间是什么关系;
- 动态行为——描述了这些对象的生命周期以及它们是如何协作以提供系统所需的功能。

UML 模型的这两个方面关系紧密, 是相互对应的。

UML 的目标是以面向对象的图形方式来描述任何类型的系统, 具有很宽的应用领域。其中最常用的是建立软件系统的模型。但 UML 同样可以用于描述非软件领域的系统, 如机械系统、企业机构或业务过程, 以及处理复杂数据的信息系统、具有实时要求的工业系统或工业过程等。总的来说, UML 是一个通用的标准建模语言, 可以对任何具有静态结构和动态行为的系统进行建模。

同时, UML 也适用于系统开发过程中从需求规格描述到系统完成后测试的不同阶段。在需求分析阶段, 可以用用例图来分析用户需求; 分析阶段侧重于关心问题域中的主要概念, 识别出分析类以及它们相互间的关系, 并用 UML 类图来描述。为实现用例, 类之间需要协作, 这可以用 UML 动态模型来描述。在设计阶段, 进一步利用 UML 的静态和动态建模技术对分析模型进行细化。在构造阶段, 其任务是用面向对象编程语言将来自设计阶段的类转换成实际的代码。UML 模型还可作为测试阶段的依据。系统通常需要经过单元测试、集成测试、系统测试和验收测试。不同的测试小组使用不同的 UML 图形作为测试依据: 单元测试使用类图和类规格说明; 集成测试使用部件图和合作图; 系统测试使用用例图来验证系统的行为; 验收测试由用户进行, 以验证系统测试的结果是否满足在分析阶段确定的需求。

总之, 标准建模语言 UML 适用于以面向对象技术来描述任何类型的系统, 而且适用于系统开发的不同阶段, 从需求规格描述直至系统完成后的测试和维护。

 **注意:** UML 是一门标准化建模语言, 它有别于我们的日常语言, 因为它比日常语言更能精确、可视化地描述系统; 它也有别于程序语言, 因为它的目的在于系统的可视化建模, 而不是系统的运行。

1.1.3 UML 的结构

UML 是一种可视化建模语言, 它有自身的内部结构, 很多书上也称为 UML 的概念模型。UML 的结构包括:

- UML 构造块——UML 用于构造模型而采用的基本元素。

- 规则——规定各种 UML 的构造块如何组织在一起的细则。
- 公共机制——可以运用于整个 UML，以达到特定目标的公共 UML 方法。

UML 的结构是 UML 对任何系统进行建模的基础。理解了 UML 的结构，就能够读懂 UML 模型，并可以建立一些基本模型。在此基础上，如果掌握了更多的建模技术，就可以使用更高深的语言特征对复杂系统进行建模。

1.1.4 UML 构造块

UML 构造块包括如下 3 类：

- 事物——对模型中主要建模元素的抽象。
- 关系——把事物组合在一起，说明两个或多个事物是如何语义相关的。
- 图——聚集了相关的事物以及事物之间的关系，组成 UML 模型的视图。

以下将分别对 UML 构造块进行说明。

1. UML 中的事物

在 UML 中包括 4 类事物，即结构事物、行为事物、分组事物和注释事物。

结构事物表征 UML 模型中的名词，它们通常表示模型中的静态部分，描述系统的物理元素或相关概念。共有 7 种结构事物。

(1) 类 (class)

类是对具有相同属性和相同操作对象的抽象。类可以用矩形来表示，在矩形中可以指定类的名称、属性和操作等，如图 1-1 所示。

(2) 接口 (interface)

接口表示一组操作的集合，操作集合中的操作在系统中往往具有较为密切的关系，如完成某一特定的功能的多个操作或实现相似功能的多个操作等。接口并不实现操作，而仅仅描述类或构件的外部可见行为。在图形上，接口通常用圆来表示，并在圆的下方注明接口的名称，如图 1-2 所示。

(3) 协作 (collaboration)

协作定义了交互的操作，是一些角色和其他元素一起工作，提供一些合作的动作，这些动作比元素的总和要大。因此，协作具有结构化、动作化、维的特性。一个给定的类可能是几个协作的组成部分。这些协作代表构成系统的模式的实现。协作在 UML 中用一个虚线画的椭圆和它的名字来表示，如图 1-3 所示。

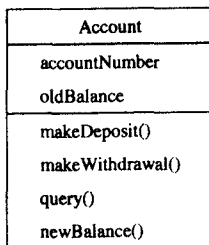


图 1-1 类



图 1-2 接口



图 1-3 协作

(4) 用例 (use case)

用例描述了一系列的动作，这些动作由系统针对特定的参与者执行，并产生参与者可观察的结果。在模型中用例通常用来组织动作事物。用例是通过协作来实现的。在 UML 中，用例通过实线椭圆和它的名字来表示，如图 1-4 所示。

(5) 主动类 (active class)

主动类的对象有一个或多个进程或线程，它和类很相像，只是它的对象代表的元素的行为和其他元素的行为可以并发。在 UML 中，主动类的画法和类相同，只是边框用粗线条，如图 1-5 所示。

(6) 组件 (component)

组件是物理的、可替换的系统部分，它实现了一个接口集合。在一个系统中，可能会遇到不同种类的组件，例如 COM+ 或 ENTERPRISE JAVA BEANS。组件在 UML 中用图 1-6 表示。

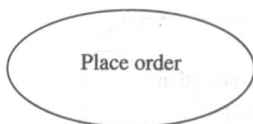


图 1-4 用例

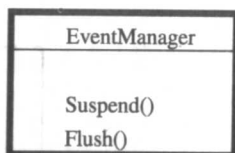


图 1-5 主动类

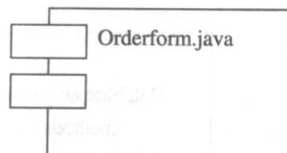


图 1-6 组件

(7) 结点 (node)

结点是一个物理元素，它在运行时存在，代表一个可计算的资源，通常占用一些内存并具有处理能力。一个组件集合一般来说位于一个结点，但有可能从一个结点转到另一个结点。结点通常用图 1-7 表示。

类、接口、协作、用例、主动类、组件和结点这 7 个元素是在 UML 模型中使用的最基本的结构事物。系统中还有这 7 种基本元素的变化体，如角色、信号（某种类），进程和线程（某种活动类），应用程序、文档、文件、库、表（组件的一种）。

行为事物是 UML 模型中的动态部分。它们是模型的动词，代表时间和空间上的动作，共有两种主要的行为事物。

(1) 交互 (interaction)

交互是由一组对象在特定上下文中，为达到特定的目的而进行的一系列消息交换而组成的动作。在交互中组成动作的对象的每个操作都要详细列出，包括消息、动作次序（消息产生的动作）、连接（对象之间的连接）。在 UML 中消息画成带箭头的直线，通常加上操作的名字，如图 1-8 所示。

(2) 状态机 (state machine)

状态机由一系列对象的状态组成，在 UML 中状态表示为图 1-9。

交互和状态机是 UML 模型中最基本的两个动态事物元素，它们通常和其他结构元素、主要的类、对象连接在一起。



说明：UML 的结构事物善于描述系统的静态部分，而行为事物用于描述系统的动态部