

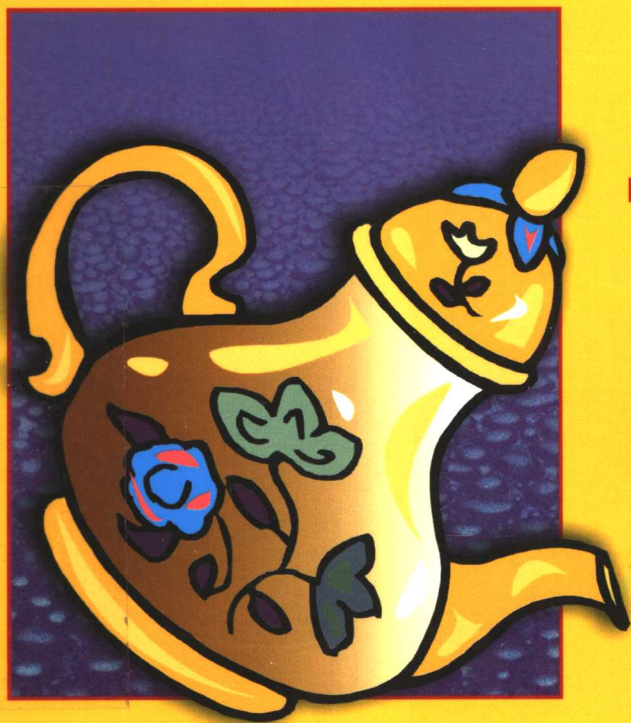
# NetBeans IDE 5.5

## 专家编程

### Pro NetBeans IDE 5.5 Enterprise Edition

(美) Adam Myatt 著

蔡黄辉 译



- 利用开源的、轻量级的NetBeans IDE平台构建您的Java EE 5应用程序



机械工业出版社  
China Machine Press

TP312/2632

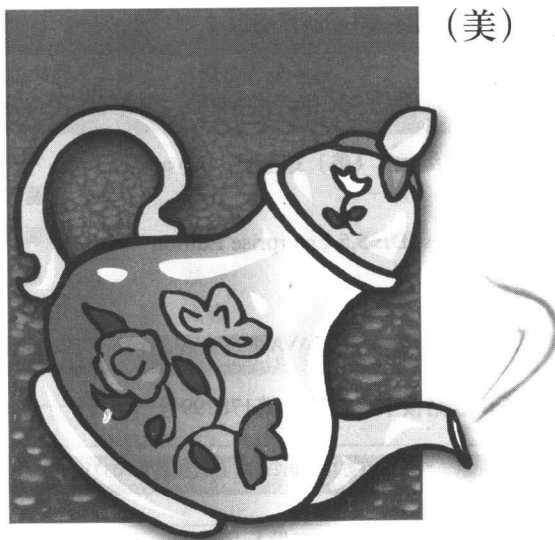
2008

# NetBeans IDE 5.5

## 专家编程

### Pro NetBeans IDE 5.5 Enterprise Edition

(美) Adam Myatt 著  
蔡黄辉 译



机械工业出版社  
China Machine Press

本书为NetBeans IDE和企业附加软件的功能提供了全面的指南。本书涉及许多主题，例如：Web应用程序、Web服务和UML建模，还有实际生活案例分析和清楚的专题信息。这些内容结合起来，为开始使用NetBeans IDE 5.5企业版平台提供了所需的全部资料。

无论您是一个经验丰富的IDE用户还是一个只使用Emacs的Java开发人员，都可以从这本书中找到需要的内容。

Adam Myatt: Pro NetBeans IDE 5.5 Enterprise Edition (ISBN 1-59059-788-5) .

Original English language edition published by Manning Publications Co., 209 Bruce Park Avenue, Greenwich, Connecticut 06830.

Copyright © 2007 by Adam Myatt.

All rights reserved.

Simplified Chinese language edition published by China Machine Press.

Copyright © 2007 by China Machine Press.

本书中文简体字版由美国Manning Publications Co.授权机械工业出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2007-3361

## 图书在版编目（CIP）数据

NetBeans IDE 5.5专家编程 / (美) 迈亚特 (Myatt, A.) 著；蔡黄辉译. —北京：机械工业出版社，2008.1

书名原文：Pro NetBeans IDE 5.5 Enterprise Edition

ISBN 978-7-111-22828-8

I. N… II. ① 迈… ② 蔡… III. JAVA语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字（2007）第178799号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码100037）

责任编辑：杨庆燕

北京京北制版厂印刷·新华书店北京发行所发行

2008年1月第1版第1次印刷

186mm×240mm · 19.25印张

定价：42.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换  
本社购书热线：(010) 68326294

## 译者序

从事软件开发好几年，接触了不少Java IDE，最初了解NetBeans是从Creator开始的。那时候使用Creator开发portlet，后来知道NetBeans集成了Visual Web Pack，也可以进行可视的Web应用程序开发，而且利用NetBeans还可以进行许多其他项目的开发，所以后来转向使用NetBeans。

NetBeans 5.5提供了许多的项目类型可以选择，而且还提供了基于现有源代码和Ant文件创建项目，这样可以非常方便地把其他项目直接导入到NetBeans中。

NetBeans还为开发基于Struts框架的Web应用程序提供了许多向导，这样，要创建Struts Form和Action就非常容易，不用修改struts的配置文件。

NetBeans对JUnit的支持也非常好。在NetBeans中可以很容易地编写单元测试和执行测试。

NetBeans默认提供了对CVS版本控制的支持，通过更新中心还可以提供对Subversion的支持，这对于使用NetBeans的开发人员来说也很方便。

这些都是我在实际的项目开发中经常使用的功能。接下来我将在项目中尝试使用NetBeans提供的Web服务功能，这个看起来好像也不错。

令我震撼的是NetBeans的协作（Collaboration）工具（本书的第17章讲到的），通过它，分布在不同区域的开发团队可以几乎实时地同时编辑同一个文件，而且还可以提交和生成项目。这样就解决了分布在各处的开发团队修改相同代码块的问题。这肯定很有趣，如果有机会，我很想尝试这种开发模式。

书中还讲到了NetBeans许多其他的功能，比如操作数据库、创建UML项目、代码缩写模板等，这些功能都很强大，对提高开发效率很有帮助。

总的说来，NetBeans 5.5提供了许多功能，本书对这些功能进行了比较详细的阐述，阅读本书，可以加快和加深对NetBeans的了解。

由于时间仓促，加上译者水平有限，书中难免有翻译错漏或不妥之处，还请广大读者和同行不吝指正！

最后，感谢我的妻子沈晓霞，感谢她在我翻译本书的过程中给予我的支持，她还为本书的校验提供了很多帮助。

# 前 言

这本书探讨了NetBeans IDE 5.5的很多特性。在学习NetBeans时，开发者可以集中在多种技术和领域。在最新版本中，用户可以使用许多特性，比如JavaServer Faces (JSF)、Ajax、统一建模语言 (Unified Modeling Language, UML)、Enterprise JavaBeans (EJB 3.0)、Java持久层 (Persistence) API、Web服务、业务流程执行语言 (Business Process Execution Language, BPEL) 等。我要写一本书，在书中陈列出使用这些技术的很好的工具。

Pro NetBeans IDE 5.5企业版是为所有层次的开发者设计的。无论是刚接触NetBeans、还是学生程序员，或者是经验丰富的专业人士，都会发现这本书提供了特性的直接解释和简单易懂的样例。这本书也集中在NetBeans的一些帮助专业软件开发者的核心特性上：Ant、JUnit、并行版本系统 (Concurrent Versioning System, CVS)、Subversion、开发者协作工具、代码覆盖等。

我个人的网站[www.ProNetBeans.com](http://www.ProNetBeans.com)上有各种内容，比如Java和NetBeans新闻、文章、指南及其他。网站上也有本书的更新、修正和勘误表。如果有任何问题或者想要提供反馈，随时可以联系我，我的邮箱是[adam@pronetbeans.com](mailto:adam@pronetbeans.com)。

## 关于作者

编写本书时ADAM MYATT是GE全球软件发展研究机构 (Software Development for GE Global Research) 的主要技术专家，这是通用电气的全球研发中心总部，位于纽约Niskayuna。Adam是一位经验丰富的Java开发人员，拥有Sun Microsystems认证的Java程序员证书。他的工作是引导全世界的开发人员按照一个严格的软件开发生命周期过程开发Java软件和Web应用程序，他还研究新的技术并设置长期策略。

Adam是当地Java用户组的一个积极参与者和开源软件的狂热者。在加入GE前，他也为几个软件公司工作过。Adam毕业于波茨坦的纽约大学的州立大学计算机科学系。

在他很少的业余时间里，Adam喜欢到新的和有趣的地方旅游、钓鱼和打牌。他也喜欢在家里工作并不停地谈论工作，这一点几乎让他的妻子发狂。

## 关于技术评议专家

SUMIT PAL有大约12年的软件架构设计和在多种平台上开发的经验，其中包括J2EE。Sumit在微软公司的SQL Server Replication组工作了两年，在Oracle公司的Oracle OLAP Server组工作了七年。

Sumit既拥有IEEE-CSDP和J2EE架构的认证，也从泰国的亚洲技术学院得到了计算机科学的MS。

Sumit对数据库内核、算法和搜索引擎技术非常感兴趣。他已经发明了一些用于查找数字之间整除性的基本广义算法，也发明了小于100的素数的整除性规则。

现在，Sumit喜欢尽量多地和他女儿一起玩。他也喜欢游泳和打羽毛球。

## 致谢

我要感谢许多人，没有他们就不可能有这本书。在写这本书的时候，一个非常热诚的团队极大地帮助了我。

首先，我要感谢我的编辑Steve Anglin，为了这个项目，他把我和Apress聚到了一起并给了我这个机会。在这个项目中，他的建议和指导的价值是无法估量的。

我要对我的项目经理Beth Christmas表达真诚的感激之情，因为工作非常困难，让我几乎不能保持进度。她的经验和专业知识帮助了我，使得写书的过程比原来意料的容易了很多。

感谢我的技术评议专家Sumit Pal，他使得这本书更丰富、更精确。我要感谢他所有包含深刻见解的建议和评论。

我也非常感谢版面文字和制作编辑Marilyn Smith和Katie Stence，他们优秀的工作使得我写的内容更便于阅读而且看起来更优雅。不知道为什么，Marilyn总能搞清比较杂乱的原因并帮助我整理思路，使得我能写出这本极好的书。

我非常感谢Apress团队中所有成员的努力。

我也要感谢整个GEGR ITMS机构对我工作的支持。有时候平衡工作和个人计划很困难，但是和一个伟大的团队一起工作比较容易做到这一点。

在我的学习和职业生涯中还有很多人和团体以各种方式指导我。感谢John Dalphin博士、所有过去和现在在SUNY Potsdam CIS的教员、CIS顾问团、我原来所在IT部门的Chris Thompson和Anthony Debonis、CDJDN。

最后，我要对我的妻子Morgan表达我衷心的感谢，她长时间的忍受我晚上和周末都要写这本书。在这个项目过程中，有了她的爱和支持，才使得这一切得以实现。

# 目 录

译者序  
前言

## 第一部分 创建和管理项目

|                                 |    |
|---------------------------------|----|
| 第1章 集成开发环境 .....                | 1  |
| 1.1 为什么使用IDE .....              | 2  |
| 1.2 NetBeans与其他IDE工具的比较 .....   | 2  |
| 1.3 小结 .....                    | 4  |
| 第2章 安装和定制NetBeans .....         | 5  |
| 2.1 下载文件 .....                  | 5  |
| 2.2 安装NetBeans IDE和核心附加软件 ..... | 5  |
| 2.3 定制NetBeans JVM 启动选项 .....   | 6  |
| 2.4 设置代理 .....                  | 6  |
| 2.5 检查更新 .....                  | 7  |
| 2.5.1 使用更新中心 .....              | 7  |
| 2.5.2 安装先下载下载的模块 .....          | 10 |
| 2.6 通过模块管理器激活特性 .....           | 10 |
| 2.7 定制IDE .....                 | 11 |
| 2.7.1 设置内部Web浏览器 .....          | 12 |
| 2.7.2 设置代码编辑器缩排 .....           | 12 |
| 2.7.3 选择字体和颜色 .....             | 15 |
| 2.7.4 设置快捷键映射 .....             | 16 |
| 2.7.5 设置高级选项 .....              | 17 |
| 2.8 操作NetBeans .....            | 18 |
| 2.8.1 “项目”窗口 .....              | 18 |
| 2.8.2 “文件”窗口 .....              | 18 |
| 2.8.3 “运行环境”窗口 .....            | 18 |
| 2.8.4 “导航”窗口 .....              | 18 |
| 2.8.5 “源编辑器” .....              | 18 |
| 2.8.6 “输出”窗口 .....              | 18 |
| 2.8.7 “属性”窗口 .....              | 19 |
| 2.8.8 “组件面板”窗口 .....            | 19 |

|                                |    |
|--------------------------------|----|
| 2.9 小结 .....                   | 19 |
| 第3章 创建一个Java项目 .....           | 20 |
| 3.1 创建一个Java应用程序 .....         | 20 |
| 3.2 使用“项目”窗口 .....             | 21 |
| 3.2.1 源包 .....                 | 21 |
| 3.2.2 测试包 .....                | 22 |
| 3.2.3 库 .....                  | 22 |
| 3.2.4 测试库 .....                | 22 |
| 3.3 使用源编辑器 .....               | 23 |
| 3.3.1 显示文件 .....               | 23 |
| 3.3.2 行号 .....                 | 23 |
| 3.3.3 代码折叠 .....               | 24 |
| 3.3.4 当前行突出显示 .....            | 24 |
| 3.3.5 代码语法错误突出显示 .....         | 24 |
| 3.3.6 注释字形和错误条纹 .....          | 25 |
| 3.3.7 在源编辑器中排列文件 .....         | 25 |
| 3.4 使用“文件”窗口 .....             | 26 |
| 3.5 引入项目和代码 .....              | 27 |
| 3.5.1 从现有的源中创建一个Java项目 .....   | 27 |
| 3.5.2 从Eclipse中引入一个现有的项目 ..... | 27 |
| 3.5.3 从JBuilder中导入现有的项目 .....  | 29 |
| 3.6 构建HelloWorld.java .....    | 29 |
| 3.6.1 在项目中添加Java文件 .....       | 29 |
| 3.6.2 编译并运行项目 .....            | 30 |
| 3.7 小结 .....                   | 31 |
| 第4章 创建一个Web应用程序项目 .....        | 32 |
| 4.1 创建一个Web应用程序 .....          | 32 |
| 4.2 操作这个Web应用程序项目 .....        | 34 |
| 4.2.1 Web页 .....               | 34 |
| 4.2.2 配置文件 .....               | 35 |
| 4.2.3 服务器资源 .....              | 35 |
| 4.3 添加Web应用程序文件 .....          | 35 |
| 4.3.1 添加HTML文件 .....           | 35 |



|                                               |    |                                   |     |
|-----------------------------------------------|----|-----------------------------------|-----|
| 4.3.2 添加CSS文件 .....                           | 37 | 5.3.1 定制主题 .....                  | 72  |
| 4.3.3 添加部署描述符 .....                           | 38 | 5.3.2 导入修改的主题 .....               | 76  |
| 4.3.4 添加JSP文件 .....                           | 40 | 5.4 使用“页面导航”工具 .....              | 76  |
| 4.4 Struts的优势 .....                           | 42 | 5.4.1 定义导航规则 .....                | 77  |
| 4.4.1 添加Struts支持 .....                        | 43 | 5.4.2 使用“页面导航”窗口 .....            | 77  |
| 4.4.2 添加表单 .....                              | 44 | 5.5 使用JSF组件 .....                 | 79  |
| 4.4.3 添加Action .....                          | 45 | 5.5.1 设置组件的属性 .....               | 79  |
| 4.4.4 配置Struts .....                          | 46 | 5.5.2 设置组件事件处理程序 .....            | 80  |
| 4.5 生成一个Web应用程序 .....                         | 47 | 5.5.3 使用表组件 .....                 | 81  |
| 4.5.1 清除并生成项目 .....                           | 47 | 5.6 使用支持Ajax的组件 .....             | 83  |
| 4.5.2 编译JSP文件 .....                           | 48 | 5.6.1 安装Ajax组件 .....              | 84  |
| 4.5.3 重复和停止生成 .....                           | 48 | 5.6.2 把组件库添加到一个项目中 .....          | 85  |
| 4.6 运行一个Web应用程序 .....                         | 48 | 5.6.3 使用Ajax组件 .....              | 85  |
| 4.7 定义Java应用程序服务器 .....                       | 49 | 5.7 数据绑定 .....                    | 86  |
| 4.7.1 使用Tomcat .....                          | 50 | 5.7.1 绑定到一个对象 .....               | 87  |
| 4.7.2 使用Sun Java系统应用程序服务器 .....               | 52 | 5.7.2 绑定到一张数据库表 .....             | 89  |
| 4.7.3 使用JBoss .....                           | 53 | 5.7.3 绑定到一个下拉列表 .....             | 93  |
| 4.7.4 使用BEA WebLogic .....                    | 54 | 5.8 使用“可视数据库查询编辑器” .....          | 94  |
| 4.7.5 设置服务器高级选项 .....                         | 55 | 5.9 小节 .....                      | 95  |
| 4.7.6 设置一个项目的应用程序服务器 .....                    | 56 | 第6章 创建Java企业项目 .....              | 96  |
| 4.8 HTTP监视 .....                              | 56 | 6.1 创建一个企业应用程序 .....              | 96  |
| 4.8.1 启用HTTP监视器 .....                         | 57 | 6.1.1 创建一个企业应用程序项目 .....          | 96  |
| 4.8.2 使用HTTP监视器 .....                         | 57 | 6.1.2 浏览企业应用程序项目结构 .....          | 97  |
| 4.9 浏览Web项目样例 .....                           | 58 | 6.2 创建一个EJB模块 .....               | 100 |
| 4.10 小节 .....                                 | 59 | 6.2.1 创建一个EJB模块项目 .....           | 100 |
| 第5章 创建Visual Web应用程序: JSF、<br>Ajax和数据绑定 ..... | 61 | 6.2.2 浏览EJB项目结构 .....             | 102 |
| 5.1 开始使用VWP .....                             | 61 | 6.2.3 使用会话Bean .....              | 102 |
| 5.1.1 安装VWP .....                             | 62 | 6.2.4 使用消息驱动Bean .....            | 106 |
| 5.1.2 配置VWP选项 .....                           | 62 | 6.3 创建一个企业应用程序客户端 .....           | 110 |
| 5.2 创建一个Visual Web应用程序 .....                  | 64 | 6.4 检查一个企业项目 .....                | 111 |
| 5.2.1 创建一个Visual Web应用程序项目 .....              | 64 | 6.5 使用持久 .....                    | 111 |
| 5.2.2 浏览Visual Web应用程序项目结构 .....              | 65 | 6.5.1 创建一个持久性单元 .....             | 112 |
| 5.2.3 设置项目属性 .....                            | 68 | 6.5.2 理解这个persistence.xml文件 ..... | 113 |
| 5.2.4 浏览Visual Web应用程序项目布局 .....              | 69 | 6.5.3 创建一个实体类 .....               | 114 |
| 5.2.5 生成一个项目 .....                            | 70 | 6.5.4 通过实体类创建JSF页 .....           | 117 |
| 5.2.6 运行一个项目 .....                            | 71 | 6.5.5 业务清单 .....                  | 118 |
| 5.3 使用VWP主题 .....                             | 72 | 6.6 小节 .....                      | 119 |



|                                     |     |
|-------------------------------------|-----|
| 第7章 创建Web服务项目：JAX-WS、SOA和BPEL ..... | 120 |
| 7.1 安装Enterprise Pack .....         | 120 |
| 7.1.1 安装捆绑的应用程序服务器 .....            | 120 |
| 7.1.2 使用预安装应用服务器 .....              | 121 |
| 7.1.3 跳过应用程序服务器安装 .....             | 122 |
| 7.2 创建Web服务 .....                   | 122 |
| 7.2.1 创建一个Web服务 .....               | 122 |
| 7.2.2 创建一个Web服务客户端 .....            | 127 |
| 7.2.3 通过WSDL生成Web服务 .....           | 132 |
| 7.2.4 创建一个消息处理程序 .....              | 132 |
| 7.2.5 创建一个逻辑处理程序 .....              | 134 |
| 7.3 使用SOA和BPEL .....                | 135 |
| 7.3.1 创建一个BPEL模块项目 .....            | 135 |
| 7.3.2 创建BPEL流程文件 .....              | 135 |
| 7.3.3 浏览“BPEL设计”窗口 .....            | 136 |
| 7.3.4 创建WSDL文件 .....                | 137 |
| 7.3.5 浏览WSDL编辑器 .....               | 140 |
| 7.3.6 使用BPEL设计器和BPEL映射器 .....       | 141 |
| 7.4 创建一个复合应用程序 .....                | 145 |
| 7.4.1 创建一个复合应用程序项目 .....            | 145 |
| 7.4.2 设置复合应用程序项目属性 .....            | 146 |
| 7.4.3 添加一个JBI模块 .....               | 146 |
| 7.4.4 测试复合应用程序和BPEL流程 .....         | 146 |
| 7.5 小结 .....                        | 148 |
| 第8章 创建UML项目 .....                   | 149 |
| 8.1 设置UML选项 .....                   | 149 |
| 8.1.1 “新建项目”属性 .....                | 149 |
| 8.1.2 “表示法”属性 .....                 | 150 |
| 8.1.3 “逆向工程”属性 .....                | 151 |
| 8.2 浏览UML项目布局 .....                 | 152 |
| 8.2.1 “组件面板”窗口 .....                | 152 |
| 8.2.2 UML“文档”窗口 .....               | 153 |
| 8.2.3 UML“属性”窗口 .....               | 154 |
| 8.2.4 UML“设计中心”窗口 .....             | 154 |
| 8.2.5 “图编辑器” .....                  | 154 |
| 8.3 选择一个UML项目类型 .....               | 155 |
| 8.3.1 创建一个与平台无关的模型项目 .....          | 156 |

|                            |     |
|----------------------------|-----|
| 8.3.2 创建一个Java平台模型项目 ..... | 156 |
| 8.3.3 对Java项目执行逆向工程 .....  | 158 |
| 8.4 使用模型元素 .....           | 161 |
| 8.4.1 添加模型元素 .....         | 161 |
| 8.4.2 过滤模型元素 .....         | 161 |
| 8.5 使用UML图 .....           | 162 |
| 8.5.1 创建类图 .....           | 162 |
| 8.5.2 创建“活动图” .....        | 165 |
| 8.5.3 导出图 .....            | 166 |
| 8.5.4 使用图布局 .....          | 166 |
| 8.6 使用设计模式和需求 .....        | 168 |
| 8.6.1 查看设计模式 .....         | 168 |
| 8.6.2 应用一个模式 .....         | 169 |
| 8.6.3 创建设计模式 .....         | 170 |
| 8.6.4 导入需求 .....           | 170 |
| 8.7 生成模型报告 .....           | 171 |
| 8.8 小结 .....               | 172 |

## 第二部分 NetBeans编码体验

|                        |     |
|------------------------|-----|
| 第9章 源编辑器 .....         | 173 |
| 9.1 排列和浏览文件 .....      | 173 |
| 9.2 访问源编辑器选项 .....     | 174 |
| 9.2.1 源编辑器上下文菜单 .....  | 174 |
| 9.2.2 源编辑器工具栏 .....    | 177 |
| 9.2.3 源编辑器键盘快捷键 .....  | 179 |
| 9.3 使用文件模板 .....       | 179 |
| 9.3.1 使用模板 .....       | 180 |
| 9.3.2 添加和新建模板 .....    | 181 |
| 9.4 在源编辑器中工作 .....     | 182 |
| 9.4.1 识别错误 .....       | 182 |
| 9.4.2 缩排代码 .....       | 184 |
| 9.4.3 创建和运行宏 .....     | 185 |
| 9.4.4 使用组件面板 .....     | 188 |
| 9.5 小结 .....           | 189 |
| 第10章 代码完成和代码模板 .....   | 190 |
| 10.1 代码完成 .....        | 190 |
| 10.1.1 使用“代码完成”框 ..... | 190 |
| 10.1.2 配置“代码完成” .....  | 192 |

|                                |     |                                          |     |
|--------------------------------|-----|------------------------------------------|-----|
| 10.1.3 使用“代码完成” .....          | 194 | 12.3 访问Javadoc .....                     | 227 |
| 10.2 代码模板 .....                | 196 | 12.3.1 绑定库和平台 .....                      | 227 |
| 10.2.1 使用代码模板 .....            | 196 | 12.3.2 查看上下文敏感的Javadoc .....             | 228 |
| 10.2.2 定制模板 .....              | 198 | 12.3.3 搜索Javadoc .....                   | 229 |
| 10.3 小结 .....                  | 200 | 12.4 小结 .....                            | 230 |
| 第11章 重构 .....                  | 201 | 第13章 管理版本控制 .....                        | 231 |
| 11.1 使用NetBeans重构选项 .....      | 201 | 13.1 使用CVS .....                         | 231 |
| 11.1.1 “移动类”重构 .....           | 202 | 13.1.1 配置一个项目使用CVS .....                 | 232 |
| 11.1.2 “重命名”重构 .....           | 203 | 13.1.2 执行普通的CVS操作 .....                  | 234 |
| 11.1.3 “安全删除”重构 .....          | 203 | 13.2 使用Subversion .....                  | 241 |
| 11.1.4 “尽可能使用超类型”重构 .....      | 204 | 13.2.1 安装Subversion .....                | 241 |
| 11.1.5 “从内层移至外层”重构 .....       | 205 | 13.2.2 执行普通的Subversion操作 .....           | 241 |
| 11.1.6 “封装字段”重构 .....          | 205 | 13.3 小结 .....                            | 244 |
| 11.1.7 “上移”重构 .....            | 209 | 第14章 用Ant管理编译和项目 .....                   | 245 |
| 11.1.8 “下移”重构 .....            | 209 | 14.1 Ant任务和目标 .....                      | 245 |
| 11.1.9 “把匿名类转换为内部类”重构 .....    | 211 | 14.2 在NetBeans中配置Ant属性 .....             | 248 |
| 11.1.10 “提取方法”重构 .....         | 213 | 14.3 NetBeans项目生成文件 .....                | 249 |
| 11.1.11 “提取接口”重构 .....         | 214 | 14.3.1 build.xml文件 .....                 | 249 |
| 11.1.12 “提取超类”重构 .....         | 215 | 14.3.2 build-impl.xml文件 .....            | 251 |
| 11.1.13 “更改方法参数”重构 .....       | 216 | 14.3.3 build-before-profiler.xml文件 ..... | 251 |
| 11.1.14 重构快捷键 .....            | 217 | 14.3.4 profiler-build-impl.xml文件 .....   | 251 |
| 11.2 用Jackpot重构 .....          | 217 | 14.3.5 project.properties文件 .....        | 252 |
| 11.2.1 添加一个Jackpot重构查询 .....   | 218 | 14.4 使用目标 .....                          | 253 |
| 11.2.2 运行一个Jackpot重构查询 .....   | 219 | 14.4.1 运行目标 .....                        | 253 |
| 11.3 小结 .....                  | 220 | 14.4.2 调试目标 .....                        | 253 |
|                                |     | 14.5 小结 .....                            | 254 |
| <b>第三部分 NetBeans和专业的软件开发</b>   |     | 第15章 JUnit测试 .....                       | 255 |
| 第12章 生成和访问Javadoc .....        | 221 | 15.1 创建一个JUnit测试用例 .....                 | 255 |
| 12.1 Javadoc元素 .....           | 221 | 15.1.1 创建一个新的测试类 .....                   | 255 |
| 12.1.1 类描述 .....               | 221 | 15.1.2 为一个现有的类创建一个测试 .....               | 257 |
| 12.1.2 类标记 .....               | 222 | 15.1.3 查看测试 .....                        | 258 |
| 12.1.3 类成员变量 .....             | 222 | 15.1.4 修改测试 .....                        | 259 |
| 12.1.4 构造器 .....               | 223 | 15.2 运行JUnit测试 .....                     | 260 |
| 12.1.5 方法 .....                | 223 | 15.2.1 查看测试结果 .....                      | 260 |
| 12.2 在NetBeans中创建Javadoc ..... | 224 | 15.2.2 生成测试用例报告 .....                    | 261 |
| 12.2.1 使用“自动注释”工具 .....        | 224 | 15.3 配置NetBeans中的JUnit属性 .....           | 262 |
| 12.2.2 生成项目Javadoc .....       | 226 | 15.4 小结 .....                            | 263 |
| 12.2.3 配置Javadoc属性 .....       | 226 |                                          |     |

|                                  |     |                                   |     |
|----------------------------------|-----|-----------------------------------|-----|
| 第16章 使用代码覆盖率工具 .....             | 264 | 17.5 协作工具使用场景 .....               | 285 |
| 16.1 开始使用Cobertura .....         | 264 | 17.5.1 团队编码 .....                 | 285 |
| 16.1.1 安装和运行Cobertura .....      | 264 | 17.5.2 修复一个出口限制的Web应用<br>程序 ..... | 286 |
| 16.1.2 检查一个Cobertura报告 .....     | 265 | 17.6 小结 .....                     | 286 |
| 16.1.3 使用Cobertura的功能 .....      | 268 | 第18章 使用NetBeans数据库工具 .....        | 287 |
| 16.2 在NetBeans中使用Cobertura ..... | 271 | 18.1 连接到数据库 .....                 | 287 |
| 16.2.1 结合项目生成文件 .....            | 271 | 18.1.1 添加一个数据库驱动程序 .....          | 287 |
| 16.2.2 为Ant目标创建快捷键 .....         | 274 | 18.1.2 添加新的数据库连接 .....            | 288 |
| 16.3 使用单元测试代码覆盖率查看器 .....        | 275 | 18.2 使用数据库表和视图 .....              | 289 |
| 16.4 小结 .....                    | 275 | 18.2.1 创建一个表 .....                | 289 |
| 第17章 使用NetBeans开发者协作工具 .....     | 276 | 18.2.2 修改一个表 .....                | 290 |
| 17.1 安装和配置协作工具 .....             | 276 | 18.2.3 保存一个表结构 .....              | 290 |
| 17.2 管理账号 .....                  | 277 | 18.2.4 重建一张表 .....                | 290 |
| 17.2.1 创建一个新账号 .....             | 277 | 18.2.5 创建一个视图 .....               | 291 |
| 17.2.2 设置现有的账号 .....             | 278 | 18.2.6 显示表数据 .....                | 291 |
| 17.2.3 修改账号 .....                | 279 | 18.3 编写SQL .....                  | 292 |
| 17.2.4 用多个账号登录 .....             | 279 | 18.3.1 使用“SQL命令”窗口 .....          | 292 |
| 17.3 生成联系人列表 .....               | 280 | 18.3.2 使用SQL模板 .....              | 293 |
| 17.3.1 添加联系人 .....               | 280 | 18.4 关联数据库结构和项目 .....             | 294 |
| 17.3.2 添加组 .....                 | 280 | 18.5 配置数据库连接池 .....               | 295 |
| 17.4 管理协作会话 .....                | 281 | 18.6 使用Apache Derby数据库 .....      | 296 |
| 17.4.1 加入会话 .....                | 281 | 18.7 小结 .....                     | 297 |
| 17.4.2 使用“Conversation”窗口 .....  | 283 |                                   |     |
| 17.4.3 共享项目和文件 .....             | 284 |                                   |     |

# 第一部分 创建和管理项目

## 第1章 集成开发环境

最初，代码是用简单的文本工具编写的，比如记事本。为了便于讨论，我把“最初”定义为20世纪90年代中期及以前，从那以后Java变得流行起来。结合使用文本编辑器和命令提示符，用户可以编写并编译代码。

这种方式没有提供最高效的开发环境，这一点很快就被证实了。比如，如果在文本编辑器中犯了一个语法错误，而没有办法意识到这个问题，一直到保存并编译这个文件。然后需要查看这个编译错误，在代码中定位出错的行，并试图确定出错的原因。在诊断代码中的问题时，编译错误并不总是完全有帮助。

许多编程新手都是从使用记事本和命令提示符环境开始的。这种方式本身并没有错，因为一些专业人员仍然在使用这种方式。对一个Java的初学者来说，有时候使用一个纯文本编辑器是最容易和最快的方式。但是，文本编辑器没有提供语言语法的帮助、编辑器集成、智能的重构支持或者其他编写代码的能力。

“查找”和“替换”是大多数文本编辑器有用的特性之一。利用这个简单的功能，程序员可以把一个单词或短语替换为另外一个单词或短语。在某些情况下，这确实起作用，但也可能产生问题。假如创建了下面的类：

```
public class SomeCode {
    public void myMethod1(String var){
        String FirstName =var.toUpperCase();
        //do something with FirstName
    }
    public void myMethod2(String var){
        String FirstName =var.toLowerCase();
        //do something else with FirstName
    }
}
```

这个SomeCode类有两个方法：myMethod1和myMethod2。如果以后需要重新命名myMethod1中的FirstName变量，就需要手动地编辑代码的每一行来修改这个名称。显然，这是一个简单的例子，但是如果myMethod1的代码有一百多行，而且FirstName出现在很多地方，那么，手动地编辑这些代码需要花很长时间。也可以使用文本编辑器的“查找”和“替换”功能，快速地将FirstName替换为一个新的变量名。但是，最初只希望修改myMethod1方法中的FirstName变量，而不希望修改myMethod2方法中的FirstName变量。使用“查找”和“替换”有可能错误地把myMethod1和myMethod2中的FirstName替换掉。当然，也可以逐个替换，但是这样费时间，而且容易发生人为错误。

一些文本编辑器为编程语言提供更高级的支持。流行的基于UNIX的工具Emacs提供了很多令人感兴趣的特征。Emacs提供高级的文本匹配和替换能力，通过插件，也能提供Java语法高亮显示、代码缩排、基本调试和编译支持。它有很多非常有用的功能，但仍然没有提供最便捷和最高效的环境。

## 1.1 为什么使用IDE

使用Emacs或者文本编辑器的人可能问的第一个问题是为什么使用IDE？一些程序员倾向于依赖一个特定的工具集或者编程语言，且不愿意改变。在当今这个不断改变的世界中一个重要的品质就是适应新技术的能力。

新的工具集可以在很多方面帮助专业程序员。一个程序员的时间应该用来写代码、修改代码和测试代码。您不需要浪费时间来设法解决如何在您的代码中重命名方法、生成项目文档、或者正确地编译一个包中所有的类。一旦您意识到您需要做什么操作，您的工具应该能简单地为您实现它。

集成开发环境（Integrated Development Environment，IDE）真正地为您的工作提供了一个完整的环境。它们把很多不同的工具以连贯的方式集合起来，所以您需要的服务和操作无缝地集成在了一起。

IDE含有如下的技术优点：

- 为操作提供图形化用户界面（Graphical User Interface，GUI）。
- 将源代码和配置文件组成一个项目的概念。
- 与编译器紧密集成。
- 结合了源代码库。
- 执行调整、分析和装载测试代码的能力。
- 集成了可重用的测试框架。
- 使用第三方插件和工具的能力。
- 单步调试代码的能力。
- 快速访问并容易生成项目文档。

一些使用IDE的更切实的业务优点如下：

- 降低了开发周期。
- 增强了代码的品质和可靠性。
- 使软件开发过程标准化。
- 为开发成员提供了一个通用的平台，减少了培训时间。

其中有些优点确实是有争议的，只有在仔细分析、实现和执行后才能意识到。一些其他因素也起着作用，但是一个真正好的Java IDE工具可以成为完成重要里程碑的基础，比如我上面给出的那个例子。

## 1.2 NetBeans与其他IDE工具的比较

NetBeans是我选择的Java IDE。这一点显而易见，因为我写了这本书，但是我有很多喜欢并使用NetBeans的正当理由。我使用过很多开发工具，比如记事本、TextPad、Emacs、vi、Macromedia Ultradeveloper、Macromedia Dreamweaver、Oracle JDeveloper、IntelliJ IDEA、

Borland JBuilder、Microsoft Visual Studio 和Eclipse。

每一种工具都有它自身的优点和缺点。它们都有各自的用户群，而且在它们周围都有完整的社区。虽然它们提供一些类似的特性，但在这些工具之间始终有一些区别。在决定使用IntelliJIDEA和Eclipse之间我曾犹豫不决。但我在使用NetBeans仅几个小时，并阅读了不同的指南后，我信服了。我下载、安装并开始使用它。很快我就发现这些特性都在我所意料它们所在的地方，它们也如我所料想的那样工作，并且只有很少或根本没有配置问题。在我看来，这就是一个工具应具有的。

我认为程序员应该使用NetBeans而不使用其他Java IDE的十个理由归纳如下，不分先后顺序：

**直观并易于使用的用来开发Swing的Matisse GUI设计器：**不管用户有少许还是没有Swing相关的知识，都可以把拖放元素拉到一个WYSIWYG（所见即所得）的设计窗口中。Matisse GUI设计器生成真正的Swing代码，而不是像其他一些工具那样只生成毫无用处的样板文件。在我最近参加的一个JavaOne会议上，我坐在一个使用JBuilder GUI设计器的先生的旁边。他在只看了我使用Matisse的两分钟后，就完全信服，并立刻自己下载它去了。

**强大的重构支持：**这是确定无疑的，通过使用一种类似于正则表达式的查询语言，Jackpot引擎允许Java类型敏感的重构。这种语言是由James Gosling设计的，使用非常简单，而且允许模式匹配和替换。对于这种查询，令人感兴趣的方面是，它们可以用来测试匹配特定的Java类型或者对象的实例。

**最好的代码探索器之一：**虽然我没有用过全部的探索器，但在我用过的许多种探索器中，NetBeans是最好的之一。用户可以探查内存、CPU和性能问题，也可以探查线程。我们可以从运行着的进程或者应用程序中加载或者分离这个探索器。它支持32位和64位，您也可以通过它来探查Enterprise JavaBeans (EJB) 模块和企业级应用程序。对于Mac的爱好者，它也支持在Mac OS X Intel系统上探查。

**UML项目支持：**程序员可以创建一个统一建模语言（UML）项目，这种项目适合于模型编码、工序或设计模式。UML项目可以直接连接到Java项目。当用户创建并修改UML对象和图时，相应的Java代码也自动生成。如果修改了被连接的Java项目中的源代码，图也会自动更新。有了导出图、生成代码和创建基于Web的项目报告这些功能，UML项目特性成为我喜欢使用的NetBeans中的最强的扩展之一。

**Ant集成：**NetBeans中的Java项目都是利用Ant的build文件组织起来的。当创建一个项目时，IDE生成build脚本和相关的目标（target）。然后用户可以启动这些目标，或者为了适合他们项目的需要，完全定制他们的build文件的结构。对于那些不熟悉Ant的用户，几乎没有影响，因为Ant目标的执行是直接连接到NetBeans中的菜单和按钮的。很多用户将发现，导入外部项目中已有的build文件很容易，而且速度也很快。NetBeans对于初学者，非常容易使用，对于专家，也非常容易定制。

**J2ME移动应用程序的支持：**即使您没有做过很多移动应用程序的开发，在看了例子，并阅读一个介绍性的指南后，您应该能很快意识到NetBeans移动工具的能力。NetBeans提供的应用程序的例子给人留下的印象很深刻。移动应用程序开发者还有一些不同寻常且功能强大的工具，如支持J2ME移动信息设备概要文件（Mobile Information Device Profile, MIDP）2.0、一个虚拟的移动设计器、一个无线连接向导及无线下载测试。

**开发者协作工具：**开发者可以登录到一个公共的或者私有的开发环境并共享代码。您可以

加入公共的会话，或者启动自己保密的私有会话。有一段时间我见过的最伟大的特性之一是，可以把代码或者整个项目拖放到聊天窗口中，在一个或者多个程序员之间共享代码。NetBeans支持多用户团队编码。当一个用户开始修改一段代码时，这段代码被高亮显示，对于共享它的其他用户，它被锁住了。在当前全球化体制下，开发团队散布在很多地方，因此这个工具被证实非常有用。

**易使用的更新中心：**您可以通过NetBeans更新中心快速地选择更新发行的站点，可以在这些站点上检查修改、更新和是否有新的模块。您也可以选择安装或者不安装您先前下载模块。这个更新中心比一些其他的Java IDE的更新工具更直观，并可以很快的更新NetBeans。

**与众不同的JSP和Tomcat支持：**NetBeans绑定了Apache Tomcat。一旦您使用新建项目向导创建一个Web应用程序项目，您就可以创建您的JavaServer Pages (JSP) 文件。然后您可以用鼠标右键点击任何一个JSP文件，然后选择“运行文件”。绑定的Tomcat服务器立即启动，您默认的Internet浏览器也打开了，在Tomcat中运行的JSP文件被显示出来。NetBeans甚至聪明地激活HTTP监视器。

**NetBeans HTTP监视器：**我做了很多与Web相关的Java开发工作。对于我来说，这是市场上所有Java IDE的最强和最独特的特性之一。在调试或者执行Web应用程序期间，HTTP监视器处于激活状态。您可以通过它来监视请求、HTTP header、cookies、会话、servlet上下文及客户端/服务器端参数。您不再需要写服务器端代码来读这些变量，把它们输出到一个日志中，然后查看这个日志文件。在NetBeans中，您可以调试您的代码，单步调试它，然后观察您需要的属性。

这些只是NetBeans已经提供的特性的一部分。其他的Java IDE也可能提供了一些上面提及的特性，但是没有一个是可以与NetBeans IDE直观的界面和集成的工具集相比。要了解NetBeans提供的所有特性，请继续阅读本书的其他章节。

### 1.3 小结

在这一章中，我们讨论了利用文本编辑器和工具进行编程的高级演变。显然，在过去50年软件发展的过程中，出现了很多的工具，也达到了很多的里程碑，但是我要找到开发者要使用正变得越来越多的IDE的基本理由。

使用类似于NetBeans这样流行的开发环境有很多好处。与和它竞争的工具相比，它能提供更高的生产力和便利，比如记事本、TextPad、Emacs和vi。用户可以利用很多特性，比如高级GUI设计器、JUnit的集成测试、性能探查器、调试工具及整个编译和打包的支持。

这一章中简要提及的很多NetBeans特性，在其他Java IDE基础上做出了重大的改进。由于以上这些及更多的理由，我相信NetBeans应该成为开发团队首选的Java IDE。



## 第2章 安装和定制NetBeans

要使用NetBeans，只要下载并安装它即可。这一章讲述如何做这些，也讲述如何检查更新和定制NetBeans的一些设置来适应自己的习惯。最后，我们快速地看一下经常用到的NetBeans窗口。

### 2.1 下载文件

到netbeans.org网站上可以下载NetBeans文件。这个网站提供了下载NetBeans软件和相关工具的几个不同连接。NetBeans的主要特性一般列在“最近下载”区域。这个页面的顶部也有一个“下载”连接。

根据进入下载区域的路径不同，网站会显示几个选择。选择期望下载的构建版使用的操作系统和区域语言，还要求选择构建类型和版本。除非喜欢冒险，否则还需要选择与构建类型一起的许可协议。

要确保下载的主产品是NetBeans IDE和NetBeans Profiler。

### 2.2 安装NetBeans IDE和核心附加软件

由于NetBeans可以安装在许多平台上，我只提及重要的安装概念。NetBeans可以安装在任何有Java虚拟机（Java Virtual Machine, JVM）的操作系统上。我一直用Java 6运行NetBeans，除非另外指定。

netbeans.org的下载页面提供了一个许可协议文档列表。在列表中有链接连到NetBeans推荐的安装手册。这些安装手册包括了在Windows、Solaris、Linux和Macintosh OS X上的基本安装过程。许可协议文档区域也有链接连接到Profiler Java开发工具（Java Development Kit, JDK）1.4附加软件和Profiler远程代理。

如果使用Profiler 5.0版，推荐在JDK1.5.04或者更新的版本上运行它。它有可能不支持JDK1.4，但是一个试验的附加软件必须允许这个功能。Sun不支持这个附加软件，所以自己决定要不要使用它。如果是用Java 1.4编写的应用程序，我建议安装最新的Java 6.0，然后把NetBeans的源兼容性设置为1.4。这样既解决了运行Profiler的所有问题，也维护了基于Java 1.4的代码。要设置一个项目的源兼容性，用鼠标右键点击一个项目名称，然后在上下文菜单中选择“属性（Properties）”。在选中的“源（Sources）”类别下，会看到一个称为“源代码级别（Source Level）”的字段，通过后面的下拉列表可以设置源代码应该兼容的Java版本。

我第一次下载和安装NetBeans，是在Windows XP上用Sun的Java 5，但后来更新到了Java 6。在执行Windows安装程序后，点击“下一步”按钮，接受许可协议，然后选择安装NetBeans的目录。就个人而言，我喜欢把所有与Java相关的产品放在一个地方。一般我以c:\java目录开始，在这个目录下，我会安装几个JDK、如同NetBeans或Eclipse这样的Java工具和放置所有Java相关项目和应用程序的一个目录。通常我采用下列的目录结构：

- c:\java\1.5.0.10。
- c:\java\1.4.2.11。

- c:\java\netbeans\5.5。
- c:\java\projects。

安装程序会搜索并询问使用哪一个JDK。如果安装了多个JDK，在窗口底部会看到一个JDK的列表，选择要使用的JDK（我推荐安装并选择JDK1.6.0），然后点击“下一步”按钮。通过安装程序继续，一直到完成。

接下来要安装NetBeans Profiler。Profiler的安装非常标准。在Windows上，一旦执行了安装程序并接受许可协议后，安装程序会提示选择一个合适的NetBeans IDE。选择刚才安装的IDE版本并点击“下一步”按钮，直到安装完成。

## 2.3 定制NetBeans JVM 启动选项

大部分人可能永远也不会想到要使用定制NetBeans JVM启动选项的能力。通过在NetBeans 配置文件中包含几个参数，就可以调整NetBeans运行所需要的JVM的内存使用配置，也可以改变使用的垃圾收集算法的类型。

如果在一台不完全标准的计算机（32位单处理器）上工作，改变JVM使用的垃圾收集程序可能不会有什么好处。但是，假如使用一个不同于Sun JVM的JVM，或者拥有一台多处理器或64位的计算机，那么就需要考虑这些选项。JVM厂商提供了一些关于垃圾收集程序运行和如何通过命令行参数配置它们的文档。在NetBeans启动过程中可以传入这些参数。

在NetBeans中，可以通过编辑NetBeans主目录中的/etc/netbeans.conf文件配置JVM启动参数。在这个文件中，可以看到一个netbeans\_default\_options属性，可以通过这个属性把JVM定制参数传递给NetBeans。首先会看到两个默认设置的参数，这两个也是我们要关注的，如下所示：

- -J-Xms32m 这个参数指定了初始分配给JVM堆的大小为32MB。
- -J-Xmx128m 这个参数指定了最多可以分配给JVM堆的大小为128MB。

在一些应用程序中增大Xms参数的设定值可以提高性能，因为JVM每次增加可用空间都要重新分配堆空间。在Java行业中有很多关于正确设置这些参数的方式的讨论，最安全的做法是把Xms参数设置为64MB或者128MB，把Xmx参数设置为尽可能的高（通常为系统所有内存的50%~75%），也要注意Xms和Xmx参数只指定JVM堆的大小，而不是JVM可以使用的所有内存数量，因为JVM中有一些项并不保存在堆中。

## 2.4 设置代理

有许多程序员，不管他们是在公司里还是在大学校园里，都需要在一个代理后面工作。NetBeans IDE的很多操作需要使用Internet连接，比如下载更新、连接到帮助文档、连接到外部数据库和Web服务器。

要配置NetBeans的代理设置，选择“工具 (Tools)” → “选项 (Options)”。默认情况下，NetBeans显示“基本选项 (Basic Options)”窗口，它含有一个“代理 (Proxy)”区域。如果不能看到它，点击“常规 (General)”标签查看该代理设置，如图2-1所示。

也可以在“高级选项 (Advanced Options)”区域中访问并修改同一个代理设置。只要点击“高级选项”按钮并选择“IDE配置 (IDE Configuration)” → “系统 (System)” → “系统设置 (System settings)”，代理设置就显示在窗口的右边面板中。一个称为“无代理主机 (Non-Proxy Hosts)”的附加字段也显示在列表中。通过设置该字段，可以指定一个主机的忽略列表，