



PEARSON  
Addison  
Wesley

由 GRADY BOOCH 作序

# Eclipse

# 精粹

Eclipse  
Distilled

(美) David Carlson 著  
张欣译



机械工业出版社  
China Machine Press



# Eclipse

# 精粹

(美) David Carlson 著

张欣 译



机械工业出版社  
China Machine Press

本书主要介绍当前最流行的Java开发环境Eclipse的特性与使用精要。全书共分两部分：第一部分说明了如何定制Eclipse工作空间、项目、视图和窗口，以便获得最高的效率，此外还说明了如何利用Eclipse的快速开发、导航和调试特性，最大化程序员的生产力、最优化代码的质量。第二部分完全侧重于敏捷开发的介绍，展示了如何用Eclipse快速实现小组所有权、重构、持续测试、持续集成以及其他敏捷开发操作。

本书可供应用开发人员参考，也适合广大程序设计爱好者自学。

Simplified Chinese edition copyright © 2006 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Eclipse Distilled* (ISBN0-321-28815-7) by David Carlson, Copyright © 2005.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as ADDISON WESLEY.

本书封面贴有Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

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：01-2005-3612

图书在版编目（CIP）数据

Eclipse精粹 /（美）卡尔森（Carlson, D.）著；张欣译. -北京：机械工业出版社，2006.2

书名原文：Eclipse Distilled

ISBN 7-111-18273-1

I. E… II. ①卡… ②张… III. JAVA语言-程序设计 IV. TP312

中国版本图书馆CIP数据核字（2005）第160080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：朱起飞

北京中兴印刷有限公司印刷·新华书店北京发行所发行

2006年2月第1版第1次印刷

787mm × 1092mm 1/16 · 13印张

印数：0 001-4 000册

定价：28.00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

本书对当今最强大的Java开发环境Eclipse做了简洁的介绍。它的组织结构非常简单明了，你可以根据自己的需要，迅速地找到自己需要的信息。如果你是Eclipse的初学者，那么在第一部分中可以全面地了解Eclipse IDE中的各种特性，包括Eclipse中的视图、窗口、编辑器和首选项等。有了这些基础工具的辅助，就可以很好地管理你的项目。第二部分介绍了如何利用Eclipse进行各种敏捷开发。这个部分中的每个主题都有自己的独立性，涉及敏捷开发处理的方方面面。你可以着重阅读某一章，深入了解敏捷开发的某个方面，也可以通读整个部分，全面地了解敏捷开发处理。

Eclipse是个开源项目，它的强大之处不仅体现在自身的特性上，它的可扩展性使Eclipse的潜能不可预测。本书的第二部分在介绍用Eclipse进行敏捷开发时，着重介绍了适用于这些敏捷开发需求的Eclipse插件。尽管Eclipse的插件有上百种，但是如果没有这种明确的说明，使用者就不知从何入手，那么这些插件的优势也就不能充分发挥出来。

本书不仅适用于初学者，便于他们初步了解Eclipse，还适用于经验丰富的Java开发者，可以作为他们的参考手册，以备他们随时查阅。本书结构简明，着重突出了Eclipse能够提高开发者生产力的特性，足以满足各个层次读者的需求。

本书由张欣组织翻译和统稿，参与翻译的还有黄璜、刘天成、潘秀燕、钱金蕾、王华红、魏胜、阎哲、王林、陈思锦、金川。

在翻译过程中，我们力求忠实、准确地把握原著的内容，但由于水平有限，翻译时间仓促，书中难免有错误和不准确之处，敬请广大读者批评指正。

张欣

于北京

2005年10月

## 关于作者

David Carlson于1991年获得了Arizona大学信息系统系的博士学位，研究方向是知识系统和面向对象技术。他有20多年的系统设计、程序设计和商业分析经验，1994年他担任咨询顾问，在此之前，他是位于Boulder的Colorado大学信息系统系的助理教授。目前，Dave在Colorado工作，是独立的咨询顾问。

此外，David Carlson还是《Modeling XML Applications with UML: Practical e-Business Applications》(Addison-Wesley, 2001)一书的作者。他设计和实现的Eclipse插件*hyperModel*实现了该书中的设计理念。

## 序

每个行业的从业者都会采用能增强手艺的工具，从而将自己的创造力发挥得淋漓尽致。对于编写代码的斗士来说，集成开发环境（IDE）就是他们的家，所有的工作都在这里完成。因此，IDE是开发者构建、部署和升级软件系统的中心。

在过去几十年中，开发者可用的资源从命令行工具升级到了独立的桌面工具，进而升级到了目前采用的集成环境。Eclipse可能是集成环境最重要的成就，不仅仅因为它是强有力的开发环境，还因为它是一种开源软件，来自各个领域采用各种语言的开发者都会给它提供广泛的支持和扩展。

Eclipse确实是一种功能繁多的平台，对于那些只想完成基本开发任务的开发者来说，这些功能未免显得太多了。在本书中，Dave做了一项杰出的工作，列出了Eclipse最重要最常用的情况，即安装、程序设计和重构、调试、测试、配置和发布。这种类型的书通常太简单，不过是产品文档的延伸，而Dave所做的远远不止这些，他提供了一个Eclipse的概念模型，还详细地介绍了在敏捷环境下使用该平台的可视性方案，这些方案既有适用于个人的，也有适用于团队的。此外，Dave还说明了Eclipse提供的人性化的工作环境和日常操作要点，通过这些，用户可以根据个人的特殊需要定制开发环境。

初学者会发现，这本书是掌握Eclipse精髓的向导，叙述雅致且严谨。我自己已经使用过Eclipse一段时间了，作为有经验的用户，我发现了很多有用的东西，使我能更加顺畅地使用IDE。

Grady Booch  
IBM院士



# 前 言

三年前我刚开始使用Eclipse时，就想读一本这样的书，但是今天我写了这样一本书。这本书并非像其他书那样假设你什么都不知道，不过如果你不熟悉JUnit或CVS，也不等于被判了死刑。如果你有Java开发的经验，或者已经用过Eclipse，本书清晰的说明和示例仍然会让你获益匪浅，使你成为一个权威用户。本书提炼出了Eclipse的大量特性和首选项设置，这使Eclipse成为不可缺少的工具，对于我来说，Eclipse已经是必不可少的了。

对于每个用Eclipse开发Java应用程序的人来说，无论是在创建扩展Eclipse的插件，还是在构建或测试企业的应用程序，本书中介绍的主题，都是知识要点。关于为Eclipse开发新的插件这个问题，有专门的书介绍（请参阅第1章结尾处的参考文献）。而本书是介绍如何使用Eclipse的。在学习Eclipse的过程中，我们采用了订单管理和产品目录这两个应用作为示例。

许多项目组采用迭代式开发过程，以迎合整个开发周期中新的或不断改变的需求，努力变得更加敏捷。你的项目组可能采用了特定的方法，如极限编程（XP），或者定制了一套适合你们的公司文化和项目需求的敏捷开发方案。成功的敏捷开发要求管理经验与软件开发经验相结合。本书介绍了Eclipse在编写、构建和测试代码过程中提供的对敏捷开发的支持。

本书是以我的个人经验以及我周围那些使用Eclipse构建产品代码的人的经验为基础的。我维护Eclipse新闻组三年了，在本书中，我加入了那些常见问题的答案和误解。在提炼这些主题的过程中，我尝试着让你更深入地了解Eclipse是如何工作的，如何才能最大程度地发挥它的价值。

如果你属于下列情况之一，那么你将从本书中受益：

- 你正在开发一种Java应用程序，刚开始接触Java，或者已经是Java专家。在创建和运行第一个Java项目时，你将在向导程序的指导下一步步前进，在调试、单元测试等情况下，则使用高级功能。
- 你正在为Eclipse创建新的插件，需要进一步了解Eclipse是如何工作，如何用于专业开发的。最成功的插件将与Eclipse用户执行的动作自然流程配合得天衣无缝。
- 你正在实践敏捷开发技术或者想进行这样的实践。即使你属于一个传统的、非敏捷的项目组，也会从应用单元测试、重构和对你交付的软件进行持续集成中受益。
- 你并不在意使用什么方法，只是想通过深入地理解Eclipse的用途以掌握其中的专门技术。
- 你是一个大学生，在用Eclipse做一个课程项目。使用这种开源的开发工具，能够完成更完整、更实际的作业和小组项目，为你快速地过渡到第一份工作做好准备。

## 本书的内容简介

本书分为两部分，无论你是Eclipse的新手还是寻求深入理解的有经验用户，都可以快速地

找到答案。本书编排是为了使各个章节能够连续地阅读，不过你也可以跳到第二部分中的专题，在需要查阅时再返回阅读前面的章节。

## 第一部分：开始

前七章会让你完全了解Eclipse IDE是如何组织、如何工作的。这一解释过程并非只是一系列屏幕图，我们有系统地逐步介绍了组织工作空间和项目、定制视图和窗口以及使用Java编辑器进行快速开发和代码导航的细节。你将学到如何通过单步调试多线程程序、显示和改变变量值、研究对象结构以及在挂起线程的背景下评估代码片断的方法对Java应用程序进行本地或远程调试。

新用户应该认真学习第一部分，这样才能理解Eclipse IDE是如何组织和配置Java项目，获取最优的编辑器特性的。Eclipse实现一个任务的方式通常有多种。选择哪种方式，有时取决于个人的工作喜好，有时由项目的结构和复杂度决定。我并不打算列出所有可用的方法，而是提供了一种基于使用Eclipse的一般经验的方法，在某些情况中，提供了几种可用的方法。

有经验的Eclipse用户可能也能从第一部分找到一些有用的东西，直接跳跃到第二部分也可以。

## 第二部分：变得敏捷

Eclipse本身是用敏捷开发过程创建的，具有给任何开发工作添加敏捷性的特性。Eclipse的创建者给它加入了一些工具，使我们的开发生涯变得轻松高效起来。

第8章介绍了敏捷开发的重点和它采用的迭代式开发周期。这一部分的其余各章分别着重介绍了敏捷开发的一个方面以及如何如何在Eclipse IDE中实现它。你可以以任何顺序阅读这些章，也可以在读完第一部分之前直接跳到其中一章。例如，如果你加入了一个已建立的项目组，可能就不需要从头开始创建自己的Java项目，而是需要检验CVS这样的储存库中的项目。在这种情况下，在开始你的研究前，应该先通读第13章。第二部分中的其他各章介绍了用JUnit进行持续测试、重构、用Ant进行持续集成以及编码标准。

第9章解释了如何用新的或改进过的插件增强Eclipse工作平台。使用集成的Update Manager可以检索本地或远程的站点，以查找相容的插件、制定自动更新计划以及管理工作平台的配置。可用的插件有上百种，而且创建新插件的速度还在不断加快。

## 前途展望

2004年2月，Eclipse社区改组成了一个非赢利性质的公司Eclipse Foundation。最初的开源代码是2001年11月由IBM公司提供的。未来它将不受任何独立个体的管辖，它的宗旨是促进Eclipse平台的创建、升级、推广和支持，培养开源社区和互补的产品、能力及服务的生态系统。给这个快速成长的生态系统提供的所有技术和开源代码都将保持公开并且没有版权。

Eclipse继续向纵深发展，作为一个开源社区项目，它前进的速度是大多数人始料不及的。很多开源软件还在开发中，而Eclipse平台已经提供了许多在它基础上构建的商业产品。要回答什么是Eclipse，已经变得越来越困难了。但是，毫无疑问，它的前景广阔，激动人心。



# 目 录

译者序  
关于作者  
序  
前言

## 第一部分 开始

第1章 Java IDE及其他 .....	2
1.1 Eclipse平台体系结构 .....	2
1.1.1 Eclipse运行时平台 .....	3
1.1.2 集成开发环境 .....	3
1.1.3 多功能客户平台 .....	7
1.2 其他Eclipse项目 .....	8
1.3 用Eclipse进行敏捷开发 .....	9
1.4 示例应用 .....	10
1.5 精粹 .....	10
1.6 参考文献 .....	10
第2章 Eclipse, 你好 .....	11
2.1 安装和启动 .....	11
2.2 Eclipse IDE工作平台 .....	12
2.3 创建新的Java项目 .....	13
2.3.1 打开Java视图 .....	14
2.3.2 创建一个Java项目 .....	14
2.3.3 创建一个Java包 .....	15
2.3.4 创建一个Java类 .....	15
2.3.5 定制代码模板 .....	16
2.3.6 编写方法 .....	18
2.4 运行应用程序 .....	18
2.5 精粹 .....	20
第3章 管理项目 .....	21
3.1 项目工作空间 .....	21
3.2 Eclipse资源 .....	22
3.2.1 项目 .....	23
3.2.2 链接资源的位置 .....	24
3.3 计划项目和依赖性 .....	26
3.4 精粹 .....	27
第4章 定制工作平台 .....	29
4.1 视图 .....	29
4.2 工作平台窗口 .....	31
4.2.1 标记窗口 .....	32
4.2.2 窗口布局 .....	35
4.3 资源编辑器 .....	35
4.4 设置首选项 .....	36
4.4.1 首选项设置的作用域 .....	37
4.4.2 共享首选项 .....	37
4.4.3 快捷键 .....	38
4.5 个人和小组首选项 .....	38
4.5.1 个人首选项 .....	38
4.5.2 小组首选项 .....	38
4.6 精粹 .....	39
第5章 快速开发 .....	40
5.1 扩展产品目录的设计 .....	40
5.2 互动的一对: 编辑器和大纲 .....	41
5.3 使用内容助手 .....	44
5.3.1 插入类型和变量名 .....	45
5.3.2 完成方法名 .....	46
5.3.3 源代码模板 .....	46
5.3.4 Javadoc内容助手 .....	48
5.4 使用快速修正 .....	49
5.5 生成获取器和设置器 .....	51
5.6 分析分层 .....	52
5.7 精粹 .....	54
第6章 Java项目配置 .....	55
6.1 Java构造路径 .....	55
6.1.1 源文件夹和输出文件夹 .....	56

6.1.2 项目依赖性 .....	57	10.3 编写测试用例 .....	111
6.1.3 项目库 .....	58	10.4 运行测试 .....	115
6.1.4 顺序和导出 .....	61	10.4.1 JUnit窗口 .....	116
6.2 创建共享的用户库 .....	62	10.4.2 定制测试配置 .....	118
6.3 Java编译器设置 .....	65	10.4.3 创建测试组 .....	119
6.4 创建记录日志的代码模板 .....	66	10.5 精粹 .....	119
6.5 精粹 .....	67	10.6 扩展 .....	120
6.6 参考文献 .....	68	10.7 参考文献 .....	120
第7章 调试代码 .....	69	第11章 重构代码 .....	121
7.1 启动调试对话 .....	69	11.1 何时重构 .....	121
7.2 检查和显示状态 .....	73	11.2 进行重构 .....	122
7.3 管理调试对话 .....	76	11.2.1 准备重构 .....	122
7.4 远程Java应用程序 .....	81	11.2.2 重命名包和类 .....	123
7.5 精粹 .....	85	11.2.3 提取接口 .....	124
		11.2.4 引入工厂 .....	127
		11.2.5 检索类引用 .....	129
		11.2.6 验证和提交结果 .....	130
		11.3 重构命令的目录 .....	130
		11.3.1 合成方法 .....	130
		11.3.2 在对象之间移动特性 .....	131
		11.3.3 组织数据 .....	131
		11.3.4 简化方法调用 .....	132
		11.3.5 一般化处理 .....	132
		11.3.6 组织类和包 .....	132
		11.4 精粹 .....	133
		11.5 参考文献 .....	133
		第12章 用Ant进行持续集成 .....	134
		12.1 自动增量构造 .....	134
		12.2 用Ant定制构造 .....	135
		12.2.1 Ant简介 .....	136
		12.2.2 构造属性 .....	136
		12.2.3 构造任务 .....	138
		12.2.4 构造UBL项目 .....	140
		12.3 Ant编辑器和大纲 .....	140
		12.4 在Eclipse中运行Ant .....	142
		12.4.1 Ant运行时配置 .....	143
		12.4.2 Ant窗口和控制台输出 .....	145
<b>第二部分 变得敏捷</b>			
第8章 敏捷开发的特征 .....	88		
8.1 敏捷宣言 .....	88		
8.2 迭代式开发 .....	90		
8.3 敏捷开发和Eclipse .....	91		
8.3.1 自适应处理 .....	91		
8.3.2 持续测试 .....	91		
8.3.3 重构 .....	92		
8.3.4 持续集成 .....	92		
8.3.5 集体所有权 .....	93		
8.3.6 编码标准 .....	93		
8.4 精粹 .....	93		
8.5 参考文献 .....	93		
第9章 更新Eclipse IDE .....	95		
9.1 查找和安装特性 .....	95		
9.2 安装没有特性的插件 .....	101		
9.3 设置更新首选项 .....	103		
9.4 精粹 .....	104		
9.5 扩展 .....	106		
第10章 用JUnit进行持续测试 .....	107		
10.1 选择测试策略 .....	107		
10.2 项目配置 .....	110		

12.4.3 自动项目构造器 .....	145	13.6 创建和应用补丁 .....	176
12.5 构造和测试完整的项目 .....	148	13.7 精粹 .....	177
12.6 精粹 .....	152	13.8 扩展 .....	177
12.7 扩展 .....	153	13.9 参考文献 .....	177
12.8 参考文献 .....	154	第14章 编码标准 .....	179
第13章 用CVS实现小组所有权 .....	155	14.1 Java编码样式 .....	179
13.1 用CVS进行小组程序设计 .....	155	14.1.1 代码格式化程序 .....	180
13.1.1 CVS服务器 .....	156	14.1.2 代码样式和模板 .....	182
13.1.2 CVS workflow .....	157	14.1.3 拼写检查程序 .....	184
13.2 共享你的项目 .....	158	14.1.4 编译器警告和错误 .....	185
13.2.1 添加一个储存库路径 .....	158	14.2 审查一致性 .....	186
13.2.2 共享项目 .....	160	14.2.1 配置Checkstyle插件 .....	187
13.2.3 小组项目集合 .....	163	14.2.2 运行Checkstyle规则 .....	188
13.2.4 CVS元数据文件 .....	163	14.2.3 用Ant自动运行Checkstyle .....	191
13.3 从CVS提取项目 .....	165	14.3 精粹 .....	192
13.4 与储存库同步化 .....	169	14.4 扩展 .....	192
13.5 版本管理 .....	174	14.4.1 Checkstyle .....	192
13.5.1 标志文件 .....	174	14.4.2 PMD分析工具 .....	193
13.5.2 提取一个版本 .....	175	14.4.3 实例化CodePro Advisor .....	193
13.5.3 分支和合并 .....	175	14.5 参考文献 .....	193

# 第一部分 开始

前七章会让你完全了解Eclipse IDE是如何组织、如何工作的。我们循序渐进地介绍了组织工作空间和项目，定制视图和窗口，使用Java编辑器进行快速开发和代码导航，配置Java项目以及使用调试器的细节。

# 第1章 Java IDE及其他

观察Eclipse，就像观察一颗切割良好的钻石。你能看到什么，取决于观察的角度和观察的深度。在某些观察者眼中，Eclipse是个具有一般IDE环境应有的特性的Java IDE，包括编辑器、调试器和具有各种首选项设置的项目构造器。而在另一些人眼中，Eclipse是工具集成的通用框架，具有扩展的、灵活的API插件。在www.eclipse.org的主页上这样说明，“Eclipse是一种通用的工具平台，一个开放的可扩展IDE，并没什么特别之处。”

上述两种观点都是有效的解释。随着Eclipse社区的迅速壮大，其他用户还会把它看作一种建模工具、一种商业原则管理应用程序、一种商业程序设计器、一种用C++语言编写的嵌入式系统的开发平台，或者一种编写HTML和进行文档管理的环境。如果采用多功能客户平台（RCP）配置，Eclipse就是任何终端用户客户应用程序的基础。

介绍Eclipse这颗钻石的各个切面的书已经有很多。本书则是把Eclipse作为一个构造Java应用程序的集成开发环境来介绍的。这些Java应用程序可能是其他开源工具、大型的企业系统或者新的Eclipse插件。无论是项目小组还是个人开发者，本书都适用。

Eclipse是由项目小组采用现代的敏捷开发过程开发的。Eclipse是由大型的分分布式项目组采用迭代式开发方法，使用Eclipse本身夜以继日地进行重构和测试开发成的。因此，Eclipse非常适用于采用这种开发方法的所有项目。本书介绍了几种敏捷开发小组常见的情况，详细解释了Eclipse如何协助完成这些任务。

本章将介绍Eclipse的下列方面：

- Eclipse平台体系结构的构造模块。
- 其他Eclipse项目提供的开发组件。
- 用Eclipse进行敏捷开发。

## 1.1 Eclipse平台体系结构

由于Eclipse的解释多种多样，因此很难简洁地说明它的体系结构。最一般的说法是“用于创建和部署插件的开发和运行环境的一种抽象视图”。此外，还有一种功能视图，把Eclipse描述为灵活的、可扩展的集成开发环境（IDE）。我们采用后面一种观点，把注意力放在Eclipse如何用一种非并行的方式提高软件开发工作的生产效率。

Eclipse IDE如下所示：

- 多平台。Eclipse 3.0的目标操作系统是Windows、Linux(motif和GTK)、Solaris、AIX、HP-UX和Mac OSX。
- 多语言。虽然Eclipse是用Java语言开发的，但是它支持用Java、C/C++和Cobol编写应用程序，此外它还支持Python、Perl、PHP和其他语言。Eclipse的插件则必须用Java编写。

• 多角色。除了编写程序，Eclipse还支持建模、测试、编写网页和许多其他角色。

图1-1展示了Eclipse IDE的功能构造模块。每个模块构造在它之下的基础模块之上。正是Eclipse平台的模块化，使它获得了空前的增长。整个平台是开源的，而且对于其他开源产品或者是添加了新模块的商业产品来说，都是免版税的。

深色的模块是本书介绍的重点。不过，我们会同时介绍一些插件，对于某些任务它们能提高平台的能力。这样的例子有基于Web的应用程序所用的单元测试框架、静态代码分析工具和确保与编码标准一致的样式检查程序。

本章接下来的几节将介绍图1-1中示出的构造模块。

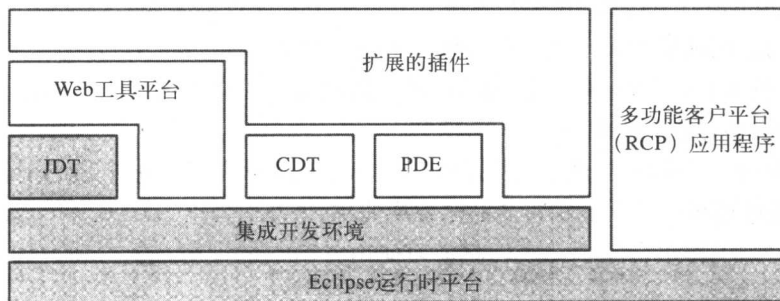


图1-1 Eclipse开发环境的功能构造模块

### 1.1.1 Eclipse运行时平台

核心的运行时平台提供了如下最基本的服务：

- 插件注册。装载插件并管理可用插件的注册表信息。
- 资源。管理操作系统文件和文件夹的独立于平台的窗口，包括链接资源的位置。
- UI组件。Eclipse的用户界面组件是以SWT和JFace库为基础的。
- 更新工具。Eclipse应用程序对安装和更新来自URL可寻址位置（包括远程的Internet站点）的插件具有内置的支持。
- 帮助工具。所有插件共享一个通用的帮助工具。

Eclipse的独立于平台的资源模型是在IDE中表示对象、文件夹和文件的重要基础，第3章“管理项目”中将介绍这一点。更新工具可以辅助你用新插件定制IDE，第9章“更新Eclipse IDE”将介绍它。

所有这些基本服务都是在非常小的内核上实现为插件的。Eclipse把所有东西都设计为插件，目的是完全一致地对待所有开发者（和所有插件）。这能确保该工具不仅对平台自身有效，对任何客户都一样有效。

### 1.1.2 集成开发环境

Eclipse IDE通过多种语言和多种角色的开发活动提供了一个通用的用户环境。构造在这个

IDE基础之上的插件不必凡事从头做起。

最重要的IDE特性如下：

- 共享窗口。许多Eclipse特性共享一些窗口，用于显示和编辑属性、罗列在代码中发现的语法错误以及列出工作任务。
- 视图。视图将把一组相关的窗口组织成适合特定任务（如Java开发、调试和版本控制）的布局。
- 首选项。一个对话框将把工作平台中安装的所有插件的首选项设置页面集中在一起。
- 搜索引擎。平台的搜索引擎提供了搜索资源和表示结果的一般功能。可以用资源特定的搜索标准（如Java、C++或UML模型资源）对这一工具进行扩展。
- 调试。为构造调试器和启动程序定义独立于语言的调试模型和UI类。
- Ant项目构造文件。Ant提供了通用的灵活构造服务，可以扩展到专门的任务。IDE包括一个Ant编辑器和运行时的配置。
- 面向开发组的。Eclipse提供了通用的API，与版本控制储存库进行集成。除了支持完整的CVS外，插件还可以被其他销售商的储存库使用。

在第4章“定制工作平台”中，我们将学习使用Eclipse的视图、窗口和首选项。第7章“调试代码”将介绍调试器，第12章“用Ant进行持续集成”将使用Ant文件。第13章“用CVS实现小组所有权”将介绍面向小组的功能。

还有一些Eclipse IDE特性是平台的一般设置不提供的，不过用户可以学到编辑器中的下列特性：

- 同步编辑器和大纲。大纲能以分级列表的形式列出正在编辑的文件的结构。选中大纲中的项目，可以使编辑器定位在该项目对应的位置。
- 内容助手。当同时按住Ctrl和Space键时，编辑器将在光标的位置提示要插入的内容。
- 模板。模板中定义了要插入文件的代码或其他文本的模式（如Java的for循环或HTML的table），是内容助手的一部分。
- 格式程序。语法规则是在首选项设置中定义和定制的，应用于编辑器选择或整个文件。
- 即时问题识别。在编辑器中进行输入操作时，错误或警告会立刻被识别，由编辑器左边缘处的标尺上的信息和图标突出显示出来。

在第5章“快速开发”中我们将学习使用这些特性，在第12章中使用Ant构造文件编辑器时，会再次见到它们。

### Java开发工具

Java开发工具（JDT）是Eclipse SDK中唯一的程序设计语言插件。不过，其他语言工具也可以使用或正在由Eclipse子项目和插件贡献者开发。图1-2展示了Eclipse的Java开发视图。

Java工具提供的基本功能如下所示：

- 编辑器、大纲、内容助手、模板和格式程序。这是为Java源文件提供的一般编辑器特性。



- Java窗口。有几个窗口用于Java项目的定位和管理。Package Explorer窗口是Java视图的基础，Java Browsing视图可以帮助开发者理解大型的多项目应用程序并在其中定位。
- 项目配置。更广泛的支持还包括配置Java项目的类路径、依赖性、库、编译器选项和许多其他特性。
- 调试器。Java工具提供了功能丰富的调试环境，可以设置断点、单步调试、观察和设置变量值以及在调试过程中改变方法的代码。

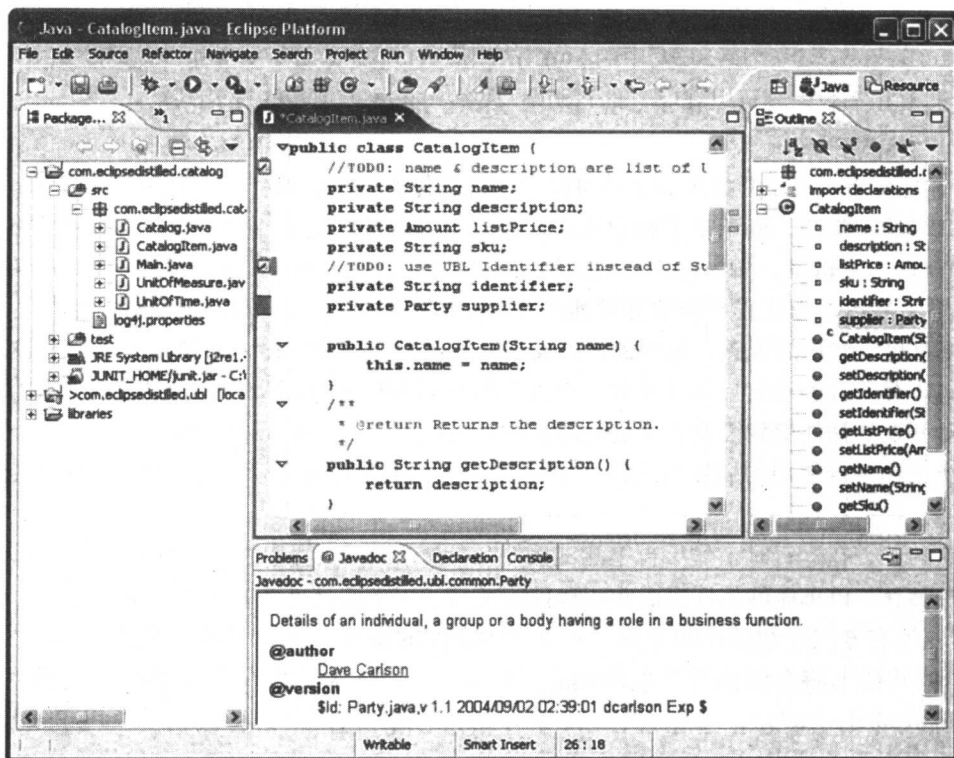


图1-2 Eclipse的Java开发视图

第6章“Java项目配置”将介绍Java项目的配置，第7章将介绍Java项目的调试。Java工具对敏捷开发技术提供了特别的支持。本章的1.3节“用Eclipse进行敏捷开发”详细介绍了这一主题，本书的第二部分“变得敏捷”是专为学习用Eclipse进行敏捷开发编写的。

### C/C++开发工具

C/C++开发工具（C/C++ Development Tools, CDT）项目正在为Eclipse平台创建功能完整的C和C++ IDE。虽然目前的重点是在Linux平台上进行开发和部署，但是有些贡献者想把这项工作扩展到其他方向，该项目的负责人对他们的参与非常感兴趣。例如，把该项目扩展到Windows、Unix和嵌入式平台，通过特殊的库、数据库或通信API提供开发向导，辅助开发。

CDT 2.0包括下列组件：

- C/C++编辑器。用C/C++语法着色规则、内容助手和格式程序专用化的一般平台文本编辑器。
  - C/C++调试器。用GDB提供的默认调试器实现。
  - C/C++程序启动程序。就像用可选的调试启动Java程序一样，CDT支持启动C/C++程序。
  - C/C++语法分析程序和语法API。语法分析程序（parser）是扩展CDT的其他插件的重要基础。
  - 搜索引擎。用能够精确定位代码定义和引用的C/C++语法选项专用化的一般平台搜索工具。
  - Makefile生成器。与Java开发中的Ant用法相似，makefile是C/C++项目开发所独有的。
- 有关细节和下载内容可以在CDT的Web站点www.eclipse.org/cdt/中找到。

### 插件开发环境

插件开发环境（PDE）提供的工具可以自动化插件的创建、操作、调试和部署操作。PDE是Eclipse SDK的一部分，而不是单独启动的工具。与一般的Eclipse平台一样，PDE具有各种平台特性（如窗口、编辑器、向导程序、启动程序等），它们与Eclipse工作平台的其余部分融合在一起，当开发者在Eclipse工作平台中开发插件时，它们在每个开发阶段都提供辅助作用。

- PDE视图。专用的视图，包括插件开发过程中最经常使用的命令的快捷方式和窗口。
- 主机工作平台和运行时工作平台。在开发插件过程中运行的工作平台是主机工作平台。当你对开发的插件感到满意，要进行测试时，可以启动另一个工作平台——运行时工作平台。这个工作平台使用的插件与主机工作平台一样，此外它还将运行在主机工作平台上开发的插件。
- 调试插件。在运行时工作平台上测试插件时，Java调试器可以进行完整的控制。
- 打包插件。PDE提供了打包插件和特性的工具，用于通过更新的站点分发给用户。

本书并没有专门介绍插件的开发，不过其中介绍的所有Eclipse功能都适用于PDE。这个丛书系列中的其他书将介绍插件开发的模式（Gamma和Beck，2004）和构造商业插件的细节（Clayberg和Rubel，2004）。

### Web工具平台

Web工具平台（Web Tools Platform, WTP）项目最初是由Eclipse委员会于2004年6月正式批准的，最初的开源代码来自于IBM和Object Web。在本书编写的同时，这些代码被评审并吸收到了第一个里程碑式的集成版本中。这个项目对Eclipse社区来说非常重要，可能也是本书大多数读者的兴趣所在。它仍然处于成型阶段，尽管如此，在此还是要浏览一下该项目的目标，总结它预计会给Eclipse IDE带来哪些集成特性。

Web工具平台项目的任务是在Eclipse平台和其他核心Eclipse技术之上构造一个通用的、可扩展的标准工具平台。该项目交付了一个框架和服务的通用基础，软件提供者可以在此之上为J2EE和以Web为中心的应用程序开发创建不同的专用解决方案。关键的目标是在把解决方案转换成真正开发的过程中，使产品的创新与不属于任何销售商的标准和技术保持一致。

WTP项目由两部分组成。Web标准工具（Web Standard Tools）这个子项目将为启用Web的应用程序提供一般的基础构造。这包括开发三级应用程序（展示、商业和数据逻辑）和把结果