

Broadview®
www.broadview.com.cn

Eclipse 插件开发

张鹏 姜昊 许力 编著

学习编程

编程

网络

学习

Ecl



电子工业出版社

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

Eclipse插件开发 学习笔记

张鹏 姜昊 许力 编著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书由浅入深、有重点、有针对性地介绍了Eclipse插件开发技术,全书分为4篇共24章。第一篇介绍Eclipse平台界面开发的基础知识,包括SWT控件的使用、界面布局、事件处理等内容;第二篇是插件开发核心技术,主要介绍插件开发的核心知识要点,包括行为(Action)、视图(ViewPart)、编辑器(Editor)、透视图(Perspective)等10章的内容;第三篇主要讲述插件开发的高级内容,包括开发高级内容、富客户端平台技术(RCP)、Draw2d,以及GEF介绍与实现等4个章节;第四篇则围绕插件开发和GEF应用两个主题,精心设计了两个程序开发实例,使读者能更加全面地理解插件开发相关技术。

本书内容全面,讲解仔细,不仅适合没有Eclipse平台技术基础的相关人士,也适合了解相关技术、具有一定插件开发能力的读者。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

Eclipse插件开发学习笔记 / 张鹏, 姜昊, 许力编著. 北京: 电子工业出版社, 2008.7
ISBN 978-7-121-05498-3

I. E… II. ①张…②姜…③许… III. 软件工具—程序设计 IV. TP311.56

中国版本图书馆CIP数据核字(2007)第185044号

责任编辑: 高洪霞

印 刷: 北京天宇星印刷厂

装 订: 三河市皇庄路通装订厂

出版发行: 电子工业出版社

北京市海淀区万寿路173信箱 邮编100036

开 本: 860×1092 1/16 印张: 41.75 字数: 1050千字 彩插: 4

印 次: 2008年7月第1次印刷

印 数: 5000册 定价: 75.00元(赠光盘1张)

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至zlbs@phei.com.cn, 盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线: (010) 88258888。



本书独一无二的优势

本书由多位具有丰富一线开发经验的资深程序设计师经过半年多的酝酿、写作、修改而完成。写作手法流畅而不失细腻，直白而不乏生动，既有精要的知识点分析，也有翔实的代码点评，从各个方面将Eclipse插件开发的技术进行了完整直观的展现。

本书的内容跳出了常规介绍Eclipse书籍的套路，围绕插件开发这个中心来展开讲解，同时考虑了读者的基础，有重点地加入了SWT和JFace等方面的内容，使得没有这方面基础的读者也可以迅速掌握插件开发的基础知识。本书的插件开发核心技术和高级进阶部分覆盖了插件开发中的大部分常用技术，其中不乏对相关其他技术的介绍，这样不仅介绍了插件的知识，更将解决问题的思路、方法及过程展现给读者，而这也正是本书要达到的最大目的。

本书的一大特色是图文并茂的叙述方式，通过各种样式丰富的图形，辅以精要的文字注释与说明，直观地进行知识点的分析和讲解，大大提高了读者的阅读兴趣和理解、掌握知识的速度。兴趣是最好的老师，而丰富多样的表现形式则能大大激发读者的学习兴趣，相信本书的努力能获得广大读者的认同。本书和市场上其他类似书籍相比，具有下面与众不同的特色。

形象

本书对书中的源代码采取绘图的讲解方式，使抽象变形象，让高深的编程理论“赤裸裸”地裸露在你的面前，使你轻而易举地领悟编程奥秘。这是本书最大的特色。

趣味

本书用拉链做素材，拉开学习的帷幕，用情趣似的学习笔记形式带你走进轻松的学习课堂，并且在书中让人感觉到烦躁的地方搭配以相关的对应图片，让你享受到学习中的乐趣，探索中的喜悦，胜利后的兴奋。

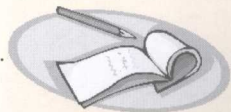
实用

本书的实用性较强，以经验为后盾，以实践为导向，以实用为目标，深入浅出地讲解Eclipse插件开发中的种种问题。特别是，在讲解时非常注重实践与理论的形象结合，为了使读者通过读懂源代码来掌握Eclipse插件开发技术，在实例举证时还给源代码绘制了大量的流程图。本书的所有源代码都已调试通过，并且放在了本书所附带的光盘中，读者拿来即可使用。

全面

本书内容全面，从基本的语法入手，以恰当的实例为导向，由浅入深地讲解Eclipse插件开发技术的基本理论知识，所讲解的内容几乎囊括了Eclipse插件开发技术的所有知识点。

Eclipse插件开发学习笔记



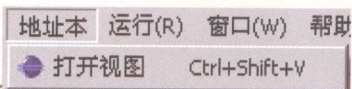


市场上介绍插件开发的书籍本来就很少，好不容易找到一本但是介绍的概念都很抽象啊，一点都看不懂，怎么办呢？

不用怕，这就是一本全面介绍插件开发的书籍，而且在介绍概念的同时画了大量的流程图，力求形象化，比如我们在讲解CreateControl中的功能时，通过下面的图解形象地表达了执行这个功能时的详细流程。



单击“打开视图”菜单



AddressView
CreatePartControl()

1 创建
Table Viewer

```
Viewer = new TableView(parent,
    SWT.MULTI | SWT.H_SCROLL |
    SWT.V_SCROLL |
    SWT.FULL_SELECTION);
```

2 设置表格的
创建界面

```
final Table table = viewer.getTable();
nameColumn = new TableColumn(table, SWT.LEFT);
nameColumn.setText("姓名");
nameColumn.setImage(Activator.getImageDescriptor(
    ImageKeys.IMAGE_PEOPLE).createImage());
nameColumn.setWidth(100);

categoryColumn = new TableColumn(table, SWT.LEFT);
categoryColumn.setText("类别");
categoryColumn.setImage(Activator.getImageDescriptor(
    ImageKeys.IMAGE_CATEGORY).createImage());
categoryColumn.setWidth(100);

Table.setHeaderVisible(true);
table.setLinesVisible(true);
```

3 初始化
Viewer

Viewer

```
viewer.setContentProvider(new
    AddressViewContentProvider());
viewer.setLabelProvider(new
    AddressViewLabelProvider());
Viewer.setInput(AddressManager.
    getManager());
```

4 为视图添加行为

```
makeActions();
hookContextMenu();
hookDoubleClickAction();
contributeToActionBars();
```



哇，真地很好，上面的图很清晰哦！

姓名	类别

最终的界面效果



很多书在说明操作步骤时，总是按部就班地罗列一大堆步骤说明，总是看了前面的忘了后面的，真麻烦！

这个难题本书也解决了，我们这本书在说明操作步骤时，采取给操作步骤编号的介绍方式。把一连串的操作界面按操作过程一一展现出来，并用箭头指示操作后出现的结果，每步操作都给了一个顺序编号。你看下面的例子。



1 属性视图显示flysky的信息

属性	值
地址元素信息	
类别	朋友
姓名	Flysky

Dingding	VIP
flysky	朋友
Lily	师长

2

属性	值
地址元素信息	
类别	VIP
姓名	朋友
	家庭
	VIP
	师长
	伴侣

改变地址元素类别为VIP

3

属性	值
地址元素信息	
类别	VIP
姓名	public

改变地址元素姓名为public

当鼠标离开“姓名”属性栏后，视图中的显示将会同时改变

Dingding	VIP
public	VIP
Lily	师长

4

图12-27 添加完属性视图后的效果图

这种介绍方式独具一格哦，怎么样？清楚怎么操作了吧！好戏还在后头呢，请看下一页吧。





本书除了界面介绍的亮点之外，请问还有什么更好的特色呢？

当然有了。我们在详细讲解代码的同时，还会把代码的执行过程通过图解的方式形象地表达出来，这样读者看代码就不会有任何难度了。下面的图12-30对前面代码的图解，请结合下图来读正文的代码，看看效果如何？

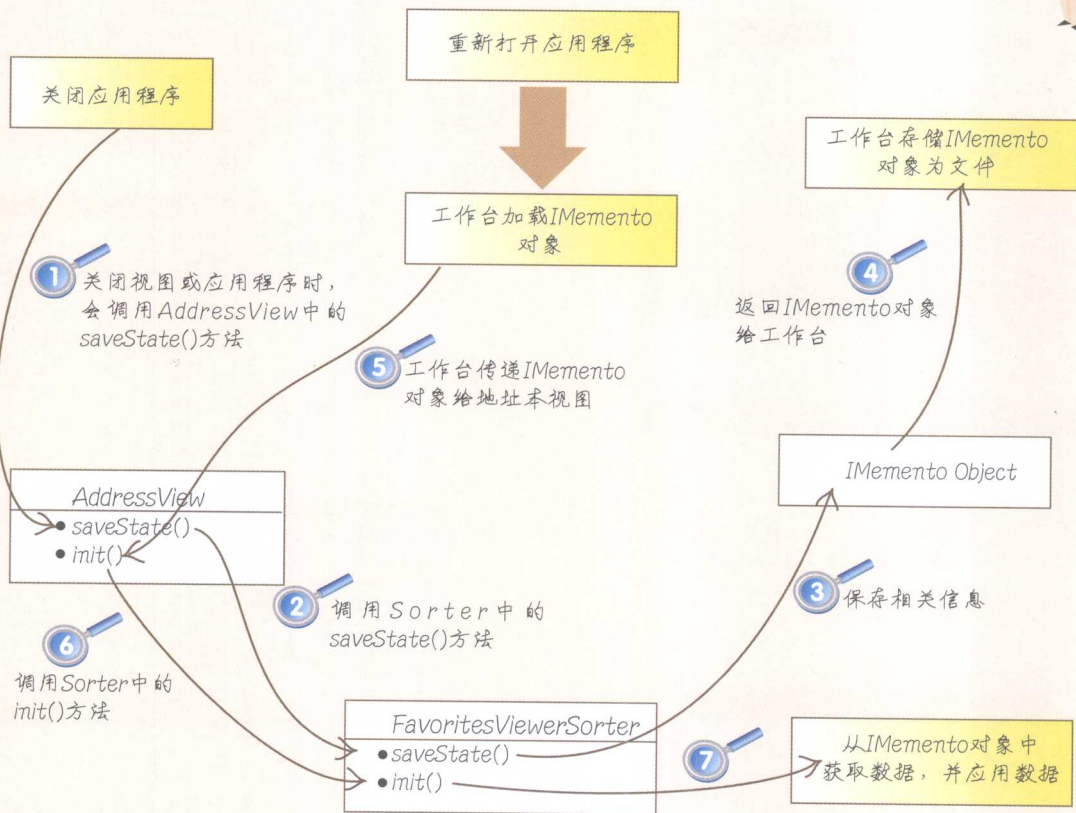


图12-30 保存并重新加载排序信息的过程

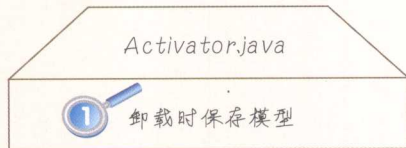


耶！！看了上面这个图，书中的代码确实简单了很多，不再像是天书了！



这本书真地很好，因为书中所介绍的概念和内容都形象而具体，请问这是什么写作方法？

本书采用的基本上是源代码绘图的写作方法，就是通过源代码绘图，把难懂的代码执行过程通过图解表达出来，这样就可以把代码所反映的基本原理也形象地表达出来了。书中的大部分代码都采取了这种写作方法。



```
Public void stop(BundleContext context)
throws Exception {

AddressManager.getManager().saveAddresses();
plugin = null;
super.stop(context);
}
```



```
public void saveAddresses(){
if(addresses == null) return;
XMLMemento memento = XMLMemento.
createWriteRoot(TAG_ADDRESSES);
saveAddresses(memento);
FileWriter writer = null;
try{
writer = new
FileWriter(getAddressesFile());
memento.save(writer);
}catch(IOException e){}finally{
try{
If(writer != null) writer.close();
}catch(IOException e){}
}
}
```



```
private void loadAddresses(){
FileReader reader = null;
try{
reader = new
FileReader(getAddressesFile());

loadAddresses(XMLMemento.createReadRoot(reader)
);
}catch(Exception e){}
finally{
try{
if(reader != null)
reader.close();
}catch(IOException e){}
}
}
```

图12-32 存储模型元素信息的步骤



你们这种源代码绘图的写作方法会很麻烦，写起来肯定很费力，你们真是用心良苦啊。

没有办法了，谁叫市场上的垃圾书那么多呢。我们只有创新才能脱颖而出了，下面这个图又是源代码绘图的典范。请好好体会本书的好处吧！

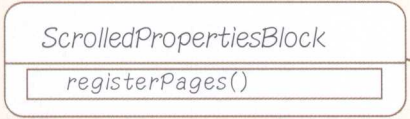


双击
表格列

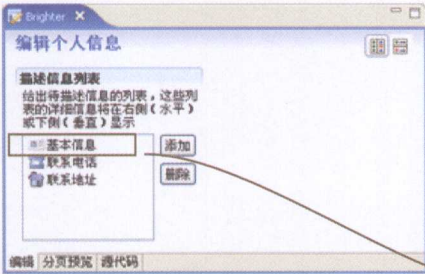
1 打开编辑器



2 创建Details部分

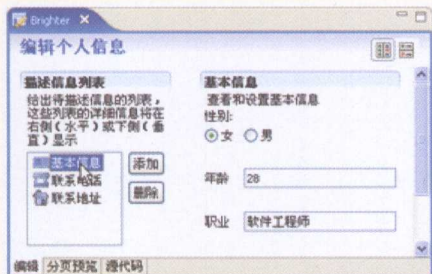


3 注册Details页面



4 初始化各页面

5 显示



6 选择基本信息

7 创建内容

8 更新显示

出现
Details部分

```
createChoiceItems(toolkit,
client, choiceKeysMap);
createTextItems(toolkit,
client, stringKeys);
```

```
AddressListProperties
• initialize()
• createContents()
• createSection()
• createTextItems()
• createChoiceItems()
```

```
BasicAddressListProperties
• createContents()
```

```
super.createContents(parent);
createSection(parent,
BasicAddressList.stringKeys,
BasicAddressList.choiceKeysMAP);
```

图 13-21 懒式加载Details部分的过程



下面又是源代码绘图的经典范例，通过下面的图和文中的代码，可以很好地理解这个对话框的代码背后的奥秘是什么，请读者慢慢欣赏吧。

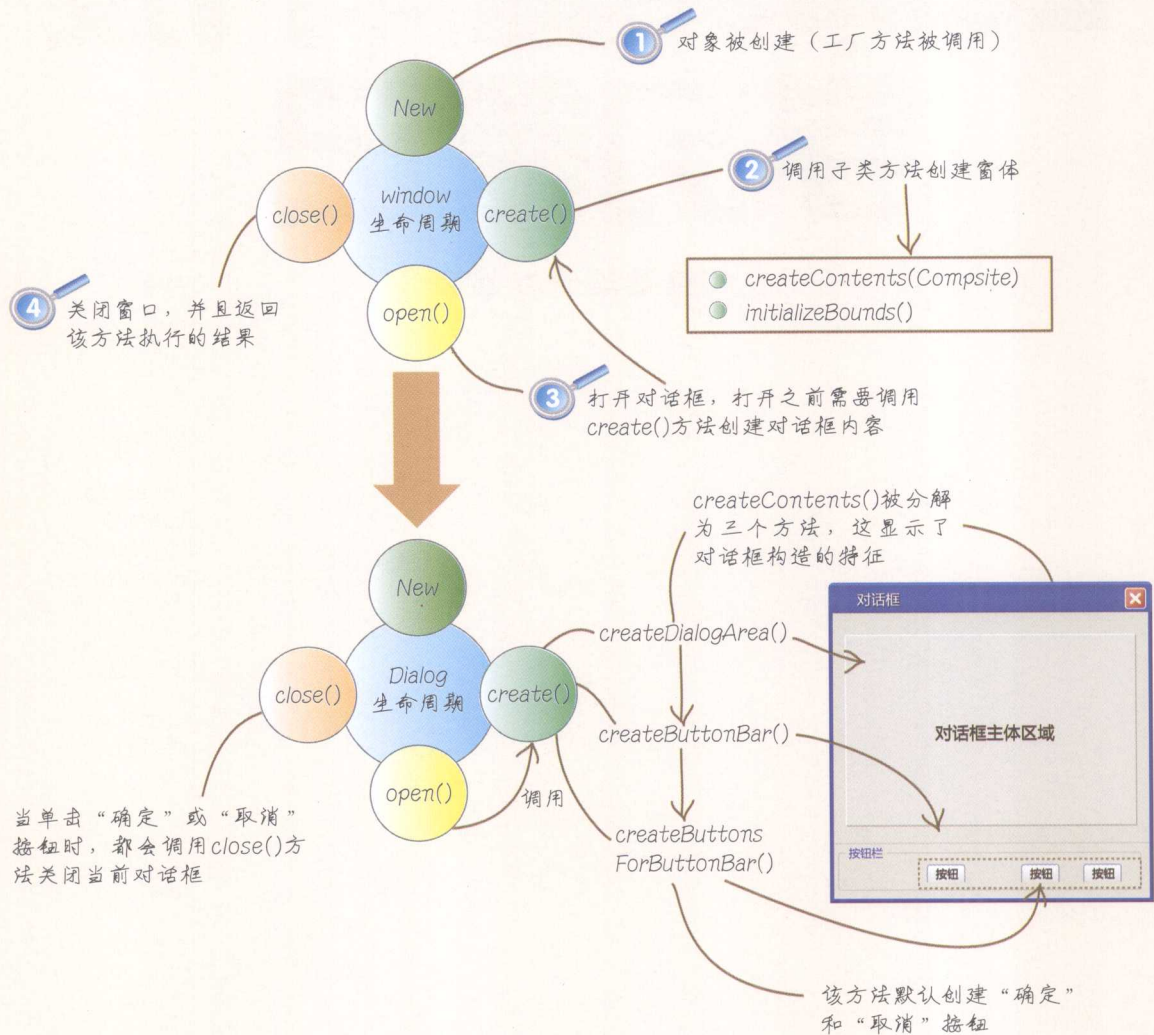


图 15-3 JFace 的“窗口”和“对话框”

前言

Foreword



IBM、Sysbase、Oracle、Bea、Borland、摩托罗拉，对于这些闪亮的名字，IT领域的学生、教师、公司程序员、项目经理都不会感到陌生，他们在不同的领域里相互竞争，然后一个共同的名字让他们联系到了一起——Eclipse。他们都是Eclipse基金会的成员，他们都为Eclipse的发展无偿提供资金和开发人员，而且这样的趋势还会加强，当然他们获得的回报也是十分丰厚的，他们可以避免落后于新的技术与架构的发展，并可以随时享用Eclipse发展的最新成果。在IBM的五大旗舰品牌DB2、WebSphere、Lotus、Tivoli、Rational长长的产品列表下面，随处可以发现基于Eclipse的产品，由于Eclipse的插件模式、扩展性、集成性，IBM以它为基础为Rational，WebSphere，Lotus和Tivoli构建了统一的平台。

作为Java集成开发环境（IDE），Eclipse作为Java的主流开发工具的地位2005年就已经确立，而且随着新版本的持续稳定发布，其统治地位亦有强化的趋势。但是仅仅作为一个开发工具，Eclipse不足以产生如此大的影响力，特别是企业级应用领域，各大厂商更加关注的是作为平台的Eclipse为成熟的企业应用提供的良好技术与发展的机遇。可以预见，一场Eclipse平台的应用风暴即将来临。

Eclipse不仅仅是一个平台，同时它也是通用、开放、标准的技术基础。它最耀眼的光环便是它的插件体系结构。虽然插件的历史很久，但是能像Eclipse这样实现彻底而又应用广泛的目前还不多。整个Eclipse就是由核心平台加插件构成的，Eclipse Platform Runtime作为整个系统的基础，为插件提供运行时容器，其本身不具备任何面向用户的业务功能，所有的业务功能都封装在Eclipse的插件当中，我们平常见到的Eclipse IDE实际上都是林林总总的Eclipse插件而已。

总而言之，利用Eclipse平台您可以：

★对Eclipse进行功能的扩展，例如建立一个和您的专业领域相关的图形建模工具，或者仅仅是实现一种特殊的代码管理模式。Eclipse提供了详细的帮助文档和源代码。

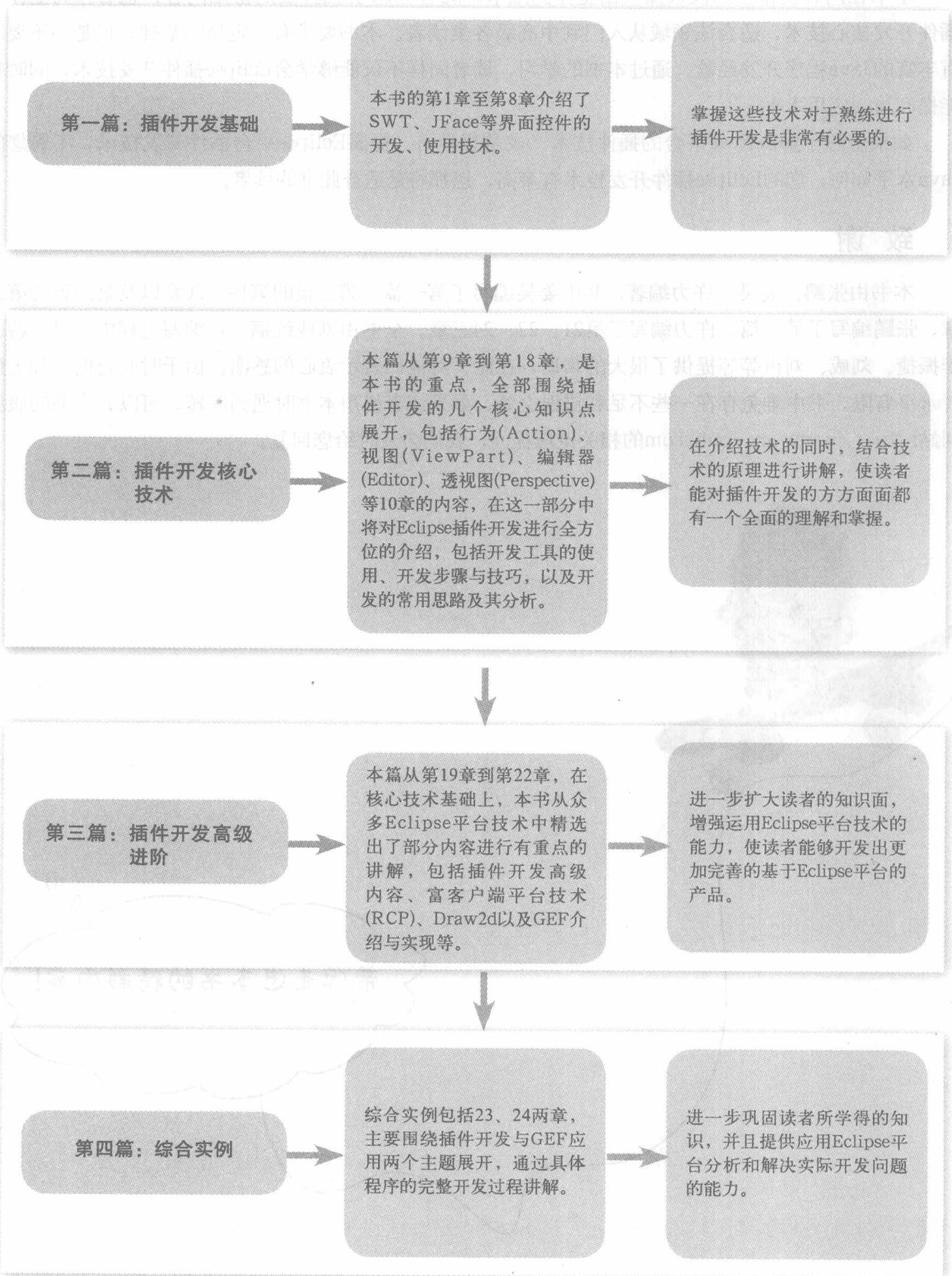
★开发一个独立的桌面应用RCP程序，Eclipse可以帮助您集中精力在业务领域模型，而不用去管太多的体系架构的问题。

★您可以为您的RCP程序提供自定义的扩展点，这样不认识您的朋友也能很容易发现您的设计意图，如果他愿意，可以很快为您的程序提供新的扩展，并保证扩展的程序具有一致的界面风格和操作规范。

本书的组织思路与内容结构

从Eclipse插件开发出发，衍生出了一个不断壮大的IT产业链，包括独立软件开发、软件外包、集成实施服务，以及相关的培训、咨询、出版物等。目前市场上专门介绍插件开发的书籍很少，很多的书籍都是IDE的应用，表面说Eclipse的使用，实际是一些其他零散的知识点，对Eclipse插件开发的介绍也很初级。这实际上并没有触及Eclipse最核心的价值，本书将集中内容，围绕Eclipse的RCP与插件开发及其相关的基础知识进行讲解与论述，从原理到实践，从界面开发、核心架构到插件应用，从Eclipse平台源代码到实际的项目，对本书的内容进行分层次有针对性的安排，使得读者能够深入浅出

地理解Eclipse插件开发的知识，熟练掌握插件开发技巧，各个层次的读者均能从此书中获益。为此精心安排了如下内容：



本书的读者对象

本书在内容安排上从浅入深，由整体到细节，既有SWT界面开发的基础内容，也有作为主体的插件开发核心技术，适合该领域从入门到中高级各类读者。本书要求有一定Java基础，但是并不要求有丰富的Java程序开发经验，通过本书的学习，读者同样不仅能够学会Eclipse插件开发技术，同时也能提高Java应用水平。

如果您想了解Eclipse平台的插件技术，或是想学习、提高Eclipse平台插件开发技术，不管您的Java水平如何，您的Eclipse插件开发技术有多高，您都将是适合此书的读者。

致谢

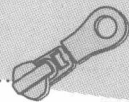
本书由张鹏、姜昊、许力编著，其中姜昊编写了第一篇、第三篇的第19、20章以及第四篇的第23章，张鹏编写了第二篇，许力编写了第21、22、24三章。全书由刘威统稿。在编写过程中，温才霖、李振捷、刘威、刘冉等等提供了很大的帮助，在此，对他们表示衷心的感谢。由于时间仓促，加上作者水平有限，书中难免存在一些不足和错误之处，如果读者使用本书时遇到问题，可以到本书的服务网站<http://www.cx-pub.com>的相关论坛提问，我们会及时给您回复。



带你走进本书的精彩内容!

目录

Content



第一篇 开发基础

第1章 Eclipse平台简介	01
1.1 Eclipse集成开发环境(IDE)介绍	02
1.1.1 安装及使用Eclipse IDE	02
1.1.2 为Eclipse IDE安装中文语言包	06
1.1.3 IDE的环境设置	07
1.1.4 使用帮助系统	09
1.2 什么是Eclipse	09
1.3 SWT/JFace技术	11
1.4 插件技术和OSGi	12
1.5 RCP技术	15
1.6 EMF技术	16
1.7 GEF技术	17
1.8 本章小结	18
第2章 SWT/JFace概述	19
2.1 SWT结构浅析	20
2.2 SWT API结构	21
2.2.1 组件类	22
2.2.2 布局类	23
2.2.3 事件类	23
2.2.4 图形类与系统资源管理	25
2.2.5 其他内容	27
2.3 JFace	27
2.3.1 查看器	27
2.3.2 资源注册表	28
2.3.3 字段帮助	29
2.3.4 操作和贡献	29
2.3.5 对话框、向导页和偏好设置	30
2.3.6 数据绑定	30
2.4 SWT与Swing	31
2.5 编写并发布SWT程序	33
2.5.1 第一个SWT程序	33
2.5.2 SWT程序的打包发布	37
2.6 本章小结	38
第3章 SWT编程基础	39
3.1 Display和Shell	40
3.1.1 Display的创建	40
3.1.2 Shell的创建	42
3.1.3 Display的事件队列和事件循环	43
3.1.4 Display的生命周期	46
3.1.5 监视器、边界和客户区域	47
3.1.6 SWT程序中的多线程	48
3.2 控件	50
3.2.1 控件类的继承结构	50
3.2.2 控件的构造和样式	52
3.2.3 控件的继承检查	53
3.2.4 控件的用户数据	55
3.2.5 控件的释放	55
3.3 图形资源	56
3.3.1 使用Color	56
3.3.2 使用Image	57
3.3.3 使用Font	59
3.4 高级内容	60
3.4.1 使用系统托盘	60
3.4.2 利用Region构造不规则窗口	61
3.4.3 在SWT中使用Swing	62
3.5 本章小结	63
第4章 使用基本控件与对话框	64
4.1 Button	65
4.2 Label	66
4.3 Text	68
4.4 List	70
4.5 Combo	73
4.6 ToolBar 和ToolItem	74
4.7 Menu和MenuItem	78
4.8 CoolBar和CoolItem	81
4.9 TabFolder和TabItem	83
4.10 对话框	85
4.10.1 消息框	85
4.10.2 文件与目录对话框	87
4.10.3 颜色对话框	89
4.10.4 字体对话框	90
4.11 本章小结	91
第5章 容器与布局管理器	92
5.1 Composite	93

5.2	Group	96
5.3	Shell	97
5.4	容器上下文菜单设置	100
5.5	容器颜色、背景和鼠标指针设置	102
5.6	布局管理概述	104
5.6.1	AbsoluteLayout布局 (No Layout)	104
5.6.2	FillLayout布局	105
5.6.3	RowLayout布局	106
5.6.4	GridLayout布局	110
5.6.5	FormLayout布局	112
5.6.6	StackLayout布局	114
5.6.7	布局的选择规则	116
5.6.8	自定义布局管理器	116
5.7	本章小结	120
第6章 界面开发工具 121		
6.1	安装Visual Editor	122
6.2	使用Visual Editor	123
6.2.1	Visual Editor的基本使用	123
6.2.2	控件布局	128
6.2.3	运行与调试	130
6.3	其他工具介绍	133
6.3.1	SWT Designer	133
6.3.2	Jigloo	133
6.4	本章小结	134
第7章 高级控件使用 135		
7.1	列表、表格和树	136
7.1.1	查看器 (Viewer) 框架	136
7.1.2	JFace 列表查看器(List Viewer)	139
7.1.3	Table控件	143
7.1.4	JFace 表格查看器(Table Viewer)	147
7.1.5	Tree控件	148
7.1.6	JFace树查看器 (Tree Viewer)	151
7.2	文本编辑器	155
7.3	滚动条、Scrollable、ScrolledComposite 和滑动条	158
7.4	进度条与进度指示器	160
7.5	浏览器与OLE	162
7.6	本章小结	165
第8章 SWT/Jface的事件处理 166		
8.1	SWT的事件处理	167
8.1.1	事件处理机制	167
8.1.2	低级事件类	170
8.1.3	高级事件类	171

8.2	常用事件	172
8.2.1	鼠标事件	172
8.2.2	键盘事件	173
8.2.3	Paint事件	174
8.2.4	应用举例	175
8.2.5	使用SWT模拟键盘/鼠标事件	177
8.3	JFace事件处理	178
8.3.1	操作 (Action) 与贡献(Contribution)	178
8.3.2	创建操作	180
8.3.3	使用贡献	181
8.4	本章小结	182

第二篇 核心技术

第9章 Eclipse插件体系结构 183	
9.1	Eclipse体系结构 184
9.1.1	Eclipse平台架构 184
9.1.2	插件工作模式 185
9.1.3	工作台层次结构 186
9.2	插件的加载过程 187
9.2.1	插件的安装 187
9.2.2	插件的发现和启动 190
9.2.3	插件信息的获取 190
9.3	插件的扩展模式 191
9.3.1	扩展和扩展点 191
9.3.2	扩展加载过程 194
9.3.3	常用扩展点 194
9.3.4	小结 195
9.4	本章小结 195
第10章 开发第一个插件项目 196	
10.1	创建插件工程 197
10.1.1	新建插件 197
10.1.2	使用向导 198
10.2	“插件开发”透视图 200
10.2.1	PDE视图 201
10.2.2	PDE运行时视图 202
10.2.3	清单编辑器 203
10.3	插件工程结构 203
10.4	插件文件 204
10.4.1	Plugin.xml文件 204
10.4.2	MANIFEST.MF文件 205
10.4.3	Build.properties文件 206
10.5	插件类 207
10.6	运行插件程序 208

10.7 调试插件	210	12.3 创建一个视图	248
10.8 发布插件	211	12.3.1 添加category	248
10.9 本章小结	212	12.3.2 在plugin.xml中声明视图	249
第11章 操作 (Actions)	213	12.4 视图类	250
11.1 Eclipse中的操作概览	214	12.4.1 视图方法	250
11.2 添加工作台窗口操作	214	12.4.2 视图控制	250
11.2.1 使用模板创建扩展	215	12.4.3 视图模型	252
11.2.2 定制操作集	217	12.4.4 视图内容	255
11.2.3 定制工作台菜单	218	12.4.5 视图标签	257
11.2.4 定制操作菜单项	219	12.4.6 视图排序	257
11.2.5 实现操作代理类	221	12.4.7 视图过滤	259
11.3 IAction与IActionDelegate接口	222	12.5 为视图添加操作	260
11.4 对象操作	224	12.5.1 视图选择	260
11.4.1 添加对象操作	225	12.5.2 添加/删除操作	261
11.4.2 操作的可见性	227	12.5.3 过滤操作	263
11.4.3 操作的过滤	228	12.5.4 快捷键支持	265
11.4.4 实现IObjectActionDelegate接口	228	12.6 视图间通信	265
11.5 视图操作	230	12.6.1 对属性视图提供支持	265
11.5.1 添加视图的上下文菜单	231	12.6.2 共享并监听地址本视图查看器	266
11.5.2 添加视图的工具栏	232	12.6.3 提供显示在Properties视图中的内容	266
11.5.3 添加视图的下拉子菜单	232	12.6.4 监听其他Workbench部分被选中的内容	271
11.5.4 实现IViewActionDelegate接口	233	12.7 添加状态栏支持	272
11.6 编辑器操作	234	12.8 视图状态	273
11.6.1 创建编辑器上下文操作	234	12.8.1 存储排序和过滤信息	273
11.6.2 添加顶层工作台菜单	234	12.8.2 存储视图元素信息	277
11.6.3 定义编辑器顶层操作	235	12.9 加载和卸载图标	279
11.6.4 添加编辑器工具栏操作	235	12.10 本章小结	281
11.6.5 实现IEditorActionDelegate接口	236	第13章 编辑器 (Editors)	282
11.7 快捷键映射	237	13.1 Eclipse编辑器体系结构概览	283
11.7.1 键绑定的策略	237	13.2 Eclipse工作环境中的编辑器	284
11.7.2 创建命令	237	13.2.1 AbstractTextEditor类	285
11.7.3 键绑定	238	13.2.2 MultiEditor类	286
11.7.4 命令与操作关联	239	13.2.3 MultiPageEditorPart类	287
11.8 本章小结	240	13.2.4 FormEditor类	287
第12章 视图 (Views)	241	13.3 为例子增加一个编辑器	289
12.1 Eclipse视图体系结构概览	242	13.3.1 声明编辑器	289
12.2 Eclipse工作环境中的视图	243	13.3.2 创建编辑器	290
12.2.1 资源导航	244	13.3.3 创建编辑器输入	291
12.2.2 PageBook视图	245	13.3.4 关联编辑器与编辑器输入	292
12.2.3 大纲视图	245	13.4 编辑器使用的数据模型	294
12.2.4 属性视图	246	13.4.1 基本数据模型	294
12.2.5 任务视图和书签视图	247	13.4.2 数据管理模型	299
12.2.6 项目资源管理器	247	13.4.3 文件映射模型	300
		13.5 编辑器页面	301

13.5.1	“编辑”页	302
13.5.2	“分页预览”页	311
13.5.3	“源代码”页	312
13.6	响应编辑器更改	313
13.6.1	更新模型	313
13.6.2	使“分页预览”页支持更新	317
13.7	保存编辑器模型	318
13.8	编辑器生命周期	322
13.8.1	已修改的编辑器	322
13.8.2	切换页面	324
13.8.3	保存内容	324
13.9	为编辑器添加操作	326
13.9.1	上下文菜单	326
13.9.2	管理编辑器操作栏	329
13.10	本章小结	333
第14章 透视图 (Perspectives) 334		
14.1	什么是透视图	335
14.2	创建一个透视图	336
14.2.1	透视图扩展点	336
14.2.2	创建透视图工厂类	338
14.3	IPageLayout	339
14.4	填充透视图	341
14.5	扩展现有透视图	344
14.5.1	添加视图和占位符	345
14.5.2	添加快捷方式	347
14.6	本章小结	348
第15章 对话框和向导		
(DialogsWizards) 349		
15.1	对话框和向导概述	350
15.2	对话框类别	350
15.2.1	SWT对话框	351
15.2.2	JFace对话框	352
15.2.3	常用JFace对话框	353
15.3	为例子增加SWT对话框	354
15.4	创建JFace对话框	355
15.4.1	使用Dialog类创建JFace对话框	355
15.4.2	为“地址本”视图创建过滤器对话框	356
15.5	向导介绍	362
15.5.1	向导对话框	363
15.5.2	向导	363
15.5.3	向导页面	364
15.6	添加向导	364

15.6.1	定义向导扩展	365
15.6.2	实现向导	366
15.6.3	实现向导页面	368
15.6.4	添加向导处理逻辑	371
15.7	本章小结	378
第16章 首选项 (Preferences) 379		
16.1	首选项页面结构	381
16.2	添加首选项页面	382
16.3	示例首选项	383
16.3.1	示例首选项页面	383
16.3.2	字段编辑器	385
16.4	为例子创建首选项页面	387
16.4.1	创建根首选项页面	387
16.4.2	创建“视图”子首选项页面	388
16.4.3	创建“编辑器”子首选项页面	391
16.5	本章小结	395
第17章 帮助内容 (Help Contents) 397		
17.1	Eclipse帮助介绍	398
17.2	使用Eclipse帮助	399
17.3	实现集成的帮助文档	400
17.3.1	添加帮助内容扩展	400
17.3.2	添加帮助内容	402
17.4	上下文相关帮助	404
17.4.1	声明帮助的上下文扩展	404
17.4.2	定义弹出信息内容	405
17.4.3	关联弹出信息同UI上下文	406
17.5	加入帮助按钮	408
17.6	本章小结	409
第18章 备忘单 (CheatSheet) 410		
18.1	使用Eclipse备忘单	412
18.2	为地址本插件创建备忘单	413
18.3	构建复合备忘单	418
18.4	链接备忘单到帮助	421
18.5	本章小结	424
第三篇 高级进阶		
第19章 插件开发高级内容 426		
19.1	自定义扩展点	427
19.1.1	扩展点基础知识	427
19.1.2	使用扩展点	435
19.2	插件的国际化支持	441
19.2.1	国际化方法	441