

计算机基础教育研究会

“计算机系统能力培养教学研究与改革课题”立项项目

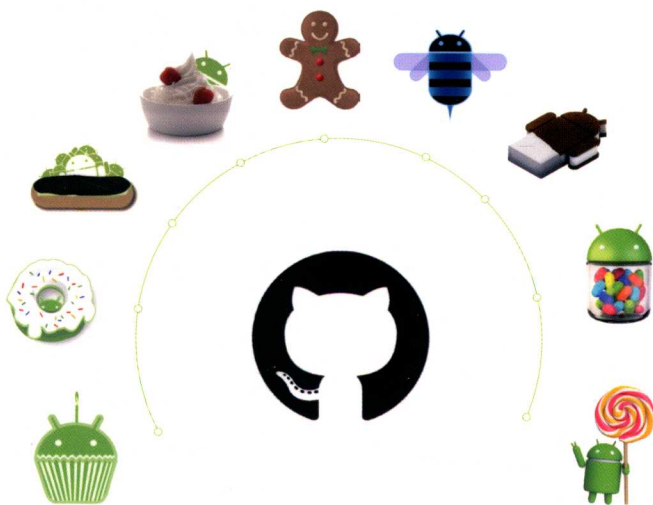


Android 实例详解

——基础进阶开发

韩迪 李建庆◎编著

- 业界接轨** | 使用大量的业界最新流行框架和开发工具
- 扩展学习** | 每章附参考文档，信手拈来学习更方便
- 典型实例** | 实例间互相独立但逻辑上融合
- 实用性强** | 不局限于理论介绍，实训皆采用了“项目驱动”的方式来讲授知识点
整本书集基础知识、核心技能、高级应用、项目案例于一体



北京邮电大学出版社
www.buptpress.com



Android 实例详解

——基础进阶开发

韩迪 李建庆◎编著

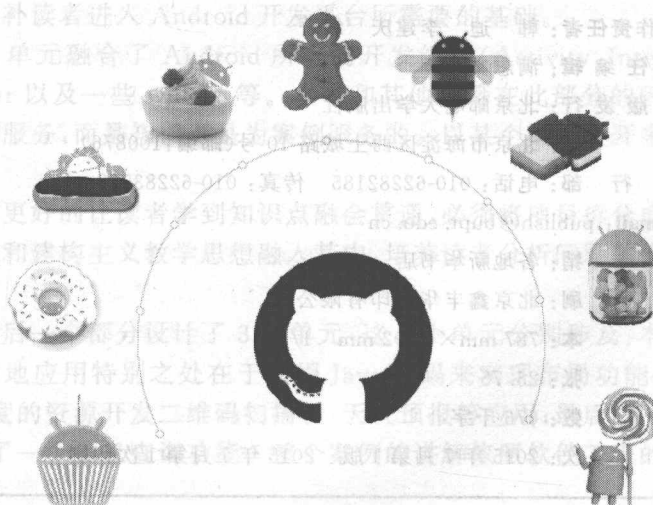
业界接轨 | 使用大量的业界最新流行框架和开发工具

扩展学习 | 每章附参考文档，信手拈来学习更方便

典型实例 | 实例间互相独立但逻辑上融合

实用性强 | 不局限于理论介绍，实训皆采用了“项目驱动”的方式来讲授知识

整本书集基础知识、核心技能、高级应用、项目案例于一体



北京邮电大学出版社
www.buptpress.com

内容简介

随着移动互联网的发展和4G网络提速,智能手机应用开发市场越来越完善。本书以此为前提,并不炫耀Android开发架构的优点,也不强调如何弥补其他系统的软肋,更多是结合Android开自身开源平台和利用Google等强有力的后盾,设计出更多方便生活,服务社会的应用程序。

本书以Android应用程序开发为主线,分问上下两册,此次为上册,适合有一定编程基础(如C/C++)的读者阅读。本册分为三个部分。分别为:

准备知识:深入浅出的罗列掌握Android开发需要的基础内容,并辅佐案例学习。

基本组件:利用实例对Android基本组件使用方法进行训练。

简单案例:以案例驱动学习为主线,将基本组件融合的综合案例教学。

图书在版编目(CIP)数据

Android实例详解:基础进阶开发/韩迪,李建庆编著. --北京:北京邮电大学出版社,2015.7

ISBN 978-7-5635-4374-8

I. ①A… II. ①韩…②李… III. ①移动终端—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆CIP数据核字(2015)第114573号

书 名: Android实例详解——基础进阶开发

著作责任者: 韩迪 李建庆 编著

责任编辑: 满志文

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路10号(邮编:100876)

发行部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京鑫丰华彩印有限公司

开 本: 787 mm×1 092 mm 1/16

印 张: 25.75

字 数: 676千字

版 次: 2015年7月第1版 2015年7月第1次印刷

ISBN 978-7-5635-4374-8

定 价: 55.00元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

北京邮电大学出版社
www.bupt.edu.cn

前言

随着计算机处理能力的高速发展,以及通信能力速度提高和云服务计算成本的降低,可穿戴设备的普及,IT行业逐渐朝着移动互联网方向蓬勃发展。同时由于人类对生产效率、生活质量的不懈追求,人们开始希望能随时、随地、无困难地享用计算能力和信息服务,由此带来了计算模式的新变革。

新的变革方向之一就是进入普适计算(Pervasive Computing 或 Ubiquitous Computing)时代。普适计算是指:是一个强调和环境融为一体的计算概念,而计算机本身则从人们的视线里消失。人们能够在任何时间、任何地点,都可以根据需要获得计算能力。

技术是以人为本的,所以将来智能手机市场发展的趋势的重点并不是 Android 新 SDK 提供了什么新功能;苹果又有了什么新的用户体验;又或者其他的手机平台又提供了什么更有趣的软件市场,而是普适计算的发展。其中包括计算机、手机、汽车、家电、可穿戴设备等所提供的综合网络服务。

所以本书以此为前提,并不在炫耀 Android 架构的优点,也不强调如何弥补其他系统的软肋,更多是结合 Android 自身的开源平台和 Google 这个强有力的后盾,设计出更多方便生活、服务社会的应用。

本书内容

本书以 Android 应用程序实例开发为主线,通过由浅入深的 13 个单元项目,全面涵盖了 Android 底层框架、通信应用程序开发、本地数据应用、网络数据应用、盈利模式分析以及云计算服务应用等多个开发领域。

本书分为 3 个部分:

第一部分:准备知识——前 1~4 个单元以 Android 预备知识点训练为主,分别融合了 Android 开发工具(其中包含业界使用的 Android Studio、Git 代码提交等)、Android 应用如何植入广告)、XML 和 JSON(轻量级的存储)、Android 布局、以及 Java 的基本功训练(内部类、事件监听、多线程以及异常处理),弥补读者进入 Android 开发平台所需要的基础。

第二部分:基本组件——5~13 单元融合了 Android 所有的开发组件(Activity、Intent、Broadcast、Service、Manifest、Handler 以及一些 widget 等)。本书和其他书籍在此部分的区别在于:讲解的安排不是案例为知识点服务,而是知识点是为案例服务的。以某个案例展开来讲解一个或者多个组件。

根据作者的教学和开发经验,要更好的让读者学到知识点融会贯通,必须将项目拆分成主要的功能模块,将项目实际开发经验和建构主义教学思想融入其中,培养读者分析问题的能力解决问题的能力。

第三部分:简单案例——本书最后一个部分设计了 3 个单元,这 3 个单元分别涉及:本地开发、网络开发以及数据库开发。本地应用特别之处在于利用 Java 代码来实现布局功能;网络开发包含了使用利用谷歌或者百度的资源开发二维码扫描机、天气预报等应用;最后的一个案例结合了 SQLite3 和多线程实现了一个信息查询功能。整个案例的讲解依照软件工程的规范化形式组织内容。

最后,包含了一篇关于 JNI 使用的附录,方便有需要的读者扩展学习:了解如何为自己的项目添加如 Open CV 等更强大图像识别库的支持。

本书三个部分环环相扣,注重对实际动手能力的指导,在遵循技术研发知识体系的严密性的同时,在容易产生错误、不易理解的环节配以详细的开发截图,并将重要的知识点和开发技巧以“知识点”、“注意”、“小技巧”等活泼形式呈现给读者。所有程序实例的讲解方面,按照“搜索关键字”(挖掘本章中在搜索引擎中的需要的关键字)、“本章难点”(帮助读者把握重点)、“项目简介”(项目的功能介绍)、“案例的设计与实现”(如何将功能需求进行分析、拆解最终实现)、“项目心得”(笔者的心得体会)“参考资料”(笔者在解决问题时候查阅的网页、书籍或者其他资料)和“常见问题”(初学者会出现的问题)。

本书特色

■ 适用于没有项目开发经验或程序设计基础薄弱的读者,以及希望快速开发安卓 App 的新手、编程爱好者、安卓爱好者。

■ 本书共分准备知识、基本组件和简单案例三个单元,每个单元中包含若干个主题,而每一个主题由 3 个左右单独的小案例组成,学习时间约为 2 小时,这些小案例是彼此之间独立的,但它们又有逻辑的关系。避免读者因为某个功能无法实现,而不能放弃整个项目学习。做到学习意义上的“高内聚,低耦合”。这样做的目的是:尽量避免很多实训类的教材,案例非常好,但是读者无法实现或者无法理解其中一个功能而不得不放弃整个项目的问题,将项目拆开虽然会带来更多的工作量,但是读者学习的效率会更高。

■ 培养分析问题、解决问题的能力,而不仅仅是一本指导书。笔者并不仅仅是教读者第一步怎么做,第二步这么做,而且思考这个项目该如何拆分,大的问题该如何变成小的问题,小的问题如何去寻找答案,解决的方法有多少种,哪一种更好,此外这个问题的解决方法还可以应用其他什么方面。希望能够达到授人鱼不如授人以渔,授人以渔不如授人以欲的目的。而且书中大部分案例都以放上应用市场,所以本书中在介绍开发过程中,同时也根据市场的反馈、用户体验等深入讲解原本代码中不妥当的地方。令读者开发经验更成熟。避免其他 Android 图书经常忽略对于错误反馈的讲解。

■ 每个章节最后给出参考链接,让读者能够有依可寻。因为每个人精通的范围是有限的,关键是为读者提供信息二次挖掘的入口。现在搜索引擎提供很丰富内容,问题的解决方法一般网上都会有,但是关键是:如何找到这些信息,然后如何整理。所以本书通过开始的“搜索关键字”和课后提供的“参考资料”的超链接及其注解,辅导读者能够更好的利用搜索引擎,提高自学能力。

■ 案例讲解中融入了大量作者在业界的开发经验,选取了大量的企业中实际的开发框架和工具,让学习者真正实现了和理论学习和业界实践相结合。

致谢

衷心感谢在本书编著过程中提供支持的机构和提供帮助的每一个人,包括在技术群和论坛中的热心网友。

首先特别感谢澳门基金会的资助。

感谢曾梓华,态度认真的 Android 工程师,移动互联网爱好者,在案例的设计和建议上,建议良多。

感谢朱冠州,在软件调试过程中的细心、踏实的工作,和较强责任心,值得借鉴。

感谢孙智威,资深全栈型工程师,一起无数次熬夜研究代码,对 Android 新技术执着和积极向上的态度,激励我们前行。

感谢团队中的每一位优秀技术成员,和你们交流,让我们找到了的进步空间;

最后感谢黄丽芳女士的理解与默默支持。

由于书中内容较多较新,难免有所疏漏,诚挚感谢读者指出书中不足,这样能和读者共同进步和提高。

作者

目 录

第一部分 准备知识

01	Android 整体接触	1
A	磨刀不误砍柴工——开发工具比较	1
B	万丈高楼平地起——从开发到打包	26
C	谈钱不伤感情——把 APP 放上市场	37
02	Android 基本功一 XML 和 JSON	56
03	Android 基本功二 Android 布局	65
04	Android 基本功三 Java 基本功训练	72
A	Android 基本功三——Java 内部类	72
B	Android 基本功三——Java 事件监听	81
C	Android 基本功三——Java 多线程	88
D	Android 基本功三——Java 异常处理	94

第二部分 基本组件

05	Activity 与 Intent	101
06	DDMS 调试与生命周期	113
07	Android 菜单功能实现	121
08	Android 对话框功能实现	130
09	Android 组件系列学习	137
A	人机交互事件(ActionBar+Spinner)	137
B	用户体验的细节(User Experience)	164
C	苹果能做我都能做(Gallery)	194
D	常用 widget 组件 1	210
E	常用 widget 组件 2	222
F	与时俱进的 Fragment	232
10	Intent 和 broadcast 组合 1:Intent 的过滤器使用	242
11	Intent 和 broadcast 组合 2:广播与短信服务	249

12	Android Service 后台服务	258
13	Android Handler 多线程	265
第三部分 简单案例		
14	Android 简单文件管理器	272
A	Java 代码布局	272
B	逻辑功能实现	288
15	网络 API 的使用	319
A	二维码和字典	319
B	天气预报	341
C	百度地图与定位	368
16	数据库结合多线程的信息查询	382
附录	Android 底层 JNI	394

第一部分 准备知识

01 Android 整体接触

A 磨刀不误砍柴工——开发工具比较

搜索关键字

IntelliJ Idea; Eclipse。

本章难点

Eclipse 最初是由 IBM 公司捐献给开源社区的,目前已经发展成为人气最旺的 Java IDE。Eclipse 插件化的功能模块吸引了无数开发者开发基于 Eclipse 的功能插件,在这样一个稳健的社区环境下,Eclipse 得到了无论是企业还是初学者的青睐。

IntelliJ Idea 为 JetBrains 公司的产品,在业界被公认为最好的 Java 开发工具之一,尤其在智能代码助手、代码自动提示、重构代码审查、创新的 GUI 设计等方面的功能可以说是超常的,也因此积累了一批特杆粉丝。

本章将基于 Android 开发环境,对他们进行比较并解剖分析两者的优点与缺点,让读者可以选择适合自己的 IDE,提高开发效率。

A.1 项目简介

A.1.1 UI 界面

Eclipse 的本身只是一个框架平台,但是众多插件的支持使得 Eclipse 拥有其他功能相对固定的 IDE 软件很难具有的灵活性。许多软件开发商以 Eclipse 为框架开发自己的 IDE。

Eclipse 最初是由 IBM 公司开发的替代商业软件 Visual Age for Java 的下一代 IDE 开发环境,2001 年 11 月贡献给开源社区,现在它由非营利软件供应商联盟 Eclipse 基金会(Eclipse Foundation)管理。截至今日,其仍然保持着比较频繁的更新频率,旨在为工程师打造更好用的 IDE。

版本代号	平台版本	主要版本发行日期	SR1发行日期	SR2发行日期
Callisto	3.2	2006年6月26日	N/A	N/A
Europa	3.3	2007年6月27日	2007年9月28日	2008年2月29日
Ganymede	3.4	2008年6月25日	2008年9月24日	2009年2月25日
Galileo	3.5	2009年6月24日	2009年9月25日	2010年2月26日
Helios	3.6	2010年6月23日	2010年9月24日	2011年2月25日
Indigo	3.7	2011年6月22日	2011年9月23日	2012年2月24日
Juno	3.8及4.2	2012年6月27日	2012年9月28日	2013年3月1日
Kepler	4.3	2013年6月26日	2013年9月27日	2014年2月28日
Luna	4.4	2014年6月25日	N/A	N/A

图 1A-1 发行历史版本

界面方面虽然 Eclipse 没有 IntelliJ Idea 的华丽惊艳,但却给与一种务实稳重的感觉,相信图 1A-2 的界面大家也比较熟悉了。

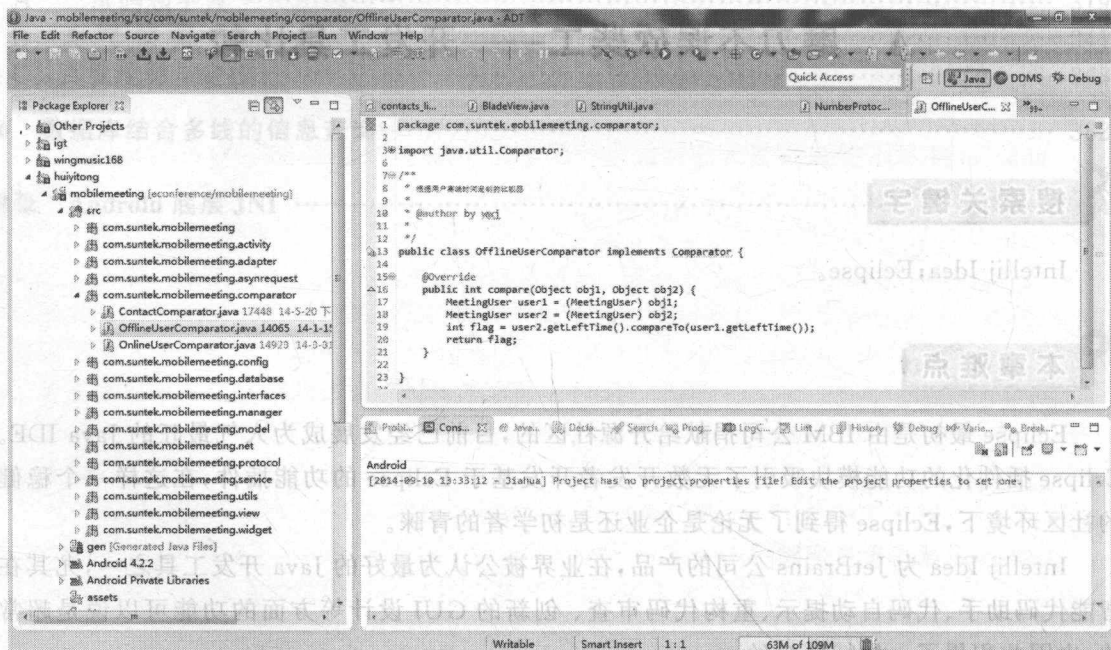


图 1A-2 Eclipse 默认主题界面

IntelliJ IDEA(下面简称 IDEA)被认为是当前 Java 开发效率最快的 IDE 工具。它整合了开发过程中实用的众多功能,几乎可以不用鼠标可以方便的完成读者要做的任何事情,最大程度的加快开发的速度。简单而又功能强大。与其他的一些繁冗而复杂的 IDE 工具有鲜明的对比。

2001 年 1 月发布 IDEA 1.0 版本,同年七月发布 2.0,接下来基本每年发布一个版本(2003 除外),当然每年对各个版本都是一些升级。3.0 版本之后,IDEA 屡获大奖,其中又以 2003 年的赢得的“Jolt Productivity Award”,“JavaWorld Editors’s Choice Award”为标志,从而奠定了 IDEA 在 IDE 中的地位。近日由谷歌推出的 Android Studio,也建立在相同的基础之上。目前版本为 13.1,2014 年 7 月 22 日发布,更新版本为 13.1.4。IDEA 的宗旨:“Develop with pleasure”。

初次使用 IDEA,最具备吸引力的地方非它的 UI 界面莫属了,其中最著名的要数其自带

的深蓝黑色为主调的 Darcula 主题,相信不少工程师对这种黑酷炫的颜色搭配有种独特的情怀。关于该主题的选择会在“常见问题”里进行详细说明。

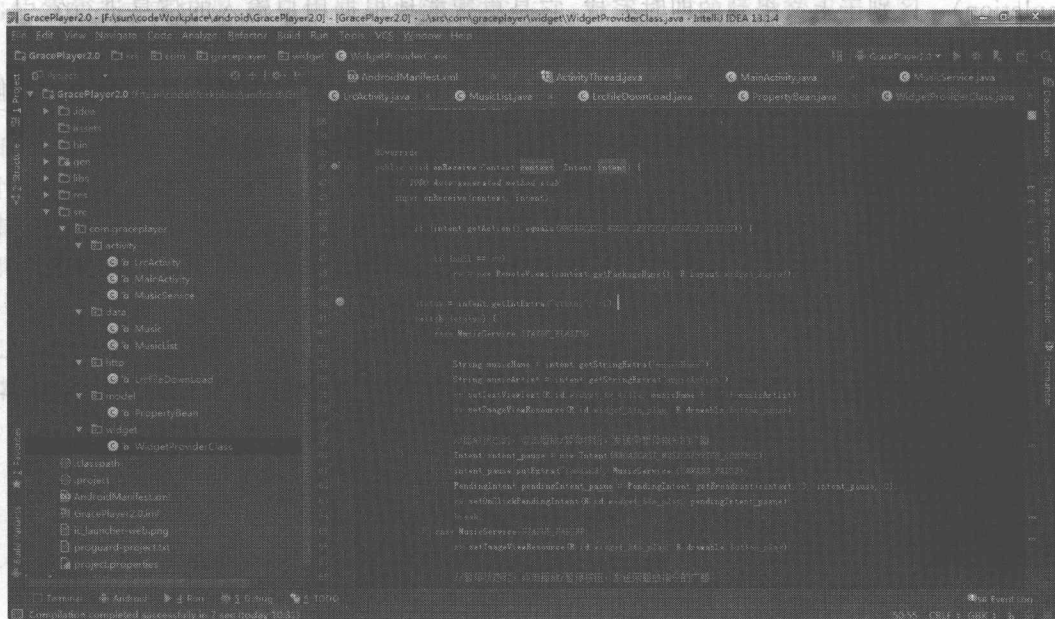


图 1A-3 IDEA 的 Darcula 主题界面

A.1.2 代码提醒

代码自动提示,智能补全是 IDE 与记事本的一个重要的区别之一,在这方面 IDEA 的表现就更强大,更智能了。它对代码的理解并不仅仅停留在【符合语法,编译无误】的层面,它对当前的上下文,相关的类、包、最近抛出的异常、每种类型最近用过的变量、当前类有哪些成员哪些方法它们都是什么名字什么类型、上下文中所有驼峰变量名的首字母和大写字母……等等都有非常准确及时的了解和“猜测”,并且给予读者最舒服最贴心的帮助。由于篇幅的问题下面就只简单介绍 IDEA 几个代码自动完成的特效。

1. 即刻完成(Instant completion)

第一个也是最吸引我的就是“即刻完成”特性,不同于其他 IDE,IDEA 可在任意地方提供这个功能,而不只是当读者要访问某个类的成员时。只需要输入单词的首字母,IntelliJ IDEA 就会立即给出最相关的、最适合此处代码编辑需要的选项供读者选择。

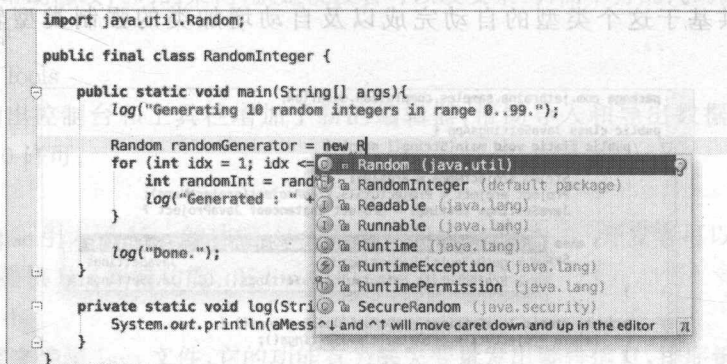


图 1A-4 即刻完成

2. 链式自动完成(Chain completion)

接下来的一个可让读者更多的提升编码效率的自动完成特性,名为链式自动完成(Chain completion)。区别于上文提到的即时完成,它是更智能地根据用户输入的信息进行索引,根据变量、属性和方法给出更深层次的代码提醒,具体到相关使用类里的具体的方法。

例如现在我们有 User 类,该类为存储登录用户的基本信息。当需要调用该类的时候,需要用到该类的时候一般会先声明具体的对象,如图 1A-5 所示。

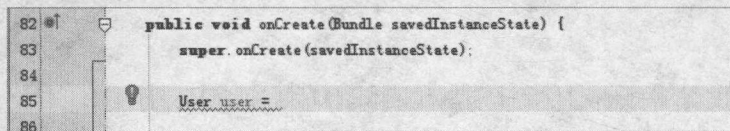


图 1A-5 声明对象

此时按下快捷键 Ctrl+Shift+Space(也即智能的代码补全功能,在常见问题中会具体介绍该快捷键),会提示并没有找到相关的类,并且建议进行搜索更深一层的方法进行调用。

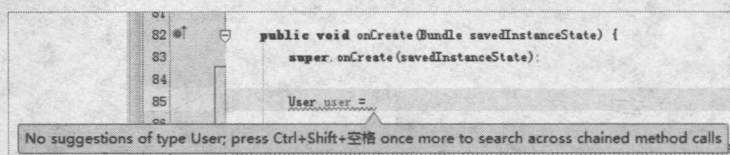


图 1A-6 按下 Ctrl+Shift+Space

接着再次按下 Ctrl+Shift+Space 键,IDEA 立刻索引整个项目,并且快速给出返回 User 类的方法,也即图 1A-7 MainApp.getUser()和 MainApp.getUserAccount()。

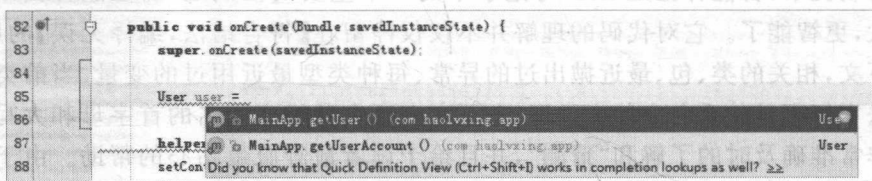


图 1A-7 链式自动完成

3. 数据流分析(Data flow analysis)

IntelliJ IDEA 被称为是最智能的 Java IDE 的原因是,IDEA 提供的各种功能都是基于对读者代码分析的结果,数据流分析就是其中一项。IDEA 分析数据流以便猜测运行环境中的变量类型,并提供基于这个类型的自动完成以及自动增加类的强制类型转换,如图 1A-8 所示。

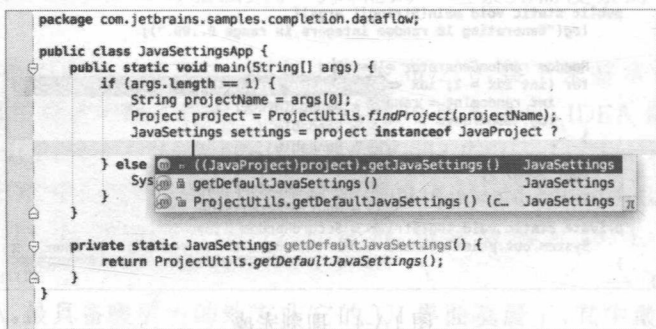


图 1A-8 数据流分析

而 IDEA 之所以能够做到如此智能与人性化,很大程度上是因为它索引缓存了整个项目目录,这使得 IDEA 需要占用系统大量的内存资源,也就造成了当配置较低的电脑里使用 IDEA 会启动速度超慢,也非常卡,此时使用 Eclipse 就不会有这样的情况出现。

A.1.3 插件

虽然大多数用户很乐于将 Eclipse 当作 Java 集成开发环境(IDE)来使用,但 Eclipse 的目标却不仅限于此。Eclipse 还包括插件开发环境(Plug-in Development Environment, PDE),这个组件主要针对希望扩展 Eclipse 的软件开发人员,因为它允许他们构建与 Eclipse 环境无缝集成的工具。由于 Eclipse 中的每样东西都是插件,对于给 Eclipse 提供插件,以及给用户

提供一致和统一的集成开发环境而言,所有工具开发人员都具有同等的发挥场所。因此 Eclipse 拥有非常丰富插件集群,如 FindBugs, Checkstyle, PMD, SourceHelper 等等,这是 IDEA 无法比拟的,目前 IDEA 官方公布的插件不足 400 个,并且许多插件实质性的东西并没有,在插件上的距离可见一斑。下面罗列了一些 Eclipse 常用的提高编写代码效率的插件:

(1) FindBugs

FindBugs 可以帮读者找到 Java 代码中的 bug,它使用 Lesser GNU Public License 的自由软件许可。

(2) Checkstyle

Checkstyle 插件可以集成到 Eclipse IDE 中去,能确保 Java 代码遵循标准代码样式。

(3) ECLemma

ECLemma 是一款拥有 Eclipse Public License 许可的免费工具,它提供了方便快捷的开发和测试环境。读者可以使用代码覆盖模式下的“launch”功能,用起来就像是真正的运行/调试模式。可以通过使用代码覆盖试图,高亮源文件,计数来分析代码。

(4) JDepend4Eclipse

JDepend4Eclipse 可以帮助遍历文件夹,协助量化设计的质量。它使用 Eclipse Public License v1.0 许可。

(5) PMD

PMD 是一款代码分析器,用来检测变量和写得不好的代码。(更新网址:<http://pmd.sourceforge.net/eclipse/>)

(6) SourceHelper

SourceHelper 可以协助编码和调试,对写好的代码提供说明。

(7) Structure101

Structure101 帮助修改代码的架构,就是说读者可以改变架构,而不打乱代码。

文本编辑插件

(8) AnyEdit Tools

AnyEdit 为输出控制台和工具栏增加了新的编辑器,帮助导入和导出数据。它使用 Eclipse Public License v1.0 许可。

(9) Eclim

Eclim 给 Eclipse 引入了 Vim 的功能,Vim 是最好的编辑器之一。开发者可以采用不同的语言来编写代码,它也提供 bug 纠错功能。Eclim 使用 GPLv3 许可。

(10) Eclipse-rbe

Eclipse-rbe 用来编辑 Java 文件,它的功能有为缺失变量发出警告信息,排序键值,转变 Unicode 编码等。它使用 GNU 库或 LGPL 许可。

依赖管理

(11) Apache IvyDE

Apache IvyDE 集成了 Apache Ivy 的依赖管理功能。它能管理“ivy.xml”中的依赖,也能够配合 WPT 和 Ant 插件的使用。

(12) M2eclipse (Maven 插件)

M2eclipse 能够管理简单的项目,也能管理多模块项目,它能在 Eclipse 中启动 maven。它提供依赖管理,能提供自动下载。

版本控制插件

(13) Subclipse

Subclipse 是一个 SVN 插件,它为 Eclipse IDE 提供 subversion 支持,使用 EPL 1.0 许可。

(14) EGit

EGit 提供同步视图,可以读.git/下排除的文件,提供 rebase 功能,为 pull 和 push 提供精简的操作。

(15) MercurialEclipse

MercurialEclipse 是个流行的版本控制系统。它提供了 clone repository 功能, push-pull 同步功能,以及简单的回滚功能。

(16) P4Eclipse

P4Eclipse 吸纳了 Perforce 管理系统的特长。它提供了以开发者为导向的一系列的功能,并且支持许多 Agile 开发流程以及传统开发流程。

框架开发插件

(17) Spring Tool Suite

Spring Tool Suite 提供了 XML 文件预览以及图形化的 spring 配置编辑器等开发环境和工具,它能让开发变得更简单。

(18) Spring IDE

Spring IDE 帮助读者开发 Spring 应用,它提供了图形编辑界面。它还能用图形的方式显示 bean 之间的关系。(更新网址:<http://springide.org/updatesite>)

(19) Hibernate

Hibernate 能够创建或更新数据库 schema,运行 hibernate 查询语句,创建映射文档。用户可以将一个 Java 类和一个相关的映射文件同步起来。

(20) Jboss Tools

JbossTools 3.2x 支持 JBoss, Hibernate, Drools, XHTML, Seam 等。(更新网址:<http://download.jboss.org/jbosstools/updates/JBossTools-2.1.2.GA>)。

其他功能插件

(21) ASM—二进制预览插件

ASM Bytecode Outline 插件可以显示当前的 Java 文件或 class 文件的分解的二进制代码,便于 Java/class 文件的二进制比较,也可以显示当前二进制代码的 ASMifier 代码。读者可以用 Eclipse 更新管理器来安装 ASM。(更新网址:<http://download.forge.objectweb.org/eclipse-update/>)。

(22) Mylyn-任务管理

Mylyn 采用任务为中心的界面,它为开发者提供了任务管理工具。另外,也集成了富文本编辑工具,以及监测工具。

(23) Eclipse Launcher

Eclipse launcher 采用 Delphi 6 写成,使用它读者可以自定义启动 Eclipse IDE 时的配置。

(24) AmaterasUML

Eclipse 的 UML 插件,支持 UML 活动图, class 图, sequence 图, usecase 图等;支持与 Java class/interface 之间的相互导入导出。

(25) Log4E

Log4E 可以帮读者更容易地写日志,而不需要受特定日志框架的约束。当然也可以自定义自己的日志模版。

总而言之,插件能让 Eclipse 用其他语言来编写。Eclipse 插件是必不可少的,因为它们让 Java 应用无缝连接,降低了 Java 的复杂度。然而,装了过多的插件会让读者的 Eclipse 变慢,所以读者需要有选择性地安装插件。

我们简单的介绍其中的 AmaterasUML 插件。

在安装 AmaterasUML 之前,首先要按照 GEF(Graphic Editing Framework)。其官方的文档地址是 http://wiki.eclipse.org/index.php/GEF_Developer_FAQ#Download_and_Install

在线安装地址 GEF -<https://hudson.eclipse.org/hudson/job/gef-maintenance/lastSuccessfulBuild/artifact/update-site/>

在这里,可以通过连接,下载 GEF 的压缩包。下载好之后,将其复制到 Eclipse 的父目录下。然后,将其解压缩。其压缩包中,按照 Eclipse 的目录结构提供了 plugins、features 等目录。因此,如果在 Eclipse 的父目录中进行解压缩,其中的内容会自动放入相应的文件中。

安装好 GEF 之后,再下载 AmaterasUML 的压缩包。其下载地址是:http://amateras.sourceforge.jp/cgi-bin/fswiki_en/wiki.cgi?page=AmaterasUML

解压缩之后,将其 jar 包复制到 Eclipse 目录下的 plugins 文件夹中保存。

重启 Eclipse,然后在 File->New->Other 下面就可以看见 AmaterasUML 的标志了。接着用户便可以导出导入或者制作类图,时序图等。

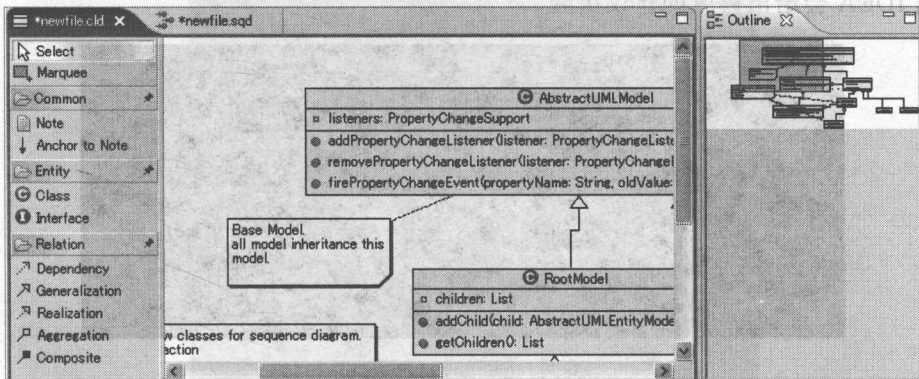


图 1A-9 数据流分析

A.1.4 重构

高效的工程师能够熟练地使用 IDE 提供的重构功能。所有的现代 IDE 都提供许多印象深刻的重构功能。但是还是那句,IDEA 的重构功能也很聪明智能。它们能读懂读者需要什么,然后针对不同的情况提供给读者最适合的解决方案。

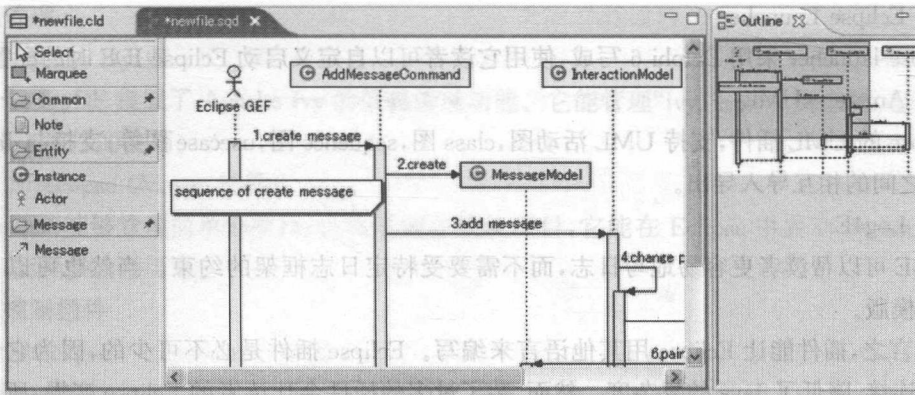


图 1A-10 时序图

例如图 1A-11 中我们可以看到一段设置按钮监听器的代码, 以及其被注释的另一份代码。此时我们想修改 leftButton 为 mBackButton, 此时就可以使用 IDE 的其中一个重构快捷, Shift+F6。

```

@Override
public void onClick(View v) {
    Places_Attractions view = findViewById(
        R.id.button);
    // view.setOnClickListener(new OnClickListener() {
    //     @Override
    //     public void onClick(View v) {
    //         // TODO Auto-generated method stub
    //     }
    // });
}

```

图 1A-11 重构代码举例

首先 IDEA 会给出修改的建议提醒。

```

@Override
public void onClick(View v) {
    Button
    leftButton
    leftButton
    leftButton
    Press shift+F6 to show dialog with more options
}

```

图 1A-12 重构代码举例

当输入完毕并且按 Enter 键后, IDEA 会对整个项目进行扫描, 并且提醒询问所有关联的地方是否进行修改, 甚至是注释了的地方, 这一来就大大得加快了编码的效率了。

单击 Do Refactor, 既可以完成该次重构。

若是一些自定义的方法, 需要添加或者删除某些参数时, 可以使用 Ctrl+F6 进行重构。如图 1A-14 的 fileIsExist(String filePath) 方法, 如果此时需要往该方法添加 String name 的参数, 并且不用每个调用它的地方都重复修改的话, 使用该“重构”的方法是非常快的。

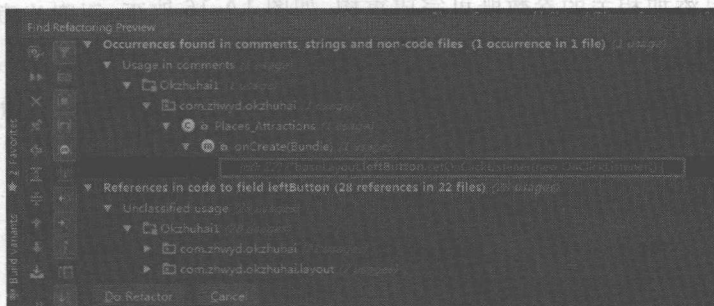


图 1A-13 重构代码提示

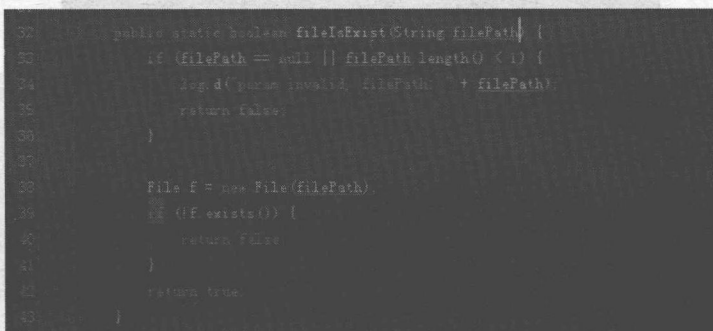


图 1A-14 重构方法参数

鼠标移至该方法上,并按 Ctrl+F6,此时就会弹出下面这样一个窗口,如图 1A-15 所示。



图 1A-15 重构方法参数

此时在窗口里添加相关的参数即可完成重构,如图 1A-16 所示。

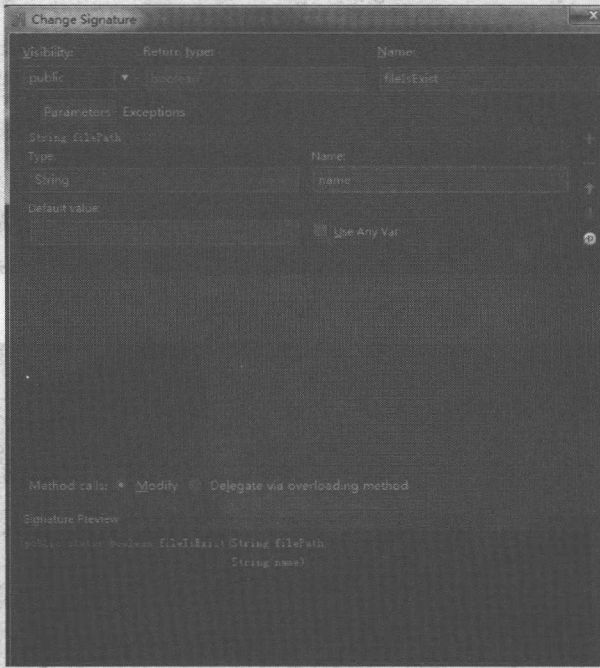


图 1A-16 重构方法参数

就算是重构布局文件中 ID 也是可以的,在这方面 IDEA 就显得比 Eclipse 智能得多,如图 1A-17 所示。

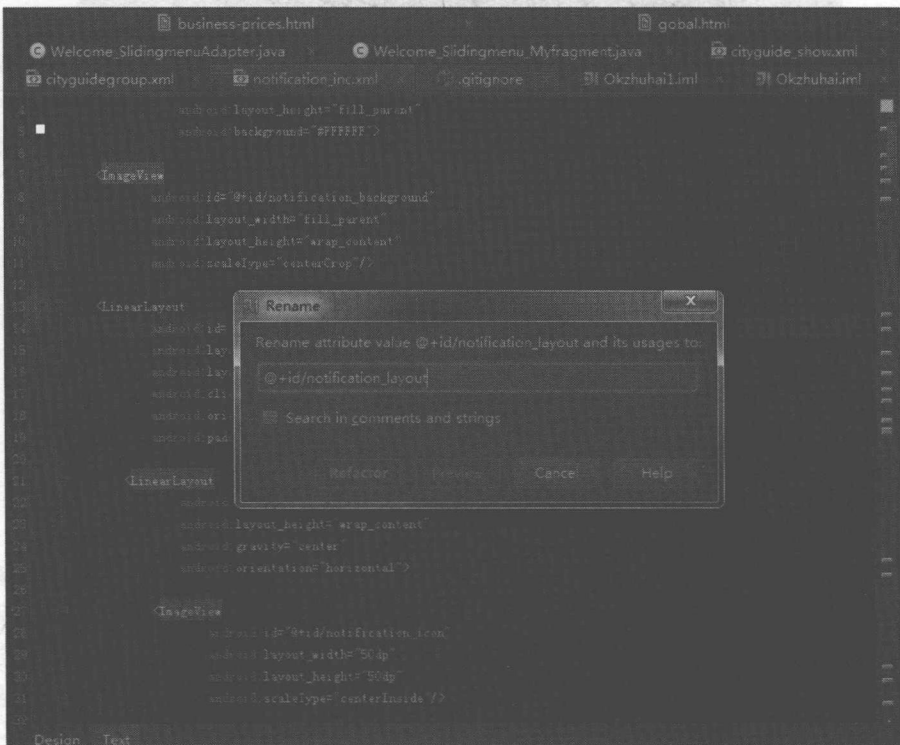


图 1A-17 重构方法参数