



Android



应用开发

隆益民 著

不放过每一个细节

为您层层剖析Android**开发过程**

Android**入门必备!**

全书基于Android**2.1**最新版讲解

中山大学出版社

Android 应用开发

隆益民 著

中山大学出版社

· 广 州 ·

版权所有 翻印必究

图书在版编目(CIP)数据

Android 应用开发/隆益民著. —广州: 中山大学出版社, 2010. 8

ISBN 978 - 7 - 306 - 03706 - 0

I. A… II. 隆… III. 移动通信—携带电话机—应用程序—程序设计 IV. TN929. 53

中国版本图书馆 CIP 数据核字(2010)第 142723 号

出 版 人: 祁 军

策划编辑: 黄龙飞

责任编辑: 李 文 黄龙飞

封面设计: 贾 萌

责任校对: 鲍 磊

责任技编: 何雅涛

出版发行: 中山大学出版社

电 话: 编辑部 020 - 84111996, 84111997, 84113349, 84110779

发行部 020 - 84111998, 84111981, 84111160

地 址: 广州市新港西路 135 号

邮 编: 510275 传 真: 020 - 84036565

网 址: <http://www.zsup.com.cn> E-mail: zdcbs@mail.sysu.edu.cn

印 刷 者: 广州中大印刷有限公司

规 格: 787mm × 1092mm 1/16 11 印张 263 千字

版次印次: 2010 年 8 月第 1 版 2010 年 8 月第 1 次印刷

印 数: 1~1000 册 定 价: 25.00 元

本书如有印装质量问题影响阅读, 请与出版社发行部联系调换

内容简介

本书兼顾不同开发背景的读者，由浅入深，由易到难，一步一步传授如何开发应用。主要内容组织如下：

第一章：Eclipse 入门。

第二章：介绍 Android，包括介绍 Android 是什么，以及 Android 平台的架构以及特性。

第三章：介绍 Android 开发环境的搭建，包括如何下载和安装 Android 开发包、常用 Android 开发工具的使用，并介绍如何构建一个最简单的应用。

第四章：介绍如何创建应用和活动。

第五章：介绍如何创建用户界面。

第六章：介绍数据存储、检索和共享。

为便于读者学习，本书第一章对 Eclipse 做了简介，基础好的读者，可略过第一章，直接从第二章开始学习。

读者对象

本书适用于任何对 Android 平台感兴趣，并希望在 Android 平台上开发移动应用的读者。无论您是一个经验老到的移动应用开发者，还是对此感兴趣但尚无移动应用开发经验的新手，本书都将对您大有裨益。

如果您具有移动应用开发经验，这对您理解本书无疑是有益的。但是，这不是必须的条件，任何人只要有基本的开发经验，并且熟悉 Java 语言，都应能轻松地掌握本书内容。

本书每章内容基本都是独立的，读者可以选择自己感兴趣的章节阅读。

本书可作为移动通信、计算机及相关专业学生教材，也可供相关研究开发人员参考。

前 言

移动开发者激动人心的时刻已经来临。随着移动电话的普及,消费者越来越青睐功能强大的智能手机。时尚的智能手机集成了GPS、加速器、触摸屏等功能,吸引了许多研发人员在智能手机平台上开发新颖的移动应用。

Android手机为吸引消费者而设计,但是赢家却是开发者。现有的智能手机平台几乎都不是开放的平台,这限制了第三方应用的开发。因此,Android作为一个开放的平台,也赢得不少开发者的青睐。

Android平台由谷歌发起,并由全球大多数移动通讯运营商和设备制造商共同创导。它实际上是一个开源框架,具有功能强大的开发类库,这吸引了数以万计的开发者加入到Android应用开发的行列。无论是否有移动应用的开发经验,你都可以很快切入到这个平台,开发出新颖的应用。

Android已经发行了多个版本,影响较大的版本如1.5、2.0、2.1版等。本书将通过实例讲解的方式,阐述如何在Android平台下开发应用软件。书中所有实例均为作者本人开发。

Android虽然是一个全新的平台,但由于它是一个开源、开放的平台,所以,包括中国移动通信集团在内的大多数全球领先的移动运营商和手机制造商都纷纷加入到Android阵营中,台湾手机制造商宏达电2008年9月推出第一款Android手机,并在2009年推出了第二代Android手机(产品代号为Magic)和第三代Android手机(代号为Hero),而三星、华为等手机巨头都纷纷推出自己的Android手机,中国移动推出的OPhone实际上也基于Android平台。

本书作为Android专门著作,目的是希望通过系统介绍Android的基本概念、基础理论、主要技术和软件开发实例,为相关研究人员提供一本有关Android整体的概念、为相关应用开发人员提供一些新的方法和思路的参考著作,并以此抛砖引玉。书中错漏之处,敬请指正。

本书可做为移动通信、计算机及相关专业学生教材,也可供相关研究开发人员参考。

隆益民

2010年6月于广州和平家园

目 录

第 1 章 Eclipse 入门	1
1.1 Eclipse 简介	1
1.1.1 历史背景	1
1.1.2 软件授权	1
1.1.3 版本类型	1
1.1.4 跨语言、跨平台	2
1.2 Eclipse Platform	2
1.2.1 概观	2
1.2.2 架构	2
1.2.3 项目与资料夹	3
1.2.4 平台核心	3
1.2.5 工作区(workspace)	3
1.3 总结	16
第 2 章 Android 介绍	17
2.1 Android 是什么?	17
2.2 Android 特性	17
2.3 Android 系统架构	17
2.3.1 应用(Applications)	18
2.3.2 应用框架(Application Framework)	18
2.3.3 类库	19
2.3.4 Android 运行时(runtime)	19
2.3.5 Linux 内核(Linux Kernel)	19
2.4 Android 开发框架	19
2.4.1 Android APIs 包	20
2.5 总结	23
第 3 章 Android 开发环境	24
3.1 开发平台的系统和软件需求	24
3.2 开发环境的安装与设置	24
3.2.1 下载并安装 JDK 1.6	24
3.2.2 Android SDK 下载和配置	25
3.3 创建第一个 Android 应用	30

3.4 理解 Android 应用	34
3.4.1 应用组件	34
3.4.2 活动组件(intents)	35
3.5 清单文件	36
3.6 Intent 过滤器	37
3.7 活动和任务	38
3.7.1 联亲和新任务	39
3.7.2 运行模式	39
3.7.3 清空栈	40
3.7.4 开始任务	41
3.8 进程和线程	42
3.8.1 进程	42
3.8.2 线程	42
3.8.3 远程过程调用	42
3.8.4 线程安全方法	43
3.9 组件生命周期	44
3.9.1 活动生命周期	44
3.9.2 服务生命周期	48
3.9.3 广播接收器生命周期	50
3.9.4 进程和生命周期	50
3.10 总结	51
第4章 创建应用和活动	52
4.1 清单文件	52
4.2 清单文件结构	52
4.3 文件约定	54
4.3.1 元素	54
4.3.2 属性	54
4.3.3 声明类名	55
4.3.4 多值	55
4.3.5 资源值	56
4.3.6 字符串值	56
4.4 清单文件特性	56
4.4.1 Intent 过滤器	56
4.4.2 图标和标签	57
4.4.3 授权	57
4.4.4 类库	58
4.5 活动属性	58
4.5.1 属性	60

4.5.2 程序基本结构	63
4.6 资源	66
4.6.1 简单值	66
4.6.2 字符串	67
4.6.3 颜色	67
4.6.4 维度	67
4.6.5 风格和主题	68
4.6.6 可绘制图标	69
4.6.7 布局	69
4.6.8 动画	69
第5章 用户界面	71
5.1 用户界面	76
5.2 视图结构	77
5.3 声明布局	77
5.3.1 编写 XML	77
5.3.2 装载 XML 资源	78
5.4 属性	78
5.4.1 ID	79
5.4.2 布局参数	79
5.4.3 布局位置	79
5.4.4 大小及边距	80
5.5 菜单	81
5.5.1 选项菜单	81
5.5.2 上下文菜单	81
5.5.3 子菜单	82
5.5.4 在 XML 中定义菜单	83
5.5.5 菜单特性	84
5.6 对话框	84
5.6.1 显示对话框	85
5.6.2 关闭对话框	87
5.6.3 AlertDialog 对话框	87
5.7 处理用户界面事件	88
5.7.1 事件监听器	89
5.7.2 事件处理器	97
5.8 触摸模式	97
5.8.1 处理焦点	99
5.9 通知用户	99
5.9.1 Toast 通知	99

5.9.2 状态栏通知	100
5.9.3 对话框通知	100
5.10 应用样式和主题	101
5.10.1 样式	101
5.10.2 主题	101
5.11 创建定制组件	101
5.11.1 基本方法	102
5.11.2 完全定制组件	103
5.12 组合控制组件	104
5.13 修改现有的视图类型	104
5.14 绑定数据到 AdapterView	105
5.15 在布局中填充数据	106
5.16 处理用户选择	107
5.17 常用布局对象	107
5.17.1 FrameLayout	108
5.17.2 LinearLayout	109
5.17.3 TableLayout	109
5.17.4 RelativeLayout	109
5.18 视图组总结	109
5.19 记事本例子	110
5.19.1 创建菜单	111
5.19.2 创建事件处理和对话框	112
5.20 总结	112
第6章 数据存储、检索和共享	115
6.1 Android 数据存储技术	118
6.1.1 保存简单应用数据	119
6.1.2 文件存储	122
6.1.3 数据库存储	122
6.1.4 内容提供者(Content Providers)简介	127
6.2 记事本例子	133
6.2.1 保存活动状态	133
6.2.2 保存到 SQLite 数据库	138
6.2.3 显示摘记列表	154
参考文献	164

第 1 章 Eclipse 入门

1.1 Eclipse 简介

1.1.1 历史背景

Eclipse 是由 IBM 耗费 4000 万美元所推出的、具有功能完整的成熟集成开发环境 (Integrated Development Environment, IDE)。第 1.0 版于 2001 年 11 月推出,此后逐渐受到欢迎。Eclipse 业已成为开放源代码项目 (Open Source Project), 大部分的开发掌握在 IBM 手中,但有一部分由 eclipse.org 的软件联盟^①主导。Eclipse 项目目前由 Project Management Committee (PMC) 所管理。综观项目全局, Eclipse 项目可分成三个子项目:

- (1) 平台 Platform。
- (2) 开发工具箱 JDT。
- (3) 外挂开发环境 PDE。

这些子项目又被细分为更多子项目。例如 Platform 子项目包含组件 Compare、Help 与 Search 等,而开发工具箱子项目包括组件 User Interface (UI)、核心 (Core) 及调试 (Debug) 等。外挂开发环境子项目包含组件 UI 与 Core 等。

1.1.2 软件授权

Eclipse 开放源码,开放源码的软件能让使用者取得软件的源代码。Eclipse 使用公共公众许可书 (Common Public License, CPL) 作为授权方式, CPL 可以容许 Eclipse 和其他开放源码软件组织以更严谨的方式散布软件及用于商业途径。CPL 经过 Open Software Initiative (OSI) 认证,内容符合开放源码授权的需求。

1.1.3 版本类型

Eclipse 各版本可以从 eclipse.org 网站的下载入口下载获得^②, 有四种类型的构建版本 (build) 可供下载:

- 释出版 (Release builds): 由 Eclipse 开发团队所确定的主要稳定版本。Release builds 经过完整测试,有定义清楚并具有一致性的功能,定位于商业软件。
- 稳定版 (Stable builds): 比 Release build 新一级的版本,由 Eclipse 开发团队测试,并认定它相当稳定。新功能通常会在此过渡版本出现。定位于商业软件 beta 版。

① <http://www.eclipse.org>

② <http://www.eclipse.org/downloads>

- 整合版(Integration builds): 此版本的各个独立的组件已经过 Eclipse 开发团队认定具有稳定度, 有可能晋级成为 Stable build。
- 当日最新版(Nightly builds): 此版本由最新的源代码编译。

1.1.4 跨语言、跨平台

多数人认为 Eclipse 是 Java IDE, 不过, Eclipse 除了包含 Java IDE(JDT)外, 还有 PDE 等组件。实际上, Eclipse 是一个万用工具平台。JDT 是 Eclipse 的外挂程序之一。Eclipse 更多是指 Eclipse 平台(Eclipse Platform), 除了 Java 开发工具集以外, 还提供各种工具的支持。如要开发 Android 程序, 则需相应的 Eclipse 外挂程序。如果想开发其他语言的程序, 则需要用到其他外挂程序, 诸如 CDT(C Development Toolkit)就可以用于开发 C/C++ 程序。Eclipse 平台跨越了计算机语言, 也跨越了人类的语言。相同的外挂机制可用来增加对不同语言的支持。这里使用一种特殊的外挂, 叫做外挂程序片断(plugin fragment)。IBM 提供一个语言套件, 支持中文(繁体与简体)、法文、德文、意大利文、日文、韩文、葡萄牙文(巴西)与西班牙文等。照理说, Eclipse 以 Java 写成, 应该可以在任何的平台执行。但实际上 Eclipse 并不是跨平台的, 因为它使用作业平台的原生图形来构建。因此要等 SWT(Standard Widget Toolkit)移植到目标平台, Eclipse 才能在该平台执行。但现实中, 这已不是什么问题, 因为 SWT 已经被移植到数个常见平台上了, 包括 Windows、Linux/Motif、Linux/GTK2、Solaris、QNX、AIX、HP-UX 与 Mac OS X。

1.2 Eclipse Platform

Eclipse 平台提供多种软件开发工具的整合机制, 这些工具会作成 Eclipse 外挂程序, 平台必须用外挂程序加以扩充才有用处。Eclipse 设计美妙之处在于除了底层的核心以外, 所有东西都是外挂。这种外挂设计让 Eclipse 具备强大的扩充性, 但更重要的是, Eclipse 提供了一个定义明确的机制, 让各种外挂程序可以彼此合作(通过延伸点 extension points)与协作(contributions), 因此新功能可以轻易且无缝地加入平台。

1.2.1 概观

第一次执行 Eclipse 时, 软件会指定建立一个 workspace 的目录。根据预设, 所有的工作都会存在此目录。若要备份工作目录, 只要备份该目录即可; 若要升级至新版的 Eclipse, 也只要将这个目录拷贝过去即可。启用新版本时要仔细阅读其 release notes, 确保它支持旧版本的 workspace; 若不支持, 只要将旧的 workspace 子目录拷贝到新的 Eclipse 目录下即可。之前的所有的喜好设定都会保留。

1.2.2 架构

Eclipse 平台由多个组件组成: 平台核心(platform kernel)、工作台(workbench)、工作区(workspace)、团队组件(team component)以及说明组件(help)。^[1]如图 1.2.1 所示。

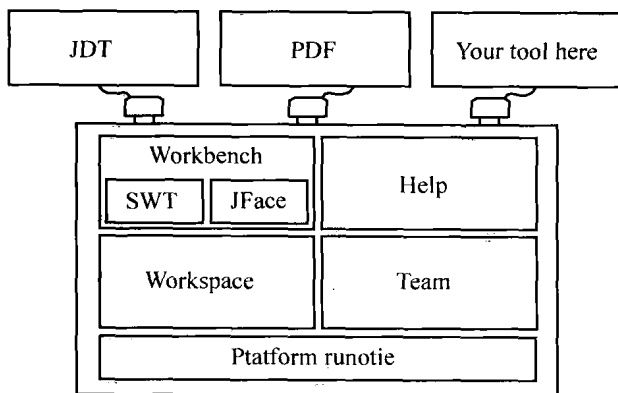


图 1.2.1 Eclipse 架构

1.2.3 项目与资料夹

Eclipse 在最高层级使用“项目”，并在项目之下使用文件夹。

1.2.4 平台核心

核心的任务是程序运转起来，并加载所需之外挂程序。当启动 Eclipse 时，首先执行的就是这个组件，再由其加载其他外挂程序。

1.2.5 工作区(workspace)

工作区负责管理使用者的资源，这些资源会被组织成一个(或多个)项目。每个项目对应到 Eclipse 工作区目录下的一个子目录。每个项目可包含多个文档和文件夹，文件夹也可连到文档系统中的任意目录。每个工作区维护一个历史纪录，记录每个资源的改变。可以立刻复原，回到前一个储存的状态。这个状态可能属于前一天或是几天前，取决于使用者对历史纪录的设定。此历史纪录可以将资源丢失的风险减到最小。工作区也负责通知相关工具有关工作区资源的改变。工具可为项目标记一个项目性质(project nature)，譬如标记为一个“Android 项目”，并可在必要时提供配置项目资源的程序代码。

1.2.5.1 工作台(workbench)

Eclipse 工作台(workbench)界面如图 1.2.2 所示，这是 Eclipse 基本图型接口，工作台是 Eclipse 之中仅次于平台核心最基本的组件^[2]，它是用 Eclipse 标准图形工具箱(Standard Widget Toolkit, SWT)和 JFace(建立在 SWT 之上)的架构。SWT 会使用操作系统的图形支持技术，使得程序的表现(look-and-feel)由操作系统决定。

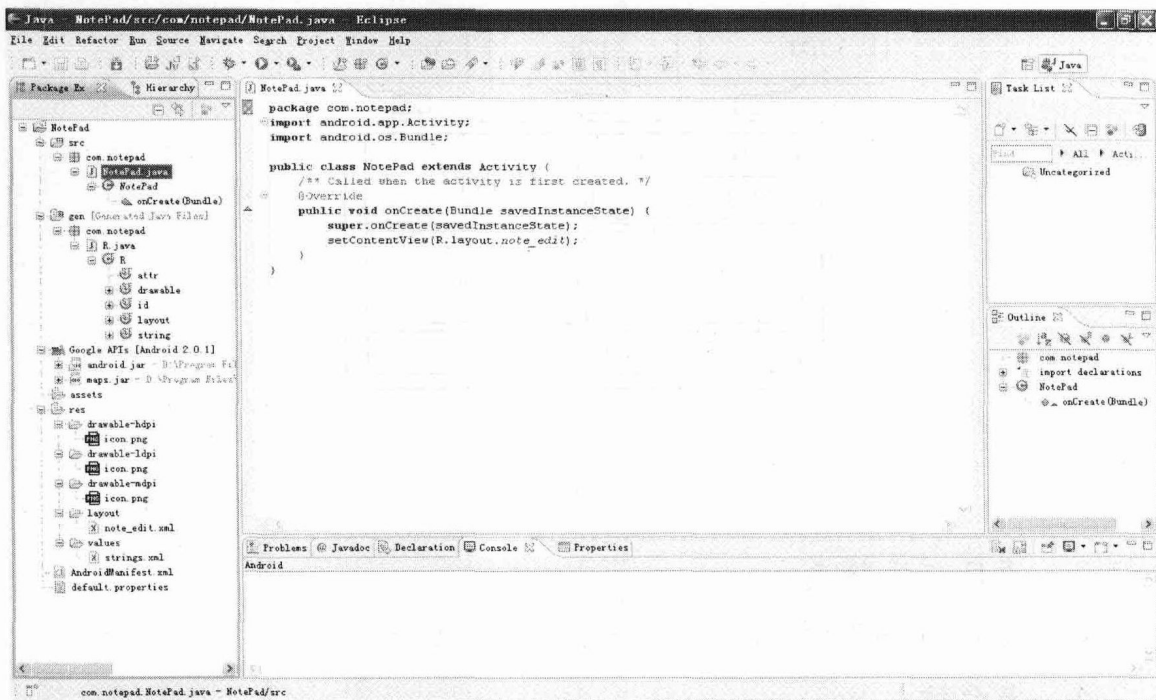


图 1.2.2 Eclipse 工作台

1. 视图

工作台有许多不同种类的内部窗口，称之为视图(view)，以及一个特别的窗口——编辑器(editor)。之所以称为视图，是因为这些窗口以不同的视角来看整个项目。例如图 1.2.2，Outline 的视图可以看项目中 Java 类别的概略状况，而 Navigato 的视图可以导览整个项目。视图支持编辑器。

视图有两个菜单。第一个是用鼠标右键按一下视图卷标来使用的菜单，它可以利用类似工作台窗口相关菜单的相同方式来操作视图；第二个菜单称为“视图下拉菜单”，使用方式是按一下向下箭头。视图下拉菜单所包含的作业通常会套用到视图的全部内容，而不是套用到视图中所显示的特定项目。重设工作台可使用「Window」→「Reset Perspective」菜单作业。重设操作会将设置还原成程序初始状态。可以从「Window」→「Show View」菜单中选取一个视图来显示它。选择「Show View」子菜单底端的「Other...」时，就可以使用其他的视图。

2. 编辑器

编辑器是特殊的窗口，出现在工作台的中央。当打开文件、程序代码或其他资源时，Eclipse 会选择最适当的编辑器打开文件。

3. 透视图

Eclipse 提供多种预先选定的视图，以事先定义好的方式排列，称之为透视图(perspective)。所有透视图的主要组件就是编辑器。每个透视图的目的是执行某特定的工

作，如编写 Android 程序，在 Debug 的透视图中，其中一个视图会显示程序代码，另一个可能换显示变量目前的值，还有一个可能会显示程序的执行结果。

4. 重新排列视图和编辑器

(1) 放置游标

放置光标表示视图可以定置在工作台窗口的位置，可定置在上方、下方等。当重新排列视图时，可能会出现几种不同的放置光标。

(2) 重新排列视图

可以变更「Navigator」视图在工作台窗口中的位置。

- 1) 按一下「Navigator」视图的标题列，并且拖曳视图以横跨工作台窗口。
- 2) 当仍在工作台窗口的顶端拖曳视图时，会出现各种放置光标。
- 3) 将视图定置在工作台窗口中的任何位置，再观察效果。
- 4) 按一下并且拖曳视图的标题列，将视图重新定置在工作台窗口中的其他位置。
- 5) 最后，将「Navigator」视图拖曳到「Outline」视图上面。这时会显示一个堆栈光标。如果放开鼠标按钮，「Navigator」就会和「Outline」视图一起堆放到附加卷标的选项卡中。

(3) 并列编辑器

工作台可以在编辑区中建立两组或更多组编辑器。也可以调整编辑区的大小，但不能将视图拖曳到编辑区。

- 1) 在「Navigator」视图中按两下可编辑的文档，以便在编辑器区中开启至少两个编辑器。
- 2) 按一下编辑器的标示，将它拖曳到编辑器区域之外。
- 3) 如果试图将编辑器放到任何视图的顶端，或放在工作台窗口之外，就会出现限制光标。

4) 观察编辑器标签的颜色：蓝色——表示编辑器目前在作用中。默认颜色在 Windows XP 中呈灰色，表示编辑器是前次作用中的编辑器。

- 5) 拖曳编辑器，将它定置在编辑器区的其他位置。

(4) 重新排列附加标签的视图

除了在工作台中拖放视图之外，也可以在附加卷标的选项卡本内重新排列视图的次序。

- 1) 选择「Window」→「Reset Perspective」，将「Resource」恢复初始状态。
- 2) 按一下「Outline」标题列，然后在「Navigator」视图顶端加以拖曳，堆栈在「Navigator」的顶端。
- 3) 按一下「Navigator」标签，将它拖曳到「Outline」标签的右侧。
- 4) 光标到了「Outline」卷标右侧且变成堆栈光标之后，放开鼠标按钮。此时「Navigator」卷标，在「Outline」标签的右侧。

(5) 最大化、最小化与还原

- ☐ 如果要将视图最大化，从菜单中选取「Maximize」。
- ☐ 如果要将编辑器最小化，从菜单中选取「Minimize」。
- ☐ 将视图还原至原始大小的方法从菜单中选择「Restore」。

1.2.5.2 菜单和工具

1. 菜单

(1)「File」菜单

具有建立、储存、关闭、打印、导入和导出工作台资源以及结束等功能。

- ☐ New(新建)：建立 Android 工程。配置显示在「Window」→「Preferences」的子菜单中。
- ☐ Close(关闭)：关闭运行中的编辑器。如果编辑器中有资料尚未储存，则会显示一个储存要求对话框。
- ☐ Close All(全部关闭)：关闭所有编辑器。如果编辑器中有资料尚未储存，则会显示一个储存要求对话框。
- ☐ Save(储存)：储存正在运行中的编辑器内容。
- ☐ Save As(另存文档)：以新名称储存现有编辑器中的内容。
- ☐ Save All(全部储存)：储存所有编辑器内容以及未储存的变更。
- ☐ Revert(回复)：将现有编辑器的内容回复成已储存文档中的内容。
- ☐ Move(移动)：移动资源。
- ☐ Rename(重新命名)：将资源重新命名。
- ☐ Refresh(刷新)：重新整理所选元素的内容。如果不是从特定选项启动，这个指令会重新整理所有项目。
- ☐ Print(打印)：打印现有编辑器的内容。
- ☐ Switch workspace(切换工作区)：切换至不同的工作区。
- ☐ Open external file(开启外部文档)：文字编辑器中开启不在工作区中的文档。
- ☐ Import(导入)：导入项目。
- ☐ Export(导出)：导出项目。
- ☐ Properties：属性配置。
- ☐ Recent file list：最近使用的文件清单。
- ☐ Exit(退出)：退出 Eclipse。

(2)「Edit」菜单

- ☐ Undo：复原。
- ☐ Redo：重做。
- ☐ Cut：剪切。
- ☐ Copy：复制。
- ☐ Paste：粘贴。
- ☐ Delete：删除。
- ☐ Select All：全选。
- ☐ Find/Replace：查找/替换。
- ☐ Add Bookmark(新增书签)：为目前的文字选项或所选取的元素新增书签。
- ☐ Add Task(新增作业)：为目前的文字选项或所选取的元素新增使用者定义的作业。

(3)「Run」菜单

- ☐ Toggle Breakpoint(切换断点)：在编辑器中所选行新增或移除断点。

- ❑ Toggle Method: 新增或移除方法。
- ❑ Toggle Watchpoint(切换监视点): 新增或移除字段监视点。
- ❑ Skip All Breakpoints: 忽略所有断点。
- ❑ Add Java Exception Breakpoint: 新增 Java 异常状况断点。
- ❑ Add Class Load Breakpoint: 建立一个 Class 断点。
- ❑ Run Last Launched(运行前一次的启动作业): 这个指令可以在运行模式下迅速重复最近一次的启动作业。
- ❑ Debug Last Launched: 调试前一次的启动作业。
- ❑ Run History(运行历程): 最近执行历程的子菜单。
- ❑ Run As(运行): 启动执行快捷方式的子菜单。
- ❑ Run(运行): Debug History(调试历史)在调试模式最近历程的子菜单。
- ❑ Debug As(调试为): 调试启动快捷方式的子菜单。
- ❑ Debug Configuration: 调试配置。
- ❑ Inspect(视察): 当执行绪暂停时, 这个指令会使用「表示式」视图, 显示在该执行绪之堆栈框或变量的环境定义下, 视察所选表达式或变量的结果。
- ❑ Display(显示): 当执行绪暂停时, 这个指令会使用「Display」视图, 显示在该执行绪之堆栈框或变量的环境定义下, 评估所选表示式的结果。
- ❑ Execute(执行): 执行
- ❑ Step into Selection: 这些指令可以单步执行所要调试的程序代码。
- ❑ External Tools: 外部工具

(4)「Source」菜单

- ❑ Toggle Comment(批注): 标注出内含现行选择项的所有字行。
- ❑ Add Block Comment(批注区块): 标注出内含现行选择项的区块。
- ❑ Remove Block Comment(解除批注区块): 取消标注内含现行选择项的区块。
- ❑ Shift Right(向右移位): 增加目前所选字行的内缩层次。只有在选择项涵盖多行或一整行时才会启用。
- ❑ Shift Left(向左移位): 减少目前所选字行的内缩层次。只有在选择项涵盖多行或一整行时才会启用。
- ❑ Format(格式): 可使用程序代码格式制作器, 来设定目前文字选择项的格式。
- ❑ Format Element(格式成员): 格式化成员。
- ❑ Sort Members: 排序成员。
- ❑ Add Import: 新增导入。
- ❑ Override/Implement Methods(重载/实例化方法): 开启「Override Method」对话框, 可以重载或实例化现行类型中的方法。适用于类型或类型中的某个文字选择项。
- ❑ Generate Getter and Setter: (产生 Getter 和 Setter) 开启「Generate Getter and Setter」对话框, 可以为现行类型中的字段, 建立 Getter 和 Setter。适用于字段与类型或类型中的某个文字选择项。
- ❑ Generate Delegate Methods(产生派生方法): 开启「Generate Delegate Methods」对话框,