



基于

Eclipse

环境的J2ME 应用程序开发

万 辉 王 军◎编著



- ◎ **Java ME开发基础** 详细讲解Java ME平台、开发环境的配置和各种无线应用开发包
- ◎ **EclipseME开发基础** 详细介绍在Eclipse下使用EclipseME插件进行Java ME简单开发和图形开发
- ◎ **Java ME游戏设计** 详细介绍用EclipseME进行Java ME手机游戏开发的基础知识，并且给出贪吃蛇、扫雷和比大小三个游戏案例，还介绍MIDP 2.0游戏开发的基础知识和开发示例



清华大学出版社

基于 Eclipse 环境的 J2ME 应用程序开发

万 辉 王 军 编著

清华大学出版社

北 京

内 容 简 介

本书全面介绍在 Eclipse 平台下进行 Java ME 开发的各项最新技术,内容涵盖了基于 Eclipse 环境的 Java ME 开发的各个方面,是一本学习 Java ME 开发技术的手册。书中介绍了 Java ME 的基本概念,并逐一讲解使用 Eclipse 开发 Java ME 应用所需的知识,包括 Eclipse Java ME 环境的建立、无线应用开发包的使用、Eclipse Java ME 工作台、使用 Eclipse 创建 Java ME 简单应用、使用 Eclipse 开发 Java ME 图形应用、Java ME 游戏设计、MIDP 编程、Java ME 通讯开发和使用 CVS 进行团队开发。书中还给出了详细的实例。

本书内容广泛,围绕 Eclipse 下的 Java ME 开发,涉及 Java 平台基础、Java ME 基础、Eclipse 基础、Java ME 应用开发基础、Java ME 游戏开发基础和配置管理等多种知识。

本书适合作为初、中级读者的指导书,也适合作为有一定 Eclipse 和 Java ME 开发经验的读者的进阶书籍,还可以用作各类院校相关专业的教材以及 Java ME 开发培训班的教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

基于 Eclipse 环境的 J2ME 应用程序开发/万辉,王军编著. —北京:清华大学出版社,2009.1
ISBN 978-7-302-18881-0

I. 基… II. ①万… ②王… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 173740 号

责任编辑:邹 杰 宋延清

装帧设计:杨玉兰

责任印制:杨 艳

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:清华大学印刷厂

装 订 者:三河市溧源装订厂

经 销:全国新华书店

开 本:190×260 印 张:25.5 字 数:632 千字

版 次:2009 年 1 月第 1 版 印 次:2009 年 1 月第 1 次印刷

印 数:1~4000

定 价:46.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:025302-01

前 言

为了帮助更多的 Java ME 开发人员学习 Eclipse 下的 Java ME 开发,作者精心编著了本书。本书从 Java ME 基础讲起,逐一介绍使用 Eclipse 下的 Java ME 进行开发所需的知识,包括 Eclipse Java ME 环境的建立、无线应用开发包的使用、Eclipse Java ME 工作台、使用 Eclipse 创建 Java ME 简单应用、使用 Eclipse 开发 Java ME 图形应用、Java ME 游戏设计、MIDP 编程、Java ME 通讯开发和使用 CVS 进行团队开发。

本书具有以下特点。

(1) 图文并茂。书中每介绍一个知识点都会给出一个详细的步骤和示意图,并且提示一些技巧或者一些注意事项。

(2) 结构合理。书中为广大的 Eclipse 和 Java ME 初学者提供 Eclipse 和 Java ME 的基本介绍。没有 Eclipse 和 Java ME 开发经验的读者可以按顺序阅读本书,获得系统和全面的知识。如果已经初步具备了 Eclipse 和 EclipseME 插件使用经验,则可以从第 5 章开始阅读。

(3) 内容新颖。本书在全面介绍 Eclipse 开发平台的基础上着重介绍在 Eclipse 平台下进行 Java ME 开发的最新技术。在 Eclipse 版本的选择上,采用最新的 Eclipse 3.4。

各章的具体内容如下。

第 1 章 介绍 Java ME 的一些基本知识,包括 Java 系统的历史和发展,Java 主要应用技术和特点、Java ME 简介,并且详细介绍 Java ME 配置、Java ME 简表和 Java ME 可选包。

第 2 章 介绍 Eclipse 下的 Java ME 开发环境的安装与配置,包括无线应用开发包的安装、Java ME 开发环境的安装、EclipseME 插件的配置。该章还详细介绍 NetBeans Mobility Pack、Nokia Java ME 和 MOTODEV Studio for Java ME 等其他开发工具。该章最后将介绍如何把 Java ME 程序部署到真实的设备上。

第 3 章 介绍 Java ME 无线应用开发包,包括 Java ME 无线应用开发包的内容、Sun Java Wireless Toolkit 的使用、SonyEricsson Java ME Wireless Toolkit 的使用。最后还将给出使用 Sun Java Wireless Toolkit 和 SonyEricsson Java ME Wireless Toolkit 开发 Java ME 程序的示例。

第 4 章 介绍 Eclipse 下的 Java ME 工作台,包括 Eclipse 工作台、Java 开发环境、Eclipse 团队开发和 Java ME 开发常用的透视图。

第 5 章 介绍如何在 Eclipse 下创建简单的 Java ME 应用程序,包括创建 MIDlet 项目、检验 Eclipse 设置、创建 Java ME 项目、添加 MIDlet 到项目和编辑 Java ME 元素。最后还将介绍如何运行和调试 Java ME 程序、打包 Java ME 程序和发布 Java ME 程序到真机。

第 6 章 介绍如何在 Eclipse 下创建 Java ME 图形应用。该章首先介绍 MIDlet 图形界面编程基础和设计思路,随后详细地介绍高级界面和低级界面的基础和程序实现,最后将介绍布局和常见问题。

第 7 章 介绍手机游戏设计概述,包括手机游戏和传统游戏开发的区别、手机游戏开发的特点、手机游戏开发的基本原则、手机游戏的实现技术。随后介绍小团队开发手机游戏的策略。最后介绍 Eclipse 对手机游戏开发的支持。

第 8 章 介绍利用 Eclipse 开发单机游戏的示例。包括贪吃蛇、扫雷和比大小共 3 个游戏的开发例子。

第 9 章 介绍 MIDP 2.0 游戏编程。该章首先介绍 MIDP 2.0 的新特性、MIDP 2.0 游戏开发概述以及新的游戏 API。接着介绍 MIDP 2.0 的基本开发：主动查询用户按键、使用后台屏幕缓存、使用图层、使用向导。最后介绍一个 MIDP 2.0 开发示例。

第 10 章 介绍 Java ME 通讯开发。该章首先介绍 Java ME 通讯概念和 Java ME 通讯方式等基础知识，并且重点介绍蓝牙通讯，接着给出一个 Java ME 通讯开发示例，最后总结 Java ME 通讯开发常见的问题。

第 11 章 介绍如何在 Eclipse 下使用 CVS 进行 Java ME 团队开发。首先介绍 CVS 的基本概念、CVS 服务器在 Windows 下和 Linux 下的安装和 Eclipse CVS 客户端概览；然后介绍 Eclipse CVS 基本操作(导入项目、检出项目、检入等)；随后介绍 Eclipse CVS 高级操作(创建分支、创建版本、比较、合并、替换等)；最后介绍一些其他操作(修改文件属性、添加文件到忽略列表)。

本书主要适合下列人员阅读和学习：Java ME 开发人员、其他开发平台转向 Eclipse Java ME 开发平台的转型人员、手机游戏开发人员、正在做毕业设计的学生。此外，也可以作为大学/大专/中专的教材，或者社会培训教材。

本书作者分工如下：第 1~4 章和第 11 章由万辉编写，第 5~10 章由王军编写。全书由万辉统稿。

配合本书将提供书中涉及的源代码，下载地址为：

<http://www.wenyuan.com.cn>

<http://www.tup.com.cn/>

本书是作者多年学习和从事项目开发的经验总结，由于作者水平有限，书中的不足之处在所难免，敬请读者指正。



目 录

第 1 章 Java ME 概述.....1

1.1	Java 平台简介.....1
1.1.1	Java 的来源.....1
1.1.2	Java 的发展.....2
1.1.3	Java 发展趋势.....5
1.2	Java 主要应用技术以及特点.....7
1.2.1	Java SE 的主要应用技术.....7
1.2.2	Java EE 的主要应用技术.....9
1.2.3	Java ME 的主要应用技术.....10
1.3	Java ME 简介.....10
1.3.1	J2ME 的基本特性.....10
1.3.2	Java ME 体系架构.....10
1.4	Java ME 配置.....12
1.4.1	CLDC 配置.....12
1.4.2	CDC 配置.....12
1.5	Java ME 简表.....13
1.5.1	KJava 简表.....13
1.5.2	MIDP 简表.....13
1.6	Java ME 可选包.....14

第 2 章 Java ME 开发环境的安装与配置.....16

2.1	无线应用开发包的安装.....16
2.1.1	Java 安装.....16
2.1.2	Sun 无线应用程序包的安装.....16
2.1.3	SonyEricsson 无线应用程序包的安装.....17
2.2	Java ME IDE 开发环境的安装.....17
2.2.1	Java ME 开发工具一览.....17
2.2.2	Eclipse 平台下 Java ME 开发平台的安装.....18
2.2.3	EclipseME 的安装.....18
2.3	EclipseME 插件的配置.....25
2.3.1	设备配置.....25

2.3.2	调试配置.....27
2.3.3	混淆器配置(可选).....28
2.3.4	OTA 设置(可选).....29
2.4	其他 Java ME 专业开发工具的安装与配置.....30
2.4.1	NetBeans Mobility Pack.....30
2.4.2	Nokia Java ME 开发工具.....33
2.4.3	Motorola Java ME 开发工具的安装.....36
2.5	部署 Java ME 程序到设备.....40
2.5.1	硬件准备.....40
2.5.2	软件准备.....40

第 3 章 使用 Java ME 无线开发工具包.....41

3.1	关于 Java ME Wireless Toolkit.....41
3.2	使用 Sun Java Wireless Toolkit.....42
3.2.1	Default Device Selection 菜单命令.....42
3.2.2	Documentation 菜单命令.....43
3.2.3	OTA Provisioning 菜单命令.....43
3.2.4	Preferences 菜单命令.....45
3.2.5	Run MIDP Application 菜单命令.....48
3.2.6	Utilities 菜单命令.....49
3.2.7	Wireless Toolkit 2.5.2 菜单命令.....49
3.3	使用 SonyEricsson Java ME Wireless Toolkit.....50
3.3.1	License 菜单.....50
3.3.2	WTK2 菜单.....50
3.3.3	Connection Proxy 菜单命令.....52
3.3.4	Connection Proxy Setting 菜单命令.....53

3.3.5	Developer World 菜单命令	53
3.3.6	Device Explorer 菜单命令	53
3.3.7	其他菜单命令	53
3.4	使用无线开发工具包示例	54
3.4.1	Sun Java Wireless Toolkit 使用示例	54
3.4.2	Sony Ericsson SDK for Java ME Platform 使用示例	58
第 4 章	Eclipse Java ME 工作台	61
4.1	Eclipse 工作台	61
4.1.1	Eclipse 开发平台简介	61
4.1.2	Eclipse 项目简介	61
4.1.3	Eclipse 界面简介	62
4.2	Java 开发环境	70
4.2.1	JDT 简介	70
4.2.2	JDT 概览	70
4.2.3	JDT 基本功能	71
4.2.4	Java 构建器	75
4.2.5	快速修复	77
4.2.6	文件、项目的导入导出	78
4.2.7	书签功能	80
4.2.8	视图、编辑器的位置设置	81
4.3	Eclipse 团队开发	81
4.3.1	Eclipse 团队开发概述	82
4.3.2	本地历史(Local History)功能	82
4.3.3	代码补丁功能	83
4.3.4	团队同步透视图	84
4.4	Java ME 开发常用的透视图	86
4.4.1	Java 透视图	87
4.4.2	Java 浏览(Java Browsing) 透视图	88
4.4.3	Java 类型等级 (Java Type Hierarchy)透视图	88
4.4.4	调试(Debug)透视图	89
第 5 章	创建简单的 Java ME 应用	91
5.1	创建 MIDlet 项目	91

5.2	检验 Eclipse 设置	92
5.2.1	Eclipse ME 插件的检查	92
5.2.2	Eclipse 的 J2ME 环境配置检查 ..	93
5.3	创建 Java ME 项目	95
5.4	添加 MIDlet 到项目	97
5.5	编辑 Java ME 元素	99
5.6	运行与调试	100
5.7	向真机发布程序	105
5.7.1	IDE 打包	105
5.7.2	WTK 打包	110
5.7.3	ANT 打包	111
5.7.4	向真机发布	120
5.8	常见问题	121

第 6 章	开发带有图形的 Java ME 应用	122
6.1	MIDlet 图形界面编程基础	122
6.2	设计思路概述	123
6.3	Display 类和 Displayable 类	124
6.3.1	Display 类	124
6.3.2	Displayable 类	127
6.4	高级界面的实现	127
6.4.1	TextBox 组件	128
6.4.2	TextBox 组件抽象命令及 事件响应	132
6.4.3	List 组件	132
6.4.4	Form 及其 Item	135
6.4.5	Alert 控件	151
6.5	程序实现	154
6.6	低级界面的绘制	159
6.7	布局	166
6.8	常见问题	167
6.8.1	界面布局的不合理	167
6.8.2	控件属性及用法分不清	168
6.8.3	锚点值的组合模糊	168
6.8.4	一些特殊用法	169

第 7 章	手机游戏设计概述	170
7.1	手机游戏与传统游戏的区别	170

7.2 手机游戏开发的特点	171	9.3.3 Sprite 类	266
7.3 手机游戏设计的基本原则	172	9.3.4 TiledLayer 类	274
7.4 手机游戏的实现技术	173	9.3.5 LayerManager 类	276
7.4.1 图像处理技术	173	9.4 主动查询用户按键	278
7.4.2 按键操作技术	185	9.5 使用后台屏幕缓存	279
7.4.3 I/O 读取写入技术	186	9.6 使用图层开发	284
7.4.4 I/O 存储技术	187	9.7 使用向导	286
7.5 小团队开发手机游戏的策略	195	9.7.1 创建手机应用程序项目	286
7.5.1 提案	196	9.7.2 创建 MIDlet 类	288
7.5.2 设计	196	9.7.3 创建 Canvas 类或者其他 相关类	290
7.5.3 代码	197	9.8 示例开发	292
7.5.4 测试	197		
7.5.5 评审	197		
7.5.6 发布	198		
第 8 章 利用 Eclipse 开发单机 游戏示例	199	第 10 章 Java ME 的通讯开发	297
8.1 贪吃蛇游戏开发	199	10.1 Java ME 通讯概念	297
8.1.1 贪吃蛇游戏概述	199	10.1.1 GCF 通用链接方法	297
8.1.2 Snake 开发架构的搭建	200	10.1.2 GCF 扩展使用	299
8.1.3 SnakeMidlet 类	203	10.1.3 GCF 链接类型	299
8.1.4 SnakeCanvas 类	206	10.2 Java ME 通讯方式	301
8.2 扫雷游戏开发	223	10.2.1 HTTP 通讯	301
8.2.1 MineMidlet 类	224	10.2.2 Socket 通讯	312
8.2.2 MineCanvas 类	225	10.2.3 数据报通讯	314
8.3 比大小游戏开发	249	10.2.4 蓝牙通讯	315
第 9 章 MIDP 2.0 编程	261	10.3 Java ME 通讯示例	328
9.1 MIDP 2.0 的新特性	261	10.3.1 服务器端数据库的搭建	330
9.1.1 运算方面	261	10.3.2 客户端的搭建	334
9.1.2 通讯方面	262	10.3.3 服务器端的搭建	351
9.1.3 游戏方面	262	10.4 Java ME 通讯开发常见的问题	359
9.1.4 多媒体方面	262		
9.1.5 安全机制方面	263	第 11 章 用 CVS 进行 Java ME 团队 开发	360
9.2 MIDP 2.0 游戏开发概述	263	11.1 CVS 的基本概念	360
9.3 新的游戏 API	265	11.1.1 CVS 概述	360
9.3.1 GameCanvas 类	265	11.1.2 CVS 仓库	360
9.3.2 Layer 类	266	11.1.3 CVS 客户端	361
		11.1.4 CVS 基本开发流程	362
		11.2 CVS 服务器的安装和管理	363
		11.2.1 Windows 下 CVSNT 服务器的 安装和管理	363

11.2.2	Linux 下 CVS 服务器的 安装和管理	368	11.4.6	将已有 CVS 项目导入	391
11.3	Eclipse CVS 客户端概述	371	11.5	Eclipse CVS 高级操作	391
11.3.1	CVS 控制台	371	11.5.1	创建分支	391
11.3.2	CVS 仓库浏览透视图	372	11.5.2	创建版本	393
11.4	Eclipse CVS 的基本操作	377	11.5.3	切换分支或者版本	393
11.4.1	导入项目到 CVS 仓库	378	11.5.4	比较	394
11.4.2	模块检出	382	11.5.5	替换	394
11.4.3	添加/删除/修改	388	11.5.6	合并	395
11.4.4	更新	389	11.6	Eclipse CVS 其他操作	395
11.4.5	检入(提交)	389	11.6.1	修改文件文本/二进制属性	395
			11.6.2	把文件添加到忽略列表	397

第 1 章 Java ME 概述

Java ME 是 Java Platform Micro Edition 的简写, Java ME 以往称为 J2ME(Java 2 Platform Micro Edition), 是为机顶盒、移动电话和 PDA 之类嵌入式和移动电子设备提供的 Java 语言平台, 它和 Java SE、Java EE 一起构成 Java 平台并驾齐驱的三驾马车。

Java ME 具备 Java 平台的各种优点: 平台无关、安全、分布式和高度的可移植性。

Java ME 的架构分为配置(Configuration)、简表(Profile)和可选包(Optional Package)。它们的不同组合形成了具体的运行环境。Java ME 的主要配置包括连接设备配置(CDC)、有限连接设备配置(CLDC)。Java ME 简表包括 KJava 简表和移动信息设备简表(MIDP)等。

本章重点

- Java 平台简介
- Java 主要应用技术及特点
- Java ME 简介
- Java ME 配置
- Java ME 简表
- Java ME 可选包

1.1 Java 平台简介

1.1.1 Java 的来源

Java 语言最初来源于 Sun 公司的一个 Green 项目, 该项目的目标是开发一个数字设备的分布式系统, 创建 Java 语言并不是 Green 项目的初衷。Green 项目是由 Patrick Naughton、Mike Sheridan 和 James Gosling 三个人在 1991 年最初发起的。Green 项目后来增加到 13 个人, 在经过 18 个月的封闭式开发之后, 他们开发出了可用的工作演示版本, 称之为 Star7。

Star7 之所以能控制很多的数字娱乐设备和家电设备, 是因为 Green 项目采用了一种全新的跨处理器的编程语言。这个语言是由 James Gosling 创立, 最初的名字叫做“Oak”, 这个名字来源于 James Gosling 窗前的一棵橡树(Oak)。James Gosling 后来被称为 Java 之父。

随着 Star7 项目逐渐取得关注, 并且开始吸引有线电视行业的潜在客户, Green 项目组独立出来, 组建了一个分公司(FirstPerson)。该公司面向 Star7 类型设备、电视机顶盒和视频点播(VOD)行业。Green 项目组开发出了一个名为 MovieWood 的新 DEMO, 但遗憾的是由于当时行业处于幼年期, MovieWood 没有成功中标。

Oak 语言虽然被发明了, 但是一直没有引起人们的重视。直到 1994 年, 随着 WWW(万维网)的快速发展, Sun 公司将其用于 Internet, 并且把 Oak 更名为 Java, 而不再集中于原来的数

字有线电视领域。不久, Green 项目组以 Mosaic 为原型开发了一个名为 WebRunner 的浏览器, 后来官方发布时更名为 Hotjava 浏览器。

说明: ① 为什么要把 Oak 改名为 Java 呢? 当 Green 项目组准备去注册 Oak 时, 发现 Oak 商标已经被另外一家公司注册了, 所以工程师们边喝着咖啡边讨论新的名字。突然有一位工程师灵机一动, 指着 he 喝的咖啡杯中产自印尼 Java 岛的 Java 咖啡, 推荐使用 Java。于是 Java 语言的名称就这样诞生了。

② Mosaic 是第一个可以显示图片的浏览器, 由美国伊利诺州的伊利诺大学的 NCSA 组织在 1993 年所发布。在当时人气爆发的互联网热中大受欢迎。当时的 Mosaic 开发的中心人物 Marc Andreessen 和 Silicon Graphics 与公司的创始人 Jim Clark 设立了 Mosaic Communication Corporation, 这家公司于 1994 年 11 月改名为 Netscape Communication Corporation, 即著名的网景公司。

③ WebRunner 的名称来源于 1982 年拍摄的电影 Blade Runner(中文译为银翼杀手)。现在有一个 Web 自动测试工具也叫 Web Runner, 这二者是没有联系的。

WebRunner 虽然只是一个演示, 但是给人留下了深刻的印象: WebRunner 最先支持动态内容显示。而在当时, 其他 Web 浏览器只能支持静态资料(如文字、表格、图片等)的显示, 尚不能在网页中执行动态程序。

WebRunner 采用的动态程序即 Java Applet 技术。Java Applet 字节量小, 适合在带宽有限的因特网上传输、下载, 但当时多数浏览器尚无法支持和执行 Java Applet, 所以 Sun 公司推出了自己的基于 Java 的浏览器 WebRunner, 以便于推广 Java Applet 技术。

在 1995 年 5 月 23 日的 SunWorld 大会上, Java 1.0 正式发布。Java 发布以后, 凭借其在 Internet 编程方面的独特优势, 获得了很多厂商的青睐, 包括得到 Netscape、IBM、Microsoft、Oracle 等大公司的支持, 使 Java 技术迅速得到推广。

1.1.2 Java 的发展

迄今为止, Java 已经度过了第 12 个年头。Java 的发展大致经历了下面 3 个阶段。

- (1) 婴儿期: 从 1995 年 Java 语言发布到 1998 年 Java 1.2 发布。
- (2) 少年期: 从 1998 年 Java2 发布到 2004 年 JDK5 发布。
- (3) 青年期: 从 2004 年 JDK5 发布到现在。

1. 婴儿期

从 1995 年 Java 语言发布开始。由于其在 Internet 上的独特优势, 引起了各大厂商的兴趣, 纷纷从 Sun 公司取得 Java 许可证以便于开发 Java 应用。

例如 Netscape 公司在 1995 年 8 月取得了 Java 许可证, 并在 1996 年 1 月推出了支持 Java Applet 的 Netscape 浏览器。Oracle、Borland 公司在 1995 年 10 月取得了 Java 许可证。1995 年 12 月 SGI、Adobe、IBM、AT&T、Intel 等公司取得了 Java 许可证。1996 年 3 月 Microsoft 和 NetManage 公司取得了 Java 许可证。1996 年 4 月 SCO、Apple、NEC 等公司取得了 Java 许可

证。1996年5月HP、Sybase公司取得了Java许可证。

虽然Java在1995年5月就发布了,但当时仅仅作为一个语言发布,还没有相应的类库。1996年1月23日,Sun发布了JDK 1.0版,它包括Java运行环境(JRE)和软件开发包(SDK)两个主要部分。JDK 1.0中除了AWT外,其他类库都不完整。

1996年5月29日,第一届JavaOne会议举行。

Sun在1997年2月18日发布了JDK 1.1,相对于JDK 1.0的最大改进就是为JVM增加了JIT(Just In Time)即时编译器。JIT提升了JDK的效率。

1997年4月2日Sun在美国旧金山举行了第二届JavaOne开发应用者大会,超过一万人参加,在当时同类技术会议中创造了纪录。

1997年5月Sun推出JDK 1.1.2。1997年7月Sun推出JDK 1.1.3。

总的来说,从Java语言诞生后的这一个阶段中,Java都处于不成熟期,但是受到业界和很多大型软件企业的关注;Java的类库还十分不完整,主要应用在互联网的Java Applet中和一些移动设备上,离真正的企业应用开发还有相当大的差距。

2. 少年期

从Java发布开始,经过3年多的发展,JDK从最初的1.0版本一直到了JDK 1.18版本。Sun公司也意识到Java应该拥有更广泛的应用领域,而不仅限于Applet和移动设备。

1998年12月4日,Sun推出了Java历史上第一个重要版本JDK 1.2,并同时发布和推出了Java2。Java2平台是Java发展历史上的一个重要里程碑。Java2不仅在性能上有了很大的提高,而且提供了较为完整的API。Sun提出了Java2平台的概念,并且把Java分成了J2ME、J2SE和J2EE。同年的JavaOne会议上,Sun发布了EJB 1.0规范,为Java分布式应用奠定了架构基础,Java开始走向企业级应用。

J2ME是Java2平台微型版(Java2 Platform Micro Edition)的简写,主要面向移动设备、嵌入式设备等的开发。Sun重新拾起了Java语言的前身Oak的主要应用领域——嵌入式移动设备,并在JDK API部分子集和添加一些新的移动设备专用API的基础上,发布了PersonalJava(简写为pJava,是现在J2ME中CDC实现的基础)。但后来发现PersonalJava比较适合大型移动设备(例如机顶盒)的开发,对小型移动设备(例如手机)的支持反而不太好。因此,针对小型移动设备,1999年Sun发布了J2ME规范。2000年9月,Sun发布了专门针对移动通信工具的中IDP开发规范1.0,作为J2ME规范的补充。

J2SE是Java2平台标准版(Java2 Platform Standard Edition)的简写,主要面向个人PC桌面Java应用程序开发。J2SE是Java应用程序开发的主要平台,能够快速开发和进行安全部署。J2SE是Java2平台的一个轻量级应用,J2SE程序可以运行于服务器、桌面系统等大部分操作系统中。

J2EE是Java2平台企业版(Java2 Platform Enterprise Edition)的简写,主要面向复杂的企业级应用。J2EE是分布式企业软件组件架构的规范,具有更高的性能、灵活性、简化的集成性和便捷性。1999年12月12日,Sun发布了J2EE 1.2规范,包括EJB 1.1、JSP 1.1、Java Servlet 2.2、JDBC 2.0等规范。2001年9月24日,Sun发布了J2EE 1.3规范,包括EJB 2.0、JSP 1.2、Java Servlet 2.3、JDBC 2.1等规范。

从 1998 年 JDK 1.2 发布之后,差不多每隔 2 年就发布一个新的 JDK 版本。2000 年 5 月 8 日 JDK 1.3 发布。Sun 在 JDK 1.3 中进行了大量的改进,包括添加类库、在 JNDI 中添加 DNS 支持、增加 JNI 的支持等。2002 年 2 月 13 日 Sun 发布了 JDK 1.4,性能有了质的飞跃,成为 JDK 历史上最成熟的版本。现有的 JDK 应用多数是建立在 JDK 1.4 基础上的。

3. 青年期

虽然 JDK 1.4 在性能上有了质的飞跃,但是 Java2 平台变得越来越复杂,越来越难以使用。虽然 Java 语言是一个纯面向对象的语言,但是有些高级特性(例如泛型、增强 for 循环)却不具备。所以 Sun 加强了对这些易用性问题的处理,2004 年 10 月, Sun 发布了 JDK 1.5(英文代号为 Tiger),同时将版本 1.5 改为 5。JDK 1.5 中添加了泛型、增强的 for 语句、自动拆包和解包、注释等特性。并且同时把 J2ME、J2SE 和 J2EE 更名为 Java ME、Java SE 和 Java EE。在发布 Java5 的同时,发布了 EJB 3.0 规范。

Java5 的发布标志着 Java 进入了青年期。

2006 年 11 月 13 日, Sun 公司宣布开放 Java 虚拟机编译器和 Java 的源代码,采用 GPLv2 授权协议。

2006 年 12 月, Sun 发布了代号为 Mustang 的 Java SE 6。Java SE 6 有下列新特性:

- Web Service 支持。Java SE 6 中添加了新的 API,支持 Web Service 开发。
- 增强的脚本功能。Java SE 6 中已经实现了 JSR 223(Scripting for the Java Platform, <http://jcp.org/en/jsr/detail?id=223>), JSR 223 允许 Java 代码在动态运行时通过脚本引擎去调用脚本语言。JSR 223 同时也提供在脚本语言中访问 Java 对象的 API。JSR 223 实际上提供一个 Java 和脚本语言的桥,允许 Java 和脚本语言进行互操作。目前 Java SE 6 中提供了 JavaScript 的默认实现。Sun 在 2006 年 9 月份招聘了 JRuby 的开发人员,最可能的 Java 支持的下一个脚本语言是 Ruby。同时 Sun 也在吸引 Perl 和 PHP 等其他脚本语言的开发人员。
- 纯 Java 数据库(Java DB)。Java SE 6 中集成了 100%兼容 JDBC 的基于 Apache Derby 的 Java 数据库 Java DB。这样用户在开发 JDBC 应用时,不需要到处去寻找和配置一个自己的 JDBC 数据库。J2SE 6 同时发布了 JDBC 4.0, JDBC 4.0 中包含多项重要的性能增强,例如提供对 XML、BLOB、CLOB 等数据类型的更好支持。
- 更丰富的桌面 API。开发者可以更简单地开发更强大的桌面应用,例如启动界面的支持、系统托盘的支持,整合了 SwingWorker,用于处理 GUI 中的多线程、JTable 排序和过滤等。
- 监控和管理。Java SE 6 添加了更多的诊断信息,例如当遇到 java.lang.OutOfMemory 异常的时候,可以得到一个完整的堆栈信息,并且当堆已经满了的时候,会产生一个 Log 文件来记录这个致命的错误。Java SE 6 中还整合了 jhat 内存分析工具。
- 编译器访问。Java SE 6 实现了 JSR 199,提供在 Java 语言中访问 Java 编译器的 API。该功能对于那些需要访问 Java 编译器的应用(例如 JSP 开发工具)来说,是十分有意义的。
- 可扩展的标注。Java SE 5 中引入了标注(Annotations)功能,Java SE 6 则允许用户定义自己的标注。

1.1.3 Java 发展趋势

Java 发展的趋势可以从 Sun 公司的新闻和每年在旧金山举办的 JavaOne 会议中看出来。

1. 可用性

Java 之父 James Gosling 在 2007 年 4 月接受 Computer Weekly 采访时说到, Java 是一个通用的编程开发语言, 但是使用 Java 开发 Web 应用却像一门艺术。Java 语言发展的趋势就是像 Macromedia 的 Flash(Macromedia 已经被 Adobe 收购)技术一样使用简便, Flash 提供比 Java 相对来说简单得多的开发环境。

Java5 中出现的注解/泛型、Java6 中出现的增强的脚本功能等都体现了 Sun 公司在提升 Java 语言的可用性上做出的努力。

Java EE 5 中对 Web Service 的支持也反映出 Sun 公司提升 Java 语言可用性方面的努力。Java EE 6 草案中也提出了对 Web Service 进行增强的计划。

Java 在可用性方面的未来发展趋势包括:

- 提供更灵活的 Java 语言。
- 为 Web 开发提供更多的支持, 包括支持更多的脚本语言、对 Web 2.0 的支持。
- 对 SOA 架构的更多支持。

2. 扩展性

Java EE 规范变得越来越庞大, 但是开发者并不需要所有的 Java EE 技术。未来可以把 Java EE 里面的技术作为一个可选的插件, 由开发者来动态添加。这样可以使 Java EE 平台变得更加简洁易用。

3. 开源化

说到开源, 不得不提起 Unix、GNU 和 Linux 这 3 个操作系统。

(1) Unix 操作系统

1965 年, 贝尔实验室、通用电气和麻省理工共同开发了操作系统 Multics(MULTiplexed Information and Computing System)。Multics 是一个分时计算机系统, 目标是支持连接 1000 部终端机并支持 300 位使用者同时上线, 这是 Unix 的前身。Multics 是计算机领域的一个非常大的成就, 但是运行缓慢并且对计算机要求非常高, 用户需要购买昂贵的计算机来运行 Multics。

在 Multics 项目进行 4 年之后, 由于仍然未达到原先的设计目标, 项目被撤消了。Multics 的后期主要开发者 Ken Thompson、Dennis Ritchie、Joe Ossanna 等人对 Multics 项目的撤消心痛不已。Thompson 等人决定继续开发一个新的操作系统, 来满足他们的编程需要。在他们请求新的计算机被拒绝之后, Thompson 等人找到一个 PDP-7 来继续实现他们的开发新操作系统的计划。新的系统最初受 Multics 的影响, 取名为 UNICS(UNiplexed Information and Computing System), 后来根据 UNICS 的谐音改为 Unix。1973 年用 C 语言重写全部 Unix。由于反托拉斯法对贝尔实验室所属的美国电报电话公司的限制(不能销售任何计算机产品), 所以 Unix 在美国电报电话公司并没有受到足够的重视, Unix 仅仅在内部工程师中使用。1975 年秋, Ken

Thompson 决定返回他的母校加州大学伯克利分校任教,期间他在 Unix 上开发了一个 Pascal 编译器。由于很多人对他编写的 Pascal 编译器很感兴趣, Ken Thompson 的学生 Bill Joy 开始使用 BSD(Berkeley Software Distribution)方式发布他们的开源 Unix 版本,允许已经从美国电报电话公司获得 Unix 许可的人支付少量名义上的费用就可以得到他们发行的 Unix 版本。1987 年美国电报电话公司联合 Sun 公司,意欲整合 BSD 和 System V,一统 Unix 天下。这个联合遭到了其他 Unix 成员例如 IBM、DEC、HP 等的强力抵制。为了抵制这个联合,抵制联盟成立了开放软件基金会(Open Software Foundation, OSF)。从此 Unix 大战开始,在此之后多种私有的或者开源的 Unix 版本相继被发布。在 Unix 大战之后 BSD 主要划分为 3 个版本:简单/稳定/易用的 FreeBSD、易于移植和适合研究的 NetBSD 和高度安全/稳定的 OpenBSD。

(2) GNU 操作系统

1971 年,作为开放源码的先驱,哈佛大学毕业毕业生 Richard Stallman 加入了麻省理工学院的人工智能实验室,主要致力于一个名为 ITS 的操作系统的发展。随后到了 20 世纪 80 年代早期,由于计算机体系结构的进步,原先的 ITS 系统已经过时,项目组也随之被解散。不像其他人都选择到商业软件公司任职, Richard Stallman 创建了 GNU 项目。GNU 是一个免费的操作系统、应用程序以及开发工具的集合,兼容 Unix。更重要的是,GNU 中采用了 General Public License(GPL)。1985 年 Stallman 建立了自由软件基金会(Free Software Foundation, FSF),用于支持在 GNU 上开发自由软件。Stallman 也是 GCC、GDB、GNU Emacs 等软件的贡献者。

(3) Linux 操作系统

1987 年, Andrew Tanenbaum 教授开发了 Minix。Minix 是一个用于教学目的的 Unix 的简化版本。Minix 并不是一个复杂的操作系统,但是它的 12000 行 C 代码和汇编源代码都是免费的,因此吸引了很多程序员的关注。1991 年 8 月, Linus Torvalds 在 Minix 的用户组上宣布开始开发一个新的免费的操作系统,实际上是开发一个新的操作系统的核心。1992 年完整的 Linux 核心和不完整的 GNU 操作系统捆绑在一起发行了。从 0.12 版本开始, Linux 开始使用 GPL 许可。

开源是从 Unix、GNU 和 Linux 这 3 个操作系统开始的,也可以说开源起源于操作系统。这 3 种操作系统主要使用 C 语言开发,所以开源运动最初主要集中于 C 语言。后来随着这 3 种操作系统的不断发展,越来越多的开源项目采用 C++。最近几年来,采用 Java 开发的开源项目数量开始超过 C 和 C++。

Sun 公司在 2006 年宣布 Java 开源之前对 Java 开源一直比较谨慎。他们主要担心如果 Java 实现开源了,可能会出现很多不兼容的 Java 虚拟机或者类库,违背了 Java 语言“一次编写,到处运行”的初衷。但是在业界(特别是 IBM)对 Java 开源的呼声非常大。Sun 虽然发明了 Java,但是 IBM 为 Java 做出了最大的贡献并且从 Java 中获益最大。2006 年 11 月 13 日 Sun 公司宣布 Java 实现采用 GPL2 许可证方式开源, IBM 同时发布了一篇声明,表示“IBM 建议 Java 走一条 Apache 之路,应当将 Java 的控制权交给 Apache 软件基金会”。

Java 的开源采用了 Linux 下常使用的 GPL2,这样会吸引更多的 Linux 开发者使用 Java 语言来开发开源应用,而且在许可方面可以自然过渡。另外一方面,由于 Java 实现采用了 GPL2,所以对不兼容的 Java 实现的担心也降低到比较低的程度。

未来的趋势是 Java 的开源应用会越来越多,开源应用也会反过来促进 Java 的进一步发展。

4. Linux 化

在 Sun 公司没有宣布 Java 实现开源之前, 由于 JDK 的许可问题, 很多 Linux 中并没有集成 JDK。这样也造成了 Linux 下的 Java 应用相对来说比较少。Linux 社区对开发 Linux 下的 Java 应用热情不高。

随着 Sun 公司宣布采用 GPL2 方式发布 Java 之后, Linux 将在 GPL 许可下把 Java 开发工具包(JDK)作为免费软件发布, 使 JDK 成为其开源库的内容之一。这有助于 Linux 下的 Java 应用的蓬勃发展。Java 的发展也会因为 Linux 下 Java 应用的普及而产生一些变化。

1.2 Java 主要应用技术以及特点

Java 的主要应用技术集中在桌面应用(Java SE)、Web 应用(Java EE)和移动设备(Java ME)等方面。

1.2.1 Java SE 的主要应用技术

Java SE 主要面向个人 PC、桌面应用和服务器应用程序。Java SE 6 加入了原本属于 Web 应用的 Web Service。

Java SE 中有两种主要产品:

- Java 运行环境(Java Runtime Environment, JRE)——JRE 提供了运行 Java 程序的基本类库、Java 虚拟机和运行 Applet 时需要的组件。JRE 有两种主要部署方式: Java Plugin 和 Java Web Start。Java Plugin 实质上是一个浏览器插件, 它可以使浏览器具备执行 Applet 的能力; Java Web Start 则可以部署单独的 Java 应用程序。这两种部署方式也是 Java EE 开发和部署的基础。JRE 中不包括 Java 编译器或者调试器。
- Java 开发工具包(Java Development Kit, JDK)——JDK 是 JRE 的扩展集, 除了含 JRE 之外, 还包括 Java 编译器和 Java 调试器。

Java SE API 中主要包括核心组件、桌面组件和其他组件等技术。

1. 核心组件

核心组件提供编写 Java 应用程序的基本功能, 例如数据库访问、安全、远程方法调用(RMI)等。核心组件包括如下几个组件:

- 基本组件。基本组件提供了 Java 平台的基本特征和基本功能, 提供了 Java 的基类 Object、基本类型的包装类(例如 Integer)、Math 类等。
- CORBA 组件。从 JDK 1.2 起, Java 提供对 CORBA 的支持。CORBA 是公共对象请求代理体系结构(Common Object Request Broker Architecture)的简写, 是由 OMG 组织制订的一种标准的面向对象应用程序体系规范。Java 提供了 CORBA 的实现, Java IDL 使得 Java 应用程序可以提供基于行业标准 OMG IDL 和 IIOP 的远程服务。除了 Java IDL 之外, Java SE 也支持 IIOP 上的远程方法调用(RMI-IIOP), 这样 Java 开发者不需要了解 IDL 就可以编写 CORBA 应用程序。

说明：分布式互操作标准主要有 Microsoft 的 COM/DCOM 标准、Sun 公司的 Java RMI 标准和 OMG(对象管理组织)的 CORBA 标准。

CORBA 是 OMG 于 1991 年提出的基于对象技术的分布计算应用软件体系结构。CORBA 标准主要分为三个部分：接口定义语言(Interface Definition Language, IDL)、对象请求代理(Object Request Broker, ORB)，以及 ORB 之间的互操作协议 IIOP，核心是对象请求代理。

- Java 命名和目录接口(Java Naming and Directory Interface, JNDI)组件。JNDI 提供了统一的命名和目录服务接口，在 Intranet 和 Internet 上提供共享服务信息(例如用户、机器、网络等)。JNDI 也用在 Java EE 中，用于在分布式环境中查找和定位组件。
- 监控和管理组件。从 Java5 开始，JDK 中就提供监控和管理组件，包括 Java VM 工具、监控和管理应用程序 API、监控和管理工具和 Java 管理扩展(JMX)。
- 工具组件。Java SE 中还包括一些工具，如 Javadoc、Java 本地接口(JNI)、调试接口等。
- XML 组件。Java SE 支持 W3C DOM 的 API、XML 解析和用 XSLT 进行 XML 转换。

2. 桌面组件

桌面组件可以提供富客户端的体验，例如 Java Plugin、组件模型 JavaBeans 和图形用户接口。桌面组件包括下列部分：

- 易用性工具组件(Accessibility)。Java 的易用性工具组件是基于美国联邦 1973 年发布的 Section 508 法案。Section 508 为 1973 年颁布的劳动康复法的补充条款，适用于联邦政府开发或购买的电子产品和信息技术。这项法案要求电子信息技术应能让残障人士使用。
- Java Plugin。Java Plugin 使得 Applet 或者 Java Bean 在 Web 浏览器中运行变得可能，所以 Java Plugin 实质上是给主流的浏览器(Mozilla 或者 IE)提供了一个浏览器插件，来支持 Java 程序在浏览器中的运行。
- JavaBeans 组件。JavaBean 是一种 Java 语言写成的可重用组件。JavaBean 要求类必须是具体的和公共的，并且具有无参数的构造器。JavaBean 中定义了很多属性和相应的 get 和 set 方法。JavaBean 贯彻了 Java 平台的“一次编写，随处运行”的宗旨，因为 JavaBean 只有 set 和 get 方法，不会和与平台相关的 ActiveX 组件交互。
- AWT 组件。AWT 是抽象窗口工具包(Abstract Window Toolkit)的简写。AWT 是 Java 平台独有的窗口系统、图形和用户接口组件工具包。AWT 是 Java 基础类(JFC)的一部分，是为 Java 程序提供图形用户接口(GUI)的标准 API。Java 1.0 中的 AWT 1.0 缺点很多，例如在不同平台下的表现不一样。Java 1.1 之后的 AWT 与 Java 1.0 中的 AWT 有很多的不同，克服了 Java 1.0 中 AWT 的很多缺点。Java2 发布之后，Swing 逐步取代 AWT，成为 Java 桌面 GUI 开发工具。现在 AWT 主要应用在 Applet 开发上。
- JavaHelp 组件。JavaHelp 是一个开源软件，是一个特征完整、可扩展、平台独立的帮助系统，可使开发人员将在线帮助融合到 Applet、应用程序、操作系统和设备之中。用户可以使用 JavaHelp 在 Web 或者企业内部网上传送在线文档。