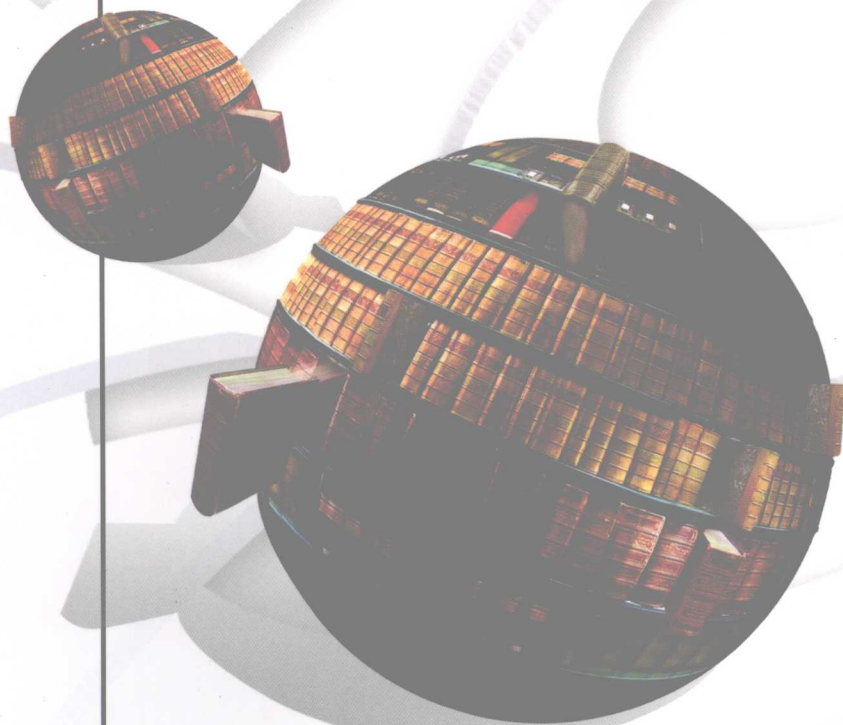


J2ME

程序设计实例教程

杨光 孙丹 编著



清华大学出版社

本书以实例为主线，全面介绍J2ME的编程设计。全书共分12章，第1章介绍J2ME的编程设计，第2章介绍J2ME的编程设计，第3章介绍J2ME的编程设计，第4章介绍J2ME的编程设计，第5章介绍J2ME的编程设计，第6章介绍J2ME的编程设计，第7章介绍J2ME的编程设计，第8章介绍J2ME的编程设计，第9章介绍J2ME的编程设计，第10章介绍J2ME的编程设计，第11章介绍J2ME的编程设计，第12章介绍J2ME的编程设计。本书可作为高等院校计算机专业及相关专业的教材，也可供从事J2ME开发的工程技术人员参考。

J2ME 程序设计实例教程

杨 光 孙 丹 编著

清华大学出版社 010-62770173 010-62770173 010-62770173 010-62770173 010-62770173 010-62770173 010-62770173 010-62770173 010-62770173 010-62770173

图 1-1-1	图 1-1-1
图 1-1-2	图 1-1-2
图 1-1-3	图 1-1-3
图 1-1-4	图 1-1-4
图 1-1-5	图 1-1-5
图 1-1-6	图 1-1-6
图 1-1-7	图 1-1-7
图 1-1-8	图 1-1-8
图 1-1-9	图 1-1-9
图 1-1-10	图 1-1-10

清华大学出版社	清华大学出版社
地址：北京清华大学学研大厦A座	地址：北京清华大学学研大厦A座
邮编：100084	邮编：100084
电话：(010)62770173	电话：(010)62770173
网址：http://www.tup.com.cn	网址：http://www.tup.com.cn
电子邮箱：tup@tup.com.cn	电子邮箱：tup@tup.com.cn

清华大学出版社

北 京

本书可作为高等院校计算机专业及相关专业的教材，也可供从事J2ME开发的工程技术人员参考。

内 容 简 介

本书以实例为主,介绍 J2ME 程序设计。全书共 12 章,首先介绍 J2ME 的基础知识及 MIDP 应用程序的开发过程;然后通过 11 个具体的移动应用程序设计实例,分别介绍 MIDP 图形用户界面、记录管理系统、无线网络编程和游戏开发等重要内容以及 PDA 可选 API、无线消息 API、移动媒体 API 和 Java 蓝牙无线技术 API 等可选包的功能。

每个程序设计实例都给出了详细的讲解,包括知识点、设计要求、类设计和代码实现,最后针对具体的实例给出了实践提示。

本书适合具有 Java 语言基础的读者使用,可以作为高等院校 J2ME 移动应用开发课程的教材,也可作为移动应用开发人员的参考书。

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

J2ME 程序设计实例教程 / 杨光, 孙丹编著. —北京: 清华大学出版社, 2008.10
ISBN 978-7-302-18163-7

I. J… II. ①杨… ②孙… III. Java 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 104778 号

责任编辑: 魏江江 赵晓宁

责任校对: 焦丽丽

责任印制: 王秀菊

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: 北京密云胶印厂

装 订 者: 三河市漂源装订厂

经 销: 全国新华书店

开 本: 185×260 印 张: 21.75 字 数: 530 千字

版 次: 2008 年 10 月第 1 版 印 次: 2008 年 10 月第 1 次印刷

印 数: 1~4000

定 价: 32.00 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: 029815-01

FOREWORD

前言

随着网络和无线通信技术的发展,以及手机等无线移动设备的处理能力不断增强,手机已经不再是一个单一的通信设备,它融合了上网、个人信息管理、移动办公、娱乐游戏等功能。各种无线移动通信终端正朝着多功能、智能化的方向发展。并且随着 3G 网络试运行,数据通信费用大大降低,各种针对移动技术的应用需求会越来越大。

J2ME 是 Sun 公司提供的移动应用开发平台。自从 Sun 公司发布 J2ME 以来, J2ME 技术便引起了软件开发商、信息服务商的极大关注,超过 500 家公司签定了使用 J2ME 的协议。一些主要的移动设备制造商,如诺基亚、西门子、三星、摩托罗拉等公司都推出了支持 Java 技术的手机。现在,越来越多的人意识到了 J2ME 技术的开发与应用所带来的无限机遇。本书主要面向那些有一定 Java 基础的开发人员和高校学生。

本书共 12 章,以具体的移动应用实例为框架,讲解了 J2ME 编程的各部分知识。书中每个实例都给出了详细的解释,包括知识点、设计要求、类设计和代码实现,代码都附有详细的注释,便于读者阅读。在每个实例的最后都给出了一些实践作业,读者可以根据要求修改实例程序,或者设计一个软件。通过实践,增强了读者对所学知识的理解与掌握,也提高了读者的学习兴趣。本书注重实践,同时也有理论叙述。

第 1~第 3 章是全书的基础,其余章节独立性很强,读者可以根据兴趣有选择性的学习。

本书第 1 章由孙丹编写,其余各章由杨光编写。书中的全部代码均在 JDK1.5+J2ME WTK2.2 环境下调试通过,并在 J2ME WTK 自带的模拟器上能够正确运行。本书代码仅供学习 J2ME MIDP 编程使用,欢迎读者对不当之处提出批评建议。

编 者

2008 年 5 月

CONTENTS

目 录

第 1 章 J2ME 入门	1
1.1 什么是 J2ME	1
1.2 J2ME 体系结构	2
1.2.1 配置层	2
1.2.2 简表层	3
1.2.3 可选包层	4
1.3 J2ME 无线工具包	4
1.3.1 安装 J2ME 无线工具包	4
1.3.2 使用无线工具包开发 MIDP 应用程序	5
1.4 MIDP 应用程序	9
1.4.1 MIDlet 的生命周期及其状态	9
1.4.2 MIDlet 套件	17
1.4.3 安装 MIDP 应用程序	20
第 2 章 电子课程表	24
2.1 知识点	24
2.1.1 图形用户界面组件简介	24
2.1.2 高级组件使用	25
2.1.3 记录管理系统	36
2.2 设计要求	41
2.3 类设计	41
2.3.1 课程表数据类设计	41
2.3.2 用户界面设计	42
2.4 代码实现	45
2.4.1 程序打包及运行效果	45
2.4.2 实现与代码分析	46
2.5 实践	75
第 3 章 计算器	76
3.1 知识点	76
3.1.1 CustomItem 简介	76
3.1.2 Graphics 简介	77
3.2 设计要求	78
3.3 类设计	79
3.4 代码实现	80
3.4.1 程序打包及运行效果	80

3.4.2 实现与代码分析	81
3.5 实践	90
第4章 城市公交线路查询系统	91
4.1 知识点	91
4.1.1 HTTP 协议简介	92
4.1.2 Java Servlet 简介	93
4.1.3 HttpURLConnection	94
4.2 设计要求	96
4.3 类设计	96
4.3.1 服务器端	96
4.3.2 客户端	96
4.4 代码实现	97
4.4.1 安装 Tomcat 服务器和部署服务器端程序	97
4.4.2 客户端程序打包及运行效果	99
4.4.3 实现与代码分析	100
4.5 实践	110
第5章 移动电子邮件	111
5.1 知识点	111
5.1.1 套接字连接	111
5.1.2 简单邮件传输协议	120
5.2 设计要求	122
5.3 类设计	122
5.4 代码实现	123
5.4.1 程序打包及运行效果	123
5.4.2 实现与代码分析	125
5.5 实践	139
第6章 俄罗斯方块	140
6.1 知识点	140
6.1.1 Canvas 简介	140
6.1.2 图形旋转	142
6.2 设计要求	142
6.3 类设计	143
6.4 代码实现	144
6.4.1 程序打包及运行效果	144
6.4.2 实现与代码分析	145
6.5 实践	158

第7章 推箱子	159
7.1 知识点	159
7.1.1 GAME API 简介	159
7.1.2 GAME API 使用	159
7.1.3 定时器简介	177
7.2 设计要求	178
7.3 类设计	178
7.4 代码实现	180
7.4.1 程序打包及运行效果	180
7.4.2 实现与代码分析	181
7.5 实践	199
第8章 文件管理器	200
8.1 知识点	200
8.1.1 FileConnection 简介	200
8.1.2 FileConnection API 使用	201
8.2 设计要求	207
8.3 类设计	207
8.3.1 FileManagerMIDlet 类	208
8.3.2 FileManager 类	208
8.3.3 视图类	209
8.4 代码实现	210
8.4.1 程序打包及运行效果	210
8.4.2 实现与代码分析	211
8.5 实践	232
第9章 名片管理器	233
9.1 知识点	233
9.1.1 PIM 简介	233
9.1.2 PIM API 使用	233
9.2 设计要求	236
9.3 类设计	237
9.3.1 CardcaseMIDlet 类和 CardcaseMenu 类	237
9.3.2 Cardcase 类	238
9.3.3 视图-控制器类	238
9.4 代码实现	239
9.4.1 程序打包及运行效果	239
9.4.2 实现与代码分析	241
9.5 实践	274

第 10 章 短信息收发软件	275
10.1 知识点	275
10.1.1 无线消息 API 简介	275
10.1.2 使用无线消息 API	278
10.1.3 PUSH 注册	280
10.2 设计要求	281
10.3 类设计	281
10.4 代码实现	282
10.4.1 程序打包及运行效果	282
10.4.2 实现与代码分析	284
10.5 实践	293
第 11 章 移动视频播放器	294
11.1 知识点	294
11.1.1 MMAPAPI 框架简介	294
11.1.2 使用 MMAPAPI	295
11.2 设计要求	298
11.3 类设计	298
11.3.1 控制器类 MobilPlayerMIDlet	299
11.3.2 播放器类 VedioPlayer	299
11.4 代码实现	299
11.4.1 程序打包及运行效果	299
11.4.2 实现与代码分析	300
11.5 实践	310
第 12 章 五子连珠	311
12.1 知识点	311
12.1.1 蓝牙 API 简介	311
12.1.2 建立蓝牙连接	313
12.2 设计要求	317
12.3 类设计	317
12.3.1 GobangMIDlet 类	317
12.3.2 蓝牙连接类	317
12.3.3 棋盘类	318
12.4 代码实现	318
12.4.1 程序打包及运行效果	318
12.4.2 实现与代码分析	319
12.5 实践	336
参考文献	337

第1章

J2ME 入门

J2ME 是 Java 2 平台的缩微版,其目的是为移动电话、个人数字助理(PDA)、电视机顶盒等资源受限的嵌入式设备提供一个 Java 程序运行环境。在这个环境中,既包含了符合嵌入式设备需求的 Java 虚拟机和核心类库,还包含了开发 Java 应用程序需要的类库和接口。

目前,J2ME 在手机、PDA 等移动信息设备上得到了广泛的支持。在本章中,首先简单介绍了 J2ME 的体系结构,然后详细讲解了在移动信息设备上开发 Java 应用程序的相关知识。

1.1 什么是 J2ME

J2ME 是 Sun 公司针对移动电话、PDA 和电视机顶盒等嵌入设备定义的 Java 技术平台,其目的是为这些资源受限的嵌入式设备提供一个公共的、灵活的、开放的 Java 程序运行环境,也为软件开发人员开发嵌入式设备上的 Java 应用程序提供一个灵活、开放的开发环境。

J2ME 平台由一系列技术规范组成,如 CDC(JSR-36)、CLDC1.1(JSR-139)、MIDP2.0(JSR-118)、Mobile 3D Graphics API for J2ME(JSR-184)、Mobile Media API(JSR-135)、Wireless Messaging API(JSR-120)等。这些技术规范是针对不同类型的嵌入式设备或者设备的不同功能定义的,在 J2ME 技术规范的制定过程中,有很多移动设备制造商和软件开发商参与其中。如诺基亚、摩托罗拉、Sun Microsystem、Symbian 等公司。设备制造商使用这些技术规范能够在自己的设备上构建一个灵活的、完整的 Java 程序运行环境,为用户和软件开发人员提供一个公共的、开放的应用程序开发和运行平台。

目前,J2ME 平台在移动信息设备领域得到了广泛的支持,各大移动设备厂商都推出了支持 J2ME 的无线移动信息设备,并提供了相应的软件开发工具包和设备模拟器。

什么是 JSR? 什么是 JCP?

JSR (Java specification requests) 称为 Java 规范请求,是指 JCP (Java community process) 的成员向 JCP 组织提出新增一个标准化技术规范或者对现有规范进行修订的正式请求文档,每个文档都有一个编号。一旦规范请求被通过,该规范就会被加入 Java 平台的下一个版本中。

JCP 是一个开放的国际组织,主要由 Java 开发者以及被授权者组成,其职能是发展和更新 Java 技术规范、参考实现(reference implementations, RI)。

测试套件 (test suites)。JCP 组织维护的规范包括 J2ME、J2SE、J2EE、XML、OSS、JAIN 等。组织成员可以提交 JSR，通过特定程序以后，进入到下一版本的规范里面。

请登录 JCP 组织的网站，了解更多的知识。JCP 组织的网址：<http://www.jcp.org>。

1.2 J2ME 体系结构

J2ME 被设计成为层次结构，主要由配置 (configuration)、简表 (profile) 和可选包 (optional package) 三部分组成，如图 1-1 所示。

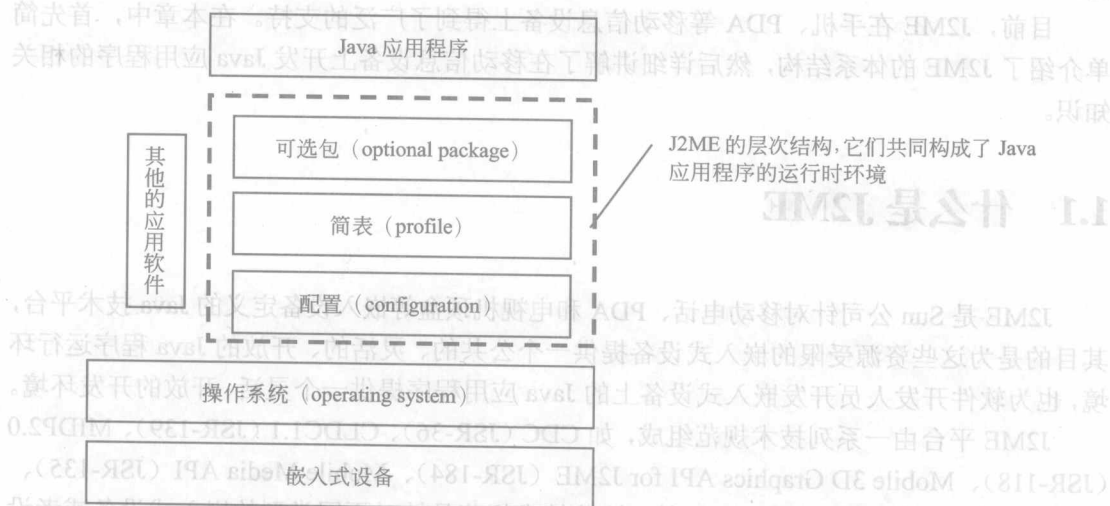


图 1-1 嵌入式设备上的软件结构

1.2.1 配置层

配置层是 J2ME 体系结构的核心。在该层中，定义了符合嵌入式设备需求的 Java 虚拟机 (Java virtual machine, JVM) 和核心类库。

由于不同嵌入式设备的计算能力有很大的差异，所以 J2ME 专家组根据嵌入式设备的计算能力将它们分为两大类：一类是连接受限的设备 (connected limited device)；另一类是连接设备 (connected device)，并为这两类设备分别定义了符合其需求的 Java 虚拟机和核心类库。针对连接受限设备的 Java 虚拟机和核心类库定义在 CLDC (connected limited device configuration) 规范中，针对连接设备的 Java 虚拟机和核心类库定义在 CDC (connected device configuration) 规范中。

- CLDC 称为连接受限的设备配置，针对内存容量较小（通常只有 128~256KB），并且只有低带宽、间歇式网络连接的低端设备。例如，手机、传呼机和 PDA 等，

这类设备通常使用电池供电。专家组根据这类设备的特点,对标准的 Java 虚拟机规范进行了精简,去掉了一些不适合的特性,为计算能力有限的低端设备定义了一个小型的 Java 虚拟机。K 虚拟机(KVM)就是 Sun 公司根据规范给出的一个参考实现,它仅有几十 K 字节。专家组还为这样的小型 Java 虚拟机定义了核心类库,这些类由两部分组成:一部分来源于 J2SE 的核心类库,是 J2SE 核心类库中定义的类的子集;另一部分是专门针对这些低端设备的特点而设计的类。

- CDC 称为连接设备配置,针对内存容量相对较大(通常有 2~4MB),并且具有高带宽、持续网络连接的高端设备。例如,电视机顶盒、网络电视机和汽车娱乐及导航系统等嵌入式设备,这类设备通常使用固定的电源供电。专家组为这类高端设备定义了功能完备的 Java 虚拟机(称为 CVM)和符合该类设备特点的核心类库。

CDC 定义的核心类也由两部分组成:一部分来源于 J2SE 的核心类库,是 J2SE 核心类库中定义的类的子集;另一部分是专门针对高端设备特点而设计的类。CDC 定义的核心类库包含了 CLDC 定义的核心类库。图 1-2 显示了 J2SE、CLDC 和 CDC 定义的核心类库之间的关系。

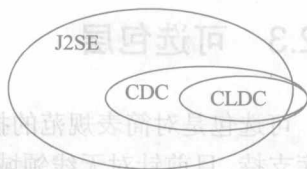


图 1-2 J2ME 配置与 J2SE 的关系

CLDC 和 CDC 分别为低端设备和高端设备提供了最小的 Java 平台,满足了同类嵌入式设备的最低要求。

1.2.2 简表层

简表层构建在配置层之上,其目的是针对设备的功能特性提供 Java 类库支持,对配置层的规范进行扩展和补充。

在简表层,对嵌入式设备按其应用的领域进一步细分,把相同应用领域的设备归为一个设备家族,并为设备家族中的通用功能特性提供 Java 类库支持。支持不同设备家族的 Java 类库分别定义在不同的简表层规范中。例如,针对移动信息设备的 Java 类库定义在移动信息设备简表(mobile information device profile, MIDP)规范中,针对个人数字助理的 Java 类库定义在 PDA 简表规范中。此外,在这些规范中还定义了 Java 应用程序的模型、生命周期和安全特性等。

简表和配置共同构成了嵌入式设备上的 Java 运行环境(Java runtime environment, JRE),在一个设备上只有一个符合其配置规范的 Java 虚拟机和核心类库,但是在一个配置上可以有多个简表规范存在。一个简表规范既可以直接建立在某个配置规范之上,也可以建立在另一个简表规范之上。设备制造商可以选择合适的简表和配置规范在设备上实现,为 Java 应用提供一个运行环境。应用程序开发人员在为嵌入式设备开发应用程序时,也要针对具体的简表与配置规范,以保证应用程序在实现相同简表和配置的设备之间是可移植的。图 1-3 列出了主要的配置与简表及其关系。

在众多的简表规范中, MIDP 是被广泛支持和使用的简表规范,构建于 CLDC 规范之上,主要针对无线、移动、双向通讯的信息设备,如手机、双向传呼机等。MIDP 规范为开发人员提供了应用程序模型、图形用户界面、网络连接、本地数据存储等功能的 Java 类

库, 目前规范的最高版本是 MIDP2.0 (JSR-118)。在本书后面的章节中, 将详细讲述基于 MIDP 的程序设计。

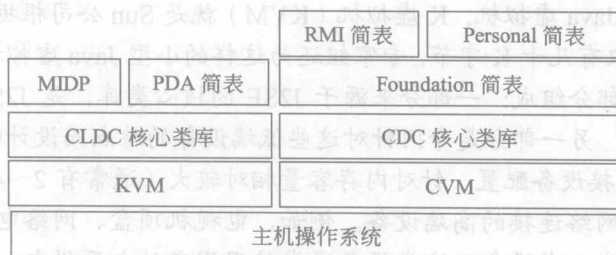


图 1-3 主要的配置与简表及其关系

1.2.3 可选包层

可选包是对简表规范的扩展, 它为用途广泛、独立于设备家族的功能提供了相应的 Java 类库支持。目前针对无线领域的可选包有: Wireless Messaging API (JSR-120)、Mobile Media API (JSR-135)、J2ME Web Services Specification (JSR-172)、Mobile 3D Graphics for J2ME (JSR-184)、Location API for J2ME (JSR-179) 等。无线设备制造商可以视设备的功能需要, 实现相应的可选包规范。随着技术的发展, 一些可选包定义的功能会被集成到简表规范的新版本中。例如, 在简表 MIDP2.0 版本中就集成了可选包 Mobile Media API 的一部分功能。

关于更多的可选包规范的详细信息, 请参考 JCP 组织的网站: <http://jcp.org/en/home/index>。

注

可选包提供的功能并不是必需的, 设备制造商可以支持, 也可不支持。如果应用程序使用了可选包的功能, 那么它在可移植性上会受到限制。然而, 配置和简表提供的功能是强制性的、不可选的, 如果支持, 就必须全部支持。

1.3 J2ME 无线工具包

在学习 MIDP 编程之前, 需要在台式机上安装一个 J2ME 开发环境。Sun、诺基亚、摩托罗拉等厂商都为开发人员提供了免费的开发工具包, 本书使用 Sun 公司的 J2ME 无线工具包 (J2ME wireless tool kit, J2ME WTK)。

1.3.1 安装 J2ME 无线工具包

J2ME WTK 是 Sun 公司提供的功能强大的无线应用开发工具包, 利用它能够开发移动电话和其他无线设备上的 Java 应用程序。J2ME WTK 支持 MIDP 和一些可选包功能,

还提供了运行 MIDP 应用程序的设备模拟器, 这样在台式机上就可以完成软件的开发、测试。但是 J2ME WTK 没有提供文本编辑功能, 必须配合记事本或 UltraEdit-32 等文本编辑器使用。目前 Sun 公司已经发布了 Java Wireless Toolkit 2.2 版, 可以登录 Sun 公司网站 <http://java.sun.com> 免费下载。本书使用 J2ME WTK2.2 版本。

注

目前 Sun 公司已发布了 Java Wireless Toolkit 2.5 测试版, 可以登录 Sun 公司网站 <http://java.sun.com> 免费下载。

5

J2ME WTK 需要有 JDK 支持才能够使用。因此, 在安装 Java Wireless Toolkit 2.2 之前需要先安装 JDK1.4.2 或更高版本的 JDK, 本书使用 JDK5.0 版本 (关于 JDK 的安装, 请参考其他资料)。

安装完 JDK 之后, 运行下载的 Java Wireless Toolkit 2.2 安装文件, 即可完成 J2ME 开发环境的安装。本书用 {toolkit} 表示 J2ME WTK 的安装目录, 作者将 J2ME WTK2.2 安装到 D:\WTK22 目录下, 生成如图 1-4 所示的目录结构。

目录 appdb 包含了设备模拟器的 RMS (记录管理系统)、文件系统和个人信息数据库 (PIM); 目录 apps 包含了 J2ME WTK 自带的演示程序和用户新建的项目; 目录 bin 包含了开发工具的可执行文件; 目录 docs 包含了 API 文档和 J2ME WTK 用户界面使用指南。

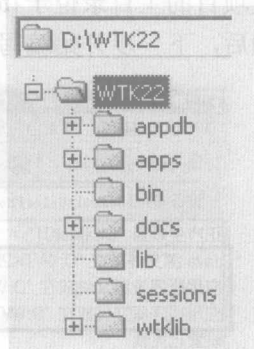


图 1-4 J2ME WTK 目录结构

1.3.2 使用无线工具包开发 MIDP 应用程序

安装完 J2ME WTK 后, 就可以开发 MIDP 应用程序了。

1. 创建新项目

J2ME WTK 以“项目”为单位管理 MIDP 应用程序, 首先打开 J2ME 无线开发工具, 方法为选择【开始】→【程序】→【J2ME Wireless Toolkit 2.2】→【KToolbar】。开发工具界面如图 1-5 所示。

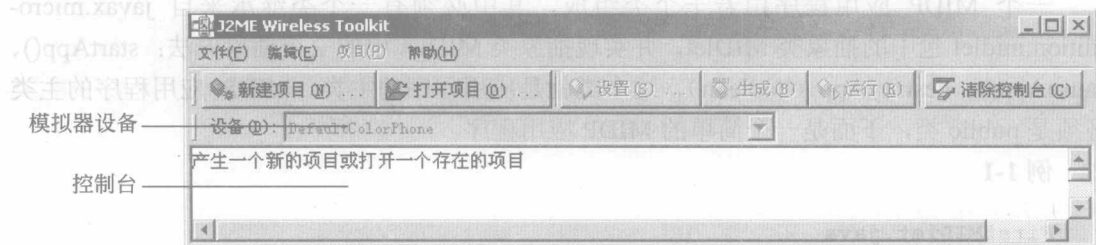


图 1-5 J2ME KToolbar

单击【新建项目】按钮, 弹出新建项目对话框, 提示用户输入项目的名字和 MIDlet 类名, MIDlet 类名是应用程序主类的名字。输入相应的名字如图 1-6 所示。

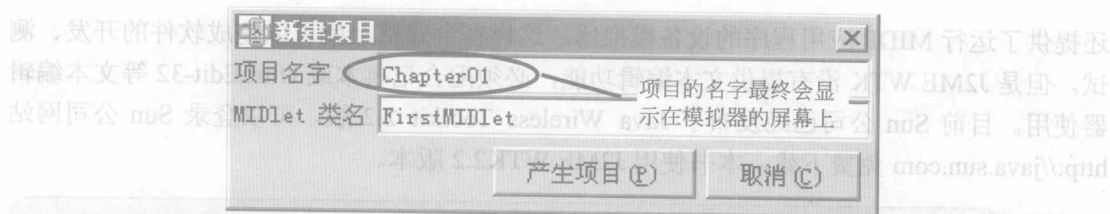


图 1-6 新建项目

单击【产生项目】按钮创建项目，弹出项目设置对话框。在此使用默认的选项设置，单击【确定】按钮，对话框消失，在控制台中显示创建新建项目的结果，如图 1-7 所示。新建项目的主目录位于 {toolkit}\apps 目录中，主目录的名字与项目的名字一致。项目创建成功后，下一步开始编写 MIDP 应用程序。

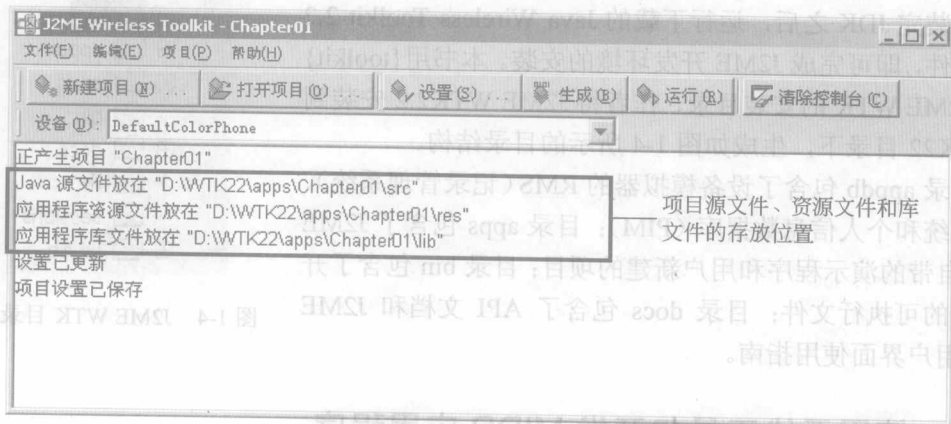


图 1-7 新建项目信息



项目的名字不能包含汉字。

2. 编写 MIDP 应用程序

一个 MIDP 应用程序由若干个类组成，其中必须有一个类继承来自 javax.microedition.midlet 包中的抽象类 MIDlet，并实现抽象类 MIDlet 中的三个抽象方法：startApp()、pauseApp()、destroyApp (boolean)，这个类就是应用程序的主类。MIDP 应用程序的主类必须是 public 类，下面是一个简单的 MIDP 应用程序。

例 1-1

FirstMIDlet.java

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class FirstMIDlet extends MIDlet implements CommandListener {
    protected void startApp() {
```

```

Display display = Display.getDisplay(this);
TextBox text = new TextBox("第一个 MIDlet",
    "我们开始学的是 MIDP 编程!", 127, TextField.ANY);
Command cmdExit = new Command("退出", Command.EXIT, 0);
text.addCommand(cmdExit);
text.setCommandListener(this);
display.setCurrent(text);
}

protected void pauseApp() {
}

protected void destroyApp(boolean undecorated) {
}

//当用户触发“退出”命令时，系统会自动调用这个方法
public void commandAction(Command cmd, Displayable d) {
    if(cmd.getCommandType() == Command.EXIT) {
        notifyDestroyed();
    }
}
}

```

例 1-1 是一个简单的 MIDP 应用程序，只有一个主类 FirstMIDlet，主类的名字必须与创建项目时输入的 MIDlet 类名一致。打开记事本，输入例 1-1 的代码，然后把这个程序源文件保存到 {toolkit}\apps\Chapter01\src 目录中，文件名为 FirstMIDlet.java。

3. 编译、预校验

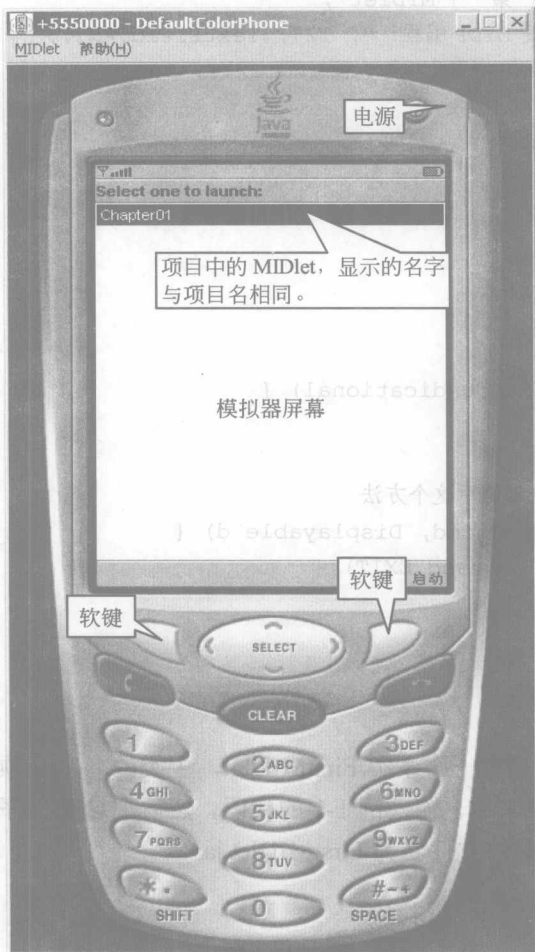
单击开发工具的【生成】按钮，将对项目源文件目录中的所有 java 文件进行编译，生成的类文件位于 {toolkit}\apps\Chapter01\tmpclasses 目录中；然后，预校验器将对 tmpclasses 目录中的类文件进行完整性检查，生成符合 CLDC 规范的类文件。校验后的类文件位于 {toolkit}\apps\Chapter01\classes 目录中。

在 J2SE 中，Java 虚拟机加载类时，首先要对类文件的完整性进行检查，确保类文件中不包含可能破坏系统安全或者违反 Java 语言规范的恶意代码，这个检查过程叫做类的校验。CLDC 规范把类校验过程的一部分放到运行时之前，在台式机上使用预校验器检查类的完整性，这样在运行时，Java 虚拟机加载类时就可以快速的完成类文件的完整性检查，同时也节省了内存的消耗，这种校验方式非常适合资源有限的低端嵌入式设备。校验后的类文件的大小会比以前增加约 5%。

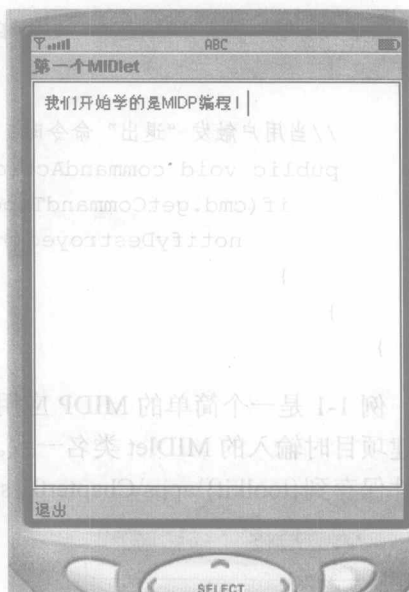
4. 在模拟器上运行、调试

单击开发工具的【运行】按钮，启动模拟器如图 1-8 (a) 所示。单击模拟器中“启动”标签下的软键，在模拟器中运行 MIDP 应用程序，效果如图 1-8 (b) 所示。如果单击“退

出”标签下的软键，MIDP 应用程序退出。单击电源按钮，则关闭模拟器。



(a) 模拟器



(b) FirstMIDlet 运行效果

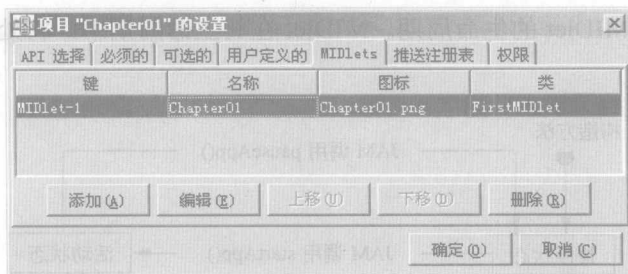
图 1-8 模拟器运行 MIDlet

注

修改项目中 MIDlet 的信息，修改方法为单击开发工具的【设置】按钮，显示当前项目的设置对话框，选择 MIDlets 选项卡，显示当前项目包含的所有 MIDlet 的列表，如图 1-9 (a) 所示。选择要修改的 MIDlet，然后单击【编辑】按钮，显示 MIDlet 详细信息编辑对话框，在此可以修改 MIDlet 的信息，修改后的 MIDlet 信息如图 1-9 (b) 所示。图标文件 App.png 需要保存在 {toolkit}\apps\Chapter01\res 目录中。单击【确定】按钮，再单击设置对话框的【确定】按钮，重新启动模拟器，显示如图 1-9 (c) 所示。

5. 打包、发布应用程序

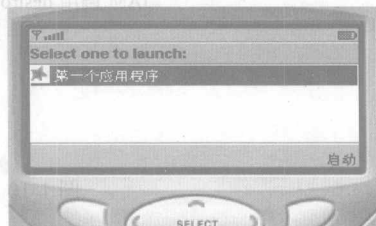
调试通过的应用程序需要打包之后才能安装到手机等移动设备中。关于应用程序的打包、发布将在 1.4.2 节和 1.4.3 节中详细讲述。



(a) 项目设置对话框



(b) 修改 MIDlet 信息



(c) 模拟器中的显示效果

图 1-9 修改项目中 MIDlet 的信息

1.4 MIDP 应用程序

MIDP 应用程序是由若干个类所组成的，其中必须有一个类继承来自 javax.microedition.midlet 包中的抽象类 MIDlet，这个类被称为应用程序的主类。运行时，移动信息设备上预置的 Java 应用管理器（Java application manager, JAM）负责创建主类（MIDlet 的子类）的实例对象。

主类必须重写其父类 MIDlet 中的三个抽象方法：

```
protected abstract void startApp()
protected abstract void pauseApp()
protected abstract void destroyApp(boolean unconditional)
```

运行时，Java 应用管理器通过调用主类中实现的三个方法控制 MIDlet 状态的变化。

注意

Java 应用管理器是由设备制造商开发的系统软件，预置在移动设备中。它主要负责管理移动设备中的 Java 应用程序的安装、卸载、启动等，并且还能控制 Java 应用程序的生命周期。

1.4.1 MIDlet 的生命周期及其状态

一个 MIDlet 从被 Java 应用管理器初始化、运行到被回收，称作一个生命周期。