

J2ME

开发技术原理 与实践教程

陆东林 宾 晟 国 刚 编著
康祥顺 审校

● 本书围绕最新的J2ME所支持的MIDP 2.0技术,详细深入地探讨了如何使用全新版本的MIDP开发时下最炙手可热的手机J2ME应用程序。

● 内容包括J2ME基础、J2ME网络以及J2ME游戏。

● 所有知识由基础训练到高级应用,从功能简单的示例到复杂的网络游戏,本着循序渐进、因势利导的方式为你一步步由浅入深、由简单到复杂、由示例到原理、由原理到应用地揭开J2ME技术的神秘面纱。

● 读者在看完本书后,就能够结合实际,快速、高效、灵活地设计出功能复杂、界面美观的专业级J2ME应用系统。

● 所有的实例基本上可以直接嵌入到自己的手机中直接执行。



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

TP312/2859

2008

J2ME开发技术原理与实践教程

陆东林 宾 晟 国 刚 编著

康祥顺 审校

电子工业出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书围绕最新的J2ME支持的MIDP 2.0技术,详细深入地探讨了如何使用全新版本的MIDP开发时下最炙手可热的运行于手机中的J2ME应用程序,内容主要包括三个部分:J2ME基础、J2ME网络以及J2ME游戏。所有知识由基础训练到高级应用,从功能简单的示例到复杂的网络游戏,本着循序渐进、因势利导的方式为读者一步步由浅入深、由简单到复杂、由示例到原理、由原理到应用地揭开J2ME技术的神秘面纱,使读者在看完本书后,就能够结合实际,快速、高效、灵活地设计出功能复杂、界面美观的专业级J2ME应用系统来,而且所有的实例基本上可以直接嵌入到自己的手机中执行。本书适合希望学习Java手机编程的初学者,也可作为手机程序开发人员的参考用书。

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

J2ME开发技术原理与实践教程/陆东林,宾晟,国刚编著. —北京:电子工业出版社, 2008.6
ISBN 978-7-121-06325-1

I. J… II. ①陆…②宾…③国… III. ①Java语言—程序设计②移动通信—通信设备—应用程序—程序设计 IV. TP312 TN929.5

中国版本图书馆CIP数据核字(2008)第044150号

责任编辑:徐云鹏 易 昆

印 刷:北京天竺颖华印刷厂

装 订:三河市金马印装有限公司

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

北京市海淀区翠微东里甲2号 邮编:100036

开 本:787×1092 1/16 印张:22.5 字数:570千字

印 次:2008年6月第1次印刷

定 价:44.00元

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zhts@phei.com.cn, 盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

Java语言是由Sun公司于上世纪90年代初期开发出来的，Sun公司开发Java的主要目的是为了实现小型电子设备的嵌入式编程。但是在开发初期，Java并没有受到人们的重视，随着Internet技术的发展，WWW技术逐渐渗入到人们工作、生活的多个方面中。同时Java优秀的跨平台能力，在Internet技术中也逐渐展现出来。于是Java迅速占领了PC机和Server端市场。

随着硬件技术的逐步发展，小型电子设备的处理能力和内存容量的增加，使得嵌入式编程在小型电子设备上得到广泛的应用。这对于Java来说，是个重要的发展机会。全球各大手机品牌，相继开始在新产品中支持Java编程。例如：诺基亚、摩托罗拉、西门子、爱立信等手机厂商，先后在推出的一系列新产品中间，增加了Java程序的运行平台。这种手机的二次开发，也可以增加手机的功能。现在在硬件设备中间还可以增加蓝牙设备，通过蓝牙来实现设备之间的网络开发，实现手机的网络游戏等的新软件开发。除了手机设备外，在其他的小型电子设备上也得到了广泛的应用，例如PDA等。

为了更好地管理类库，根据针对的设备不同，Sun将Java分成了三个不同版本：J2SE标准版，主要针对PC机；J2EE企业版，主要针对服务器编程，在J2SE的基础上添加了一些新的技术；J2ME精简版，主要针对小型电子设备，在J2SE的基础之上，删减了部分类，只留下核心类库。本书主要针对J2ME的开发，假定读者已经了解J2SE，会编写J2SE程序。

随着Java的发展，出现了很多针对Java开发的集成开发环境，具有代表性的是JBuilder和Eclipse。在本书中采用的是Eclipse开发环境。Sun为小型移动电子设备编程设计了模拟器，先将程序在模拟器上运行成功以后再写入手机中，在本书中采用的模拟器是WTK2.1。

本书共分为17章来介绍J2ME，涉及J2ME的界面、游戏、网络 and 存储系统等，每一章的内容分别如下：

第1章：介绍Java的发展过程，介绍专门针对小型电子设备开发的J2ME。由于小型电子设备的种类繁多，J2ME分为三层结构：Java核心类库、J2ME的配置和J2ME简表。并介绍MIDP类库，以及J2ME中的应用程序MIDPlet的原型。

第2章：为了J2ME程序的开发，Sun公司开发了WTK手机模拟器，本章介绍模拟器的安装。很多公司设计了针对Java程序开发的集成开发工具，以JBuilder和Eclipse为代表，这一章还将介绍这两种集成开发工具的安装和配置。

第3章：介绍手机程序的用户界面，设计用户界面的控件，及直接添加到屏幕上的控件和Form表单中的控件。除了基本的控件之外还有Command指令。在高级用户界面中，J2ME跟J2SE处理事件的方式相同，采用监听接口的处理方式。

第4章：介绍低级用户界面，主要用Graphics对象在Canvas屏幕上绘图，绘图包括基本的几何图形、字符串和图像，除了绘图之外，Canvas中间的事件处理采用的是事件方法。在Canvas中有作用于不同事件的方法，当事件产生时，程序自动调用相关的事件处理方法。

第5章：介绍小型移动电子设备中的记录存储系统，及如何创建和打开记录，添加、删除、修改、查询记录，还有关于记录存储系统的高级操作。

第6章：介绍小型移动电子设备的网络开发，网络连接的通用连接框架，及通用连接框架中间的七个连接接口。

第7章：介绍J2ME中小型电子设备基于HTTP协议的网络开发，及HTTP协议中的不同的请求方式。

第8章：介绍J2ME中基于Socket的网络开发，分别介绍了服务器端的ServerSocketConnection类和客户端的SocketConnection类。类似J2SE中的Socket网络开发，本章通过Socket实现小型移动电子设备和PC机之间的通信。

第9章：小型电子设备的硬件发展；使红外和蓝牙技术在小型电子设备上得到广泛的应用，本章分别介绍红外和蓝牙技术的通信。

第10章：介绍小型电子设备的信息通信的应用开发，利用WMA类库中的类实现通信。

第11章：通过MIDP 2.0中设计的Media类库，实现小型移动电子设备的多媒体开发，实现音频和视频文件的播放。

第12章：在MIDP 2.0中，设计了专门针对小型移动电子设备上的游戏开发的类库，在Game类库中利用Canvas的子类GameCanvas实现屏幕类，利用层的概念，大大方便了小型移动电子设备上的游戏开发，其中还专门设计了背景层TiledLayer和游戏图像层Sprite。除了这些之外还设计了对这些进行管理的LayerManager类。在Sprite类中设计了图像的旋转和判断各Sprite之间或者是与TiledLayer之间是否相遇的方法。在GameCanvas中设计了专门的事件，通过getKeyState()方法，可获得键盘当前的状态，判断键盘的状态来实现相应的操作，整个事件需要通过线程来完成。

第13章：介绍MIDP中的3D开发，在J2ME中实现3D效果分为两种，一种是通过运算绘图在屏幕上生成3D效果，另一种是利用M3G文件实现3D效果。

第14章：游戏实例，利用MIDP中的Game API类库中设计的游戏开发的类，完成坦克大战的游戏，综合利用了层的概念。

第15章：游戏实例，利用Game API中的TiledLayer层，实现图像的分割和每块图像的随机显示完成拼图游戏中的图像。利用GameCanvas中的事件方式来处理图像的移动。

第16章：利用低级用户界面Canvas完成连连看游戏，其中的图像，选中的方块标记都是通过Graphics中的绘图实现的。

第17章：通过Game API中的GameCanvas实现俄罗斯方块游戏。将每一个俄罗斯方块中的图像转换成二维数组，通过判断二维数组中的值来判断图像。

本书的顺利完成还得益于多名技术骨干和专家的大量参与和支持，在这里要感谢他们的辛勤劳动：孙更新、满载龙、王萍萍、孙强、杜娜、王寿苹、聂江武、仇培铭，另外还要感谢电子工业出版社的相关同志，没有你们的大力支持和厚爱，这本书是无法与广大读者见面的。

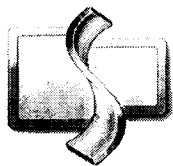
为方便读者阅读，若需要本书配套资料，请登录“华信教育资源网”（<http://www.hxedu.com.cn>），在“资源下载”频道的“图书资源”栏目下载。

目 录

第1章 J2ME概述	1
1.1 J2ME背景介绍	1
1.2 连接设备简介	5
1.3 MIDP简介	8
1.4 MIDlet简介	10
1.5 本章小结	15
第2章 J2ME开发环境配置	16
2.1 J2ME通用开发工具	16
2.2 J2ME集成开发工具	17
2.3 本章小结	28
第3章 高级用户界面	29
3.1 MIDP高级用户界面设计概述	29
3.2 Display类和DisplayAble类	30
3.3 屏幕类	32
3.4 表单中的组件	42
3.5 事件处理	52
3.6 MIDP界面实例	62
3.7 本章小结	69
第4章 底层用户界面	70
4.1 Canvas画布屏幕设计	70
4.2 图形绘制与Graphics类	70
4.3 文字绘制与Font类	76
4.4 图像处理与Image类	77
4.5 Canvas事件处理	80
4.6 绘图示例：表格	82
4.7 绘图示例：计算器	92
4.8 本章小结	101
第5章 记录存储系统	102
5.1 记录存储系统概述	102
5.2 记录存储系统基本操作	102
5.3 记录存储系统高级操作	104

5.4 备忘录实例	108
5.5 本章小结	118
第6章 通用连接框架	119
6.1 通用连接框架概述	119
6.2 通用连接框架的各种连接	119
6.3 通用连接框架的七个接口	129
6.4 本章小结	129
第7章 基于HTTP协议的网络开发	130
7.1 HTTP协议概述	130
7.2 建立HTTP连接	132
7.3 本章小结	141
第8章 基于Socket和数据报编程的网络开发	142
8.1 Socket和数据报概述	142
8.2 Socket编程	143
8.3 本章小结	168
第9章 基于红外和蓝牙协议的网络开发	169
9.1 无线数据传输概述	169
9.2 蓝牙技术通信	170
9.3 红外技术通信	173
9.4 蓝牙技术实例	175
9.5 本章小结	185
第10章 Messaging应用开发	186
10.1 无线消息WMA概述	186
10.2 WTK提供的模拟工具	186
10.3 WMA类库的介绍和使用	188
10.4 WMA实例	191
10.5 本章小结	197
第11章 多媒体开发	198
11.1 MM API概述	198
11.2 媒体管理Manager类	198
11.3 播放器接口Player类	199
11.4 程序实例	200
11.5 本章小结	206

第12章 MIDP游戏编程	207
12.1 MIDP 2.0 Game API概述	207
12.2 游戏容器GameCanvas类	207
12.3 游戏精灵Sprite类	212
12.4 游戏地图TiledLayer类	218
12.5 图层管理LayerManager类	221
12.6 游戏实例：蜜蜂射击	223
12.7 本章小结	241
第13章 Mobile 3D程序开发	243
13.1 3D开发概述	243
13.2 及时运算生成3D模型	244
13.3 外部建模导入3D模型	250
13.4 本章小结	253
第14章 单屏幕游戏：坦克大战	254
14.1 程序分析	254
14.2 程序模块分析	255
14.3 本章小结	282
第15章 单屏幕游戏：拼图游戏	283
15.1 程序分析与设计	283
15.2 本章小结	292
第16章 游戏实例：连连看	293
16.1 游戏介绍分析	293
16.2 游戏的功能	293
16.3 本章小结	330
第17章 游戏：俄罗斯方块	331
17.1 游戏分析	331
17.2 功能设计	332
17.3 本章小结	350



第1章 J2ME概述

J2ME (Java 2 Platform Micro Edition) 是移动商务开发的最佳工具。例如：手机、PDA、无线通信等小型电子设备，都可以采用J2ME作为开发工具，而且随着Java技术的发展，应用也是越来越广泛。Java的特点有哪些？为什么Java可以广泛应用于各个领域？Java与J2ME分别指什么？它们之间有什么关系和联系？

1.1 J2ME背景介绍

为了更好地学习J2ME，我们应该先了解Java语言的历史，Java语言的发展状况，Java语言的运行原理，还包括J2ME的组成部分，以及各部分的简介。

1.1.1 Java技术应用现状

20世纪90年代初，Sun公司主要用C++从事一种手持式电子设备的软件开发，在开发过程中，发现C++的某些特性容易出现错误，严重影响软件的可靠性。例如，C++中允许类之间可以多继承，即一个类可以有多个父类，如果两个父类中有相同的属性，那么子类将两个父类的属性和方法都继承下来，就会有冲突。

所以Sun公司的开发小组发现了C++的一些问题后，在C++的基础上开发了一种新的编程语言，即OAK，橡树的意思。OAK借用了C++的很多元素，删除了C++的很多特性，增加了程序的可靠性，在编译的过程当中，可尽可能检测多数错误来减轻程序负担。OAK诞生之初，并没有受到人们的重视，直到Netscape公司第一个支持OAK的浏览器诞生，OAK才逐渐被人们重视，并在1994年更名为Java。

Java程序语言是一种面向对象的高级编程语言，除具有面向对象编程语言的多态性、封装性、安全性之外，Java虚拟机独具的跨平台的功能使Java还独具高移植性。Java的高移植性使得Java自面世以来，已成为Internet开发的一种主要语言，Java语言的出现使得Web技术发生了翻天覆地的变化。

随着Java跨平台功能的增强，以及消费类和嵌入式电子设备的发展，小型消费类电子设备和嵌入式设备的软件开发也越来越受到人们的重视。而且Sun公司最初开发Java的目的也就是为手持式电子设备开发软件的，所以Sun公司将Java语言也向小型电子设备方向进行了延伸，为消费类电子设备提供了专用的API库。

越来越成熟的Java语言已经应用到了各行各业，各种类型的软件开发中。包括服务器端的、PC端的以及消费类电子设备和嵌入式设备等移动设备的软件开发。Java在整个软件行业中有着举足轻重的地位。据不完全统计，现在支持Java语言的手机已经超过了80%，而且网站、网上银行等也采用了Java的服务器技术。我们可以毫不客气地说，我们的现代生活已经离不开Java。

1.1.2 J2ME简介

前面介绍了Java的发展历史和Java现在的应用状况,不难看出,Java已经深入到了人们生活的各行各业,例如:手持式移动电子设备、服务器、PC等,对于不同的领域,Java针对每种行业不同的特点,设置了不同的版本。Java一共分为三种版本:

J2SE (Java 2 Platform Standard Edition), Java标准版, Java最主要、最核心的版本,包含了Java最核心的部分。主要用于PC端的桌面程序的开发和低端商务应用系统开发。例如: C/S网络系统开发, PC机程序的开发。

J2EE (Java 2 Platform Enterprise Edition), Java企业版, Java的高级版本, 在Java核心的基础上, 添加了一些针对Web服务器的开发, 引用了一些新的技术, 专门为企业提供电子商务系统开发以及Web服务器的解决方案。例如: OA办公系统等网络服务系统。

J2ME (Java 2 Platform Micro Edition), Java精简版, 由于J2ME主要针对小型电子设备的开发, 这些设备都有一个最大的特点, 内存空间小, 所以, J2ME在Java核心的基础上进行了删减, 留下了所有小型设备通用的部分, 所以是Java精简版。主要用于无线通信设备、手机、PDA等小型电子设备和嵌入式设备的软件开发, 是移动商务的最佳编程语言。

综上所述可以看出Java的三个版本分别针对不同的领域, 如图1.1所示为三个版本分别针对的不同领域。

在Java的三个版本中, 基本的语法是完全相同的, 只是根据针对的对象设备不同, API库有些差异。J2SE包含了Java基本的、核心的元素, 是J2EE和J2ME的基础。J2EE是Java的高级版本, 在J2SE的基础上增加了新的类库, 引入了新技术。例如: EJB、JSP、Spring等主要用于服务器端的程序。J2ME在J2SE的基础上删减了部分内容, 而且还引入了专门的API类库, 主要用于移动设备等小型电子设备的软件开发。J2ME包含了专门针对小型电子设备的界面开发、网络开发和游戏开发。三个版本的关系如图1.2所示。



图1.1 Java的三个不同版本

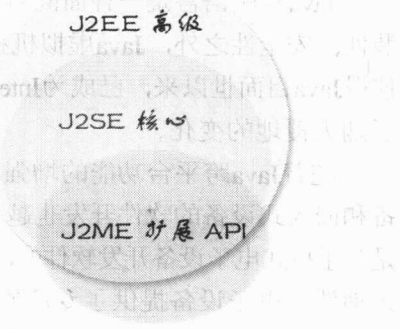


图1.2 三个版本的关系

1.1.3 J2ME体系结构

根据前两节内容, 我们可知, J2ME主要针对小型电子设备、移动商务的软件开发, 然而, 现在在市面上流行的小型电子设备可谓是种类繁多。例如: 手机、商务PDA等电子设备。

而且这些种类繁多的电子设备又没有一个统一的标准,硬件设备和性能上都有很大的差异。例如:设备的内存容量不一样,显示设备不一样(有彩色的,黑白的等),输入设备也不一样(有的是普通手机标准键盘,还有的是计算机的标准键盘)。小型电子设备的差异,使得我们不太可能创造一个适用于任何设备的软件。为了适应小型电子设备没有统一标准的特点, J2ME在基础核心的基

础上,提出了Configuration(配置)、Profile(简表)和可选包三部分,根据不同的电子设备,这三部分具有不同的内容。各部分负责不同的内容,它们之间的关系如图1.3所示。

软件开发商开发出来的软件,希望能在某一类特定的设备上运行,例如手机软件开发商开发出来的软件就希望可以在手机上运行,软件开发商并不希望自己开发出来的软件能够在PDA上运行。但是,软件开发商也不希望给每一类电子设备都开发一种语言。所以针对不同的电子设备,核心不用变,都是Java核心,根据不同的电子设备自己的特点去实现Configuration、Profile和可选包三种规范,这样针对每一类电子设备就形成了针对自己特点的Java运行环境。在某一类特定设备上,还可以提供自制的Java包,增加该特定设备所需要的包。例如:手机这种设备, Configuration是核心内容,它包含了最小库和虚拟机的规范,只是提供了J2ME所必需的包。只有Configuration是不够的, Profile是功能包,它具体实现功能的规范和功能扩展包,例如网络连接、特定设备的用户界面等。可选包是辅助功能,对于一些特定设备的生产厂商来说,增加可选包,可以使得该特定设备具有其他设备所没有的功能,当然,手机已经没有什么特殊功能,可以没有可选包。

Configuration指定了某一类特定设备的基础类库和虚拟机规范,是该类设备J2ME平台应具备的基本功能,各设备厂商只需要实现相应的规范,那么该设备就具备了J2ME的运行环境。

Profile给某一类设备提供一个完整的运行环境, Configuration只是提供了一个最小类库的规范, Profile还附加了API库,是对Configuration的一个补充。

可选包是进一步的扩展,可以将Configuration以及对应的Profile,再加上可选包组合起来应用,例如在一些特定的设备上,为了一些特殊的市场需求,硬件厂商在设备上添加了一些只用于该设备的包,增加了设备的特点,当然这是以降低软件的可移植性作为代价的。

1.1.4 J2ME配置

针对小型电子设备的种类繁多的特性、每类设备的内存数量、处理器的处理速度以及是否支持网络连接等特点,对某些具有相似属性的设备, Configuration定义了最小的类库和虚拟机的规范,指定了J2ME开发平台的基本功能。定义出的Configuration要求给出最小通用特性,适用于大量的设备,而且没有可选的特性,减少了一些J2SE的核心,修改了J2SE某些类。

到目前为止, J2ME一共定义了两种规范: CLDC (Connected Limited Device Configuration, 有限连接设备)和CDC (Connected Device Configuration, 连接设备)。CDC采用Java的经典控制器JVM,而CLDC采用的是KVM, KVM是Sun公司专门为小型电子设备设计的控制器。 Configuration只是定义了最核心的库和虚拟机规范,所以, Configuration不能单独使用,应该和Profile组合使用。 Configuration中没有实现所有的功能。例如Configuration中没有定义

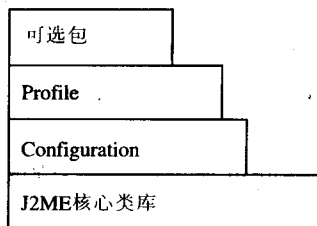


图1.3 J2ME各部分关系

网络连接的库，如果想连上网络，实现设备的网络功能，需要实现Profile规范。

Configuration并不是虚拟机，它只是定义了虚拟机应该具有的规范和J2ME平台的核心库。其中CLDC主要针对的是大量小型设备，通常它们的内存有限，处理器速度慢，而且具有间歇的网络连接，例如手机、PDA等。而另一类CDC，它所针对的设备相对来说，内存会更大，处理器的速度更高，具有更大的网络带宽，例如电视机顶盒等。

1.1.5 J2ME简表

Configuration只是提供了核心类库和虚拟机的规范，所以，Configuration需要和一些高级的API组合使用。我们可以认为Profile是Configuration的扩充。Profile定义了用户界面API等其他特性。Profile是针对特定行业或者特殊设备设计的。

Configuration上可以定义一个或者多个Profile，目前在CLDC上的Profile有MIDP（Mobile Information Device Profile，即移动信息设备Profile），在CDC上有FP（Foundation Profile，即基础Profile）、PP（Personal Profile，即个人Profile）、PBP（Personal Basic Profile，即个人基础Profile）。其中PP和PBP是以FP为基础的。

MIDP针对移动电话等低端PDA，提供了用户界面、网络连接、局部数据存储等功能。CLDC与MIDP一起为手持设备提供了完整的Java运行环境。

FP是CDC中Profile的基础。为了使不同类型设备有更多选择，可以按照需要增加设备的功能，CDC中的Profile采用了分层机构，FP是CDC中最低层的。它提供了网络连接，没有用户界面，如果需要界面的话，就需要跟PP或者PBP一起使用。

PP主要针对需要用户界面的设备，例如通信装置设备或者游戏控制台等设备。它包括Java抽象窗口工具包，提供了Web能力。

PBP为网络连接的设备提供了应用程序环境，例如电视机顶盒等设备。

1.1.6 J2ME虚拟机

我们首先来回顾一下C语言的执行过程，如图1.4所示。



图1.4 C语言的执行过程

C语言源程序经过编译，生成二进制文件，我们叫目标文件，目标文件经过链接，生成可执行文件，然后我们可以运行，当我们在运行的时候，程序会向操作系统发出一系列指令，由操作系统根据指令的要求控制计算机硬件，试想一下，如果操作系统发生了改变，换了另外一个操作系统，原来的指令，新操作系统可能已经没有办法识别，所以C语言的跨平台能力有缺陷。我们现在来看一看Java的运行过程，如图1.5所示。

Java的源程序经过编译生成字节码文件，所谓的字节码文件是一套Java语言的编码，这些编码通过Java虚拟机进行解析，然后由虚拟机通过解析到的内容向操作系统发出指令，如果操作系统不一样，虚拟机会根据现在的操作系统，解析成现在的操作系统可以识别的指令。然后由操作系统来控制计算机的硬件。我们不难看出，虚拟机增加了Java的可移植性，使得

Java语言具有跨平台的能力，操作系统发生变化时程序是不用变的。随着网络技术的不断发展，对于在网络上运行的程序，这种跨平台的能力显得尤为重要，这也是Java盛行的最主要原因。

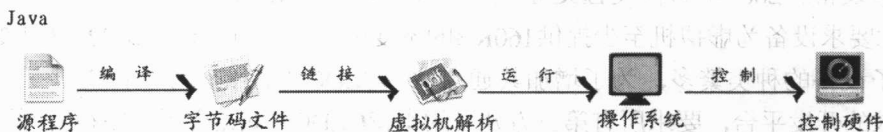


图1.5 Java执行过程

对于虚拟机，我们可以认为它就是一个虚拟的，通过软件模拟的CPU，Java虚拟机一共由五部分组成：一组指令集、一组寄存器、一个栈、一个无用单元收集堆、一个方法区域。这五部分不依赖任何实现技术或者组织方式。指令集：Java所支持的字节码，每种字节码执行一种基本的CPU运算。寄存器：Java虚拟机的寄存器用来保存机器的运行状态，与微处理器中的某些专用寄存器类似。无用单元收集堆：Java的堆是一个运行时数据区，类的实例从中分配空间。方法区：方法区与传统语言中的编译后代码或者UNIX进程中的正文段类似，保存方法代码和符号表。

1.2 连接设备简介

1.2.1 CDC概述

本书编程的重点在小型设备上，所以，我们介绍CDC时不会像介绍CLDC那样深入，对于CDC我们只是了解CDC中的API。

CDC所对应的目标设备是消费性电子产品和嵌入式设备。相对而言，CDC所面对的设备，内存会更大，处理器的运行速度会更快，有更大的网络带宽。例如：智能电话、商务PDA、家用电器以及汽车导航系统等。CDC对设备的要求：要求设备运行32位微处理器，拥有超过2MB的内存空间。

相对于CLDC而言，CDC是整个J2SE API更完整的子集，包括CLDC中定义的所有相同的类和接口，还有一个额外的软件包（`Javax.Microedition.Io`）。

CDC在CLDC中添加的功能：

- 支持浮点数（包括`java.lang.Float`、`java.lang.Double`和`java.lang.StrictMath`）；
- 支持本地进程；
- 支持高级多线程；
- 支持文件系统；
- 支持J2SE类型网络（`java.net`）；
- 支持J2SE的`java.util.jar`、`java.util.zip`等软件包。

1.2.2 CLDC概述

Configuration中包含虚拟机规范和设备所要求的最小类库，CLDC是其中的一种，本书主要针对小型设备，所以我们会将重点放在CLDC上。

CLDC适用于那些具有间歇的无限网络连接,而且连接的速度比较慢,处理器处理速度较慢,而且内存也受到限制的小型设备。CLDC对设备的要求:CPU是16位或者32位的处理器,具有独立的电池等提供电源,内存在128KB~512KB之间的小型设备。例如:手机、寻呼机等通信设备。2005年5月,CLDC正式发布了1.0版(JSR 30)。

CLDC要求设备为虚拟机至少提供160KB的不变内存,同时还应提供32KB可变内存。由于小型电子设备的种类繁多,为了增加其通用性,CLDC为有限资源设备提供了标准化、满足最小要求的开发平台,要求所有第三方开发的支持CLDC的Java应用程序在公开发布时,要使用jar包的格式。

CLDC中未做处理的事务:

- 友好的用户界面
- 事件处理
- 用户交互

以上三项事务没有实现,这三项事务在Profile中处理。CLDC中不支持多进程、分批技术,也不会协调延迟,但是必须有实体运行虚拟机,同时CLDC不支持浮点数和浮点运算。

CLDC为最小要求开发平台,它实现的功能如下:

- Java核心和虚拟机的规范
- Java最基本的类库
- 数据针对设备的输入和输出
- 无线网络数据传输
- 设备程序的安全

CLDC通过以下三个不同层次的安全处理,增加了设备本身的安全性能,使得程序不会影响到设备本身的安全:

- 底层安全:程序必须符合Java语言的规范,设备虚拟机只能运行在符合Java语言规范的程序上,使得设备本身无损害;
- 应用层安全:在设备允许的条件下,才可以使用Java提供的类库,对于那些可能会影响到该设备安全的类库,设备不允许程序调用;
- 点对点安全:通过无线网络进行信息的交换时,通过数字签名和数据加密的方式,使得数据受到保护。

CLDC类库是将J2SE类库精减到最基本开发平台所需的类,包括java.lang、java.io、java.util,同时CLDC提供了专门针对小型电子设备开发的javax.microedition.io包下的类。下面将对这几个包中的类进行详细的描述。

java.lang软件包的核心类:

- System: 提供一些关于系统的静态方法,跟J2SE提供相同的方法
- Object: 超级父类,跟J2SE一样,是所有类的父类
- Class: 显示正在运行的Java应用程序中的类和接口
- Runtime: 提供程序与运行环境相交互的方法,可以通过这个类调用应用程序
- Thread: 线程类
- Throwable: Java语言中所有异常和错误的超级父类

java.lang核心数据类:

- **Boolean**: 布尔 (boolean) 原始数据类型
- **Byte**: byte的原始数据类型
- **Character**: char原始数据类型
- **Integer**: int原始数据类型
- **Long**: long原始数据类型
- **Short**: short原始数据类型

大家不难从中看出, 核心数据类中没有浮点型, 这是因为CLDC删去了冗余的浮点型数据和数据处理。

java.lang帮助类:

• **Math**: 数学类, 包含基本数学运算的方法。注意, 浮点运算方法删掉了, 只剩下整型的方法: `abs()`、`min()`和`max()`

• **String**: 跟J2SE中的String一样, 代表字符串类

• **StringBuffer**: 跟J2SE一样, 代表具有缓冲区的字符串对象, 及可以修改的字符串

java.io输入输出类:

用来从外部设备输入的类:

• **ByteArrayInputStream**: 带内存缓冲器的输入流, 每次读取一个字节

• **DataInput**: 接口, 从二进制输入流读取字节并把它们转换成原始的Java数据类型

• **DataInputStream**: 是实现了该接口的类

• **InputStream**: 抽象类, 字节输入流的父类

• **InputStreamReader**: 读取字节按照指定的字符编码转换成字符

• **Reader**: 抽象类, 字符输入流的父类

向外部指定设备输出的类:

与上面输入类相对应的分别为: **ByteArrayOutputStream**、**DataOutput**、**DataOutputStream**、**OutputStream**、**OutputStreamReader**、**PrintStream**、**Writer**。以上所有的类都不包含浮点数的处理。

java.util收集类:

• **Vector**: 矢量, 可以调整大小的数组

• **HashTable**: 不受大小限制的容器, 将键映射到值

• **Stack**: 堆栈, 按照先进后出原则存放数据的容器

• **Date**: 时间和日期类, 精确到毫秒

• **Calendar**: 抽象类, 用整型字段来获得或者设置日期

• **Random**: 生成整型的随机值流

• **TimeZone**: 时区的偏移量, 用于调整时区来校正时间

以上所提到的类都跟J2SE中所学到的类基本相同, 除了这些类之外还提供了一个新的软件包——**javax.microedition.io**, 其中包含了这些小型电子设备网络连接所用到的流, 在后面的内容中我们会详细介绍。

1.3 MIDP简介

1.3.1 MIDP设备

MIDP就是移动信息Profile，它是专门对CLDC的补充，现在的移动设备种类繁多，为了增加程序的可移植性，所以，MIDP只是规定了那些必需的功能设计。这些功能主要包括：

- 应用程序的发布
- 应用程序的生命周期
- 应用程序的签名模块
- 点对点通信安全
- 网络
- 永久存储
- 用户界面（GUI）
- 音频

而那些系统级的编程，不在我们的考虑之列，所以这里没有包含系统级的编程。

MIDP建立了一个第三方开发环境，包含了网络和用户界面等的编程，所以适合如下的硬件设备：包含单手键盘或者触摸屏的输入设备，具有256KB的非可变内存和8KB可变内存用于应用程序永久保存数据、支持间断的有限带宽的网络、可播放声音、具有显示屏的设备。

例如：移动电话、无线上网的低端PDA等设备。

我们利用MIDP在这些可移动的手持设备上编程时，MIDP本身不会自己去设计操作系统等，所以，要MIDP可以在设备上顺利编程，设备上应该先设计如下的软件：首先得有个操作系统对硬件进行管理，也就是说MIDP中的类不会直接去控制设备的硬件，而是将指令给操作系统，由操作系统来控制计算机硬件。当然这个系统跟我们PC机上的系统有很大的差异，去掉了系统的很多内容。但它需要具备读取无线网络的能力，还得具备读写可变内存的能力，及从输入设备上读取数据，可以向屏幕绘图的内容能力。硬件设备具备了以上的软件环境之后，我们就可以用MIDP通过以上的软件来控制计算机进行硬件编程了。

1.3.2 MIDP类库

MIDP中首先包含CLDC中所涉及的四个核心包，分别是java.lang、java.util、java.io和javax.microedition.io包。除了这些包外，还提供了如下所示的包，表1.1中列出了J2ME中所提供的界面的包。

表1.1 用户界面包

javax.microedition.lcdui	提供用户界面和底层图形操作
javax.microedition.lcdui.game	开发移动设备的游戏程序

在表1.2中列出了IO包中包含的J2ME中的网络以及数据传输的流。

表1.2 J2ME支持网络的包

javax.microedition.io	对于网络的支持包
-----------------------	----------

在小型移动电子设备中，有专门用来永久保存数据的存储器，表1.3列出了具有永久存储的类的包。

表1.3 Java永久存储包

javax.microedition.rms	永久存储数据，并可将数据取出
------------------------	----------------

表1.4中列出了包含MIDlet类的包。

表1.4 包含MIDlet类的包

javax.microedition.midlet	移动信息设备的应用程序和运行环境的交互
---------------------------	---------------------

表1.5列出了J2ME编程中具有可以用在多媒体编程的类中的包。

表1.5 音频和音调包

javax.microedition.media	移动媒体包
javax.microedition.media.control	定义用于播放器的控制类型

1.3.3 MIDP 2.0新特性

2002年10月，JCP（Java Community Process）继MIDP 1.0之后，推出了MIDP 2.0规范，和MIDP 1.0相比，它增加了很多新的功能，例如：增加了用户界面、多媒体技术、游戏API和网络连接功能，并且为无线信息设备提供了端对端（end-to-end）的安全机制，同时将OTA（Over-The-Air）应用程序下载包括到了规范中。

MIDP的一个完整开发过程，包括程序编码和调试，程序编写完整，调试没有错误，就已经完成了其中的大部分任务，还有最后一件事情，就是程序的部署，这个过程叫做Provisioning。用户可以通过这个下载和安装程序。

我们可以利用硬件开发商开发的硬件设备，将程序下载到我们的移动设备上，但是，大部分用户没有这个技能，而且需要硬件设备，安装不方便。用户希望可以通过无线网络下载应用程序，像接收和发送短信一样，这种方式叫做OTA。

MIDP 2.0对用户界面的提高：允许用户对应用程序中的层（Layout）和虚拟元素进行更好的控制。它还增加了以下功能设计：

- 可查询设备的色彩、高亮前景色彩、高亮背景色彩和边界等。
- 可激活背景光和震动。
- 可查询List、Choice和Alert中使用的图片，选择最佳宽度和高度。
- 可查询高亮和普通元素的边界样式。
- 拥有短标签和长标签两个命令。

MIDP 2.0支持对声音的处理，其中增加了一个可选包ABB（Audio Building Block），以前ABB是包含在MMAPI（Mobile Media API）中的，现在被纳入到MIDP中了，也就是说