高职高专院校“十三五”精品示范系列教材 软件技术专业群

国家骨干高等职业院校建设项目成果

Java ME 手机应用程序开发

易 灿 李志勇 编 著

刘彦姝 主 审

教材特色：

- 平台课＋模块课 搭建专业群课程
- 实例驱动＋项目同步 优化内容 新颖实用
- 教材＋案例＋实战素材 立体化资源相结合

中国水利水电出版社

www.waterpub.com.cn

高职高专院校“十三五”精品示范系列教材（软件技术专业群）

Java ME 手机应用程序开发

易　灿　李志勇　编　著

刘彦姝　主　审

中国水利水电出版社
www.waterpub.com.cn

·北京·

内 容 提 要

本书通过一些实际案例详细地讲述了 Java ME 手机游戏开发的开发环境、基础知识和相关开发技术以及手机游戏编程的基本思想和思路。

本书共十章，分三个模块设计。第一个模块为入门的基础知识讲解，包括第 1、2 章；第二个模块为 Java ME 理论基础知识的应用，包括图形用户界面、相关组件、信息保存等综合内容，并辅以一个小实训项目来具体分析，本模块涵盖第 3、4、5、6 章；第三个模块为 Java ME 高级应用模块，在这个模块里面，通过一个市面上常见的手机游戏类型的综合项目阐述，让读者能基本掌握 Java ME 手机游戏开发的相关技术和开发流程，本模块主要包括第 7、8、9、10 章。

本书适合高职高专软件技术、计算机应用及相关专业学生学习使用，也适合作为具有其他语言或者平台游戏开发经验并且想使用 Java ME 开发游戏的游戏开发者参考。

图书在版编目（CIP）数据

Java ME 手机应用程序开发 / 易灿，李志勇编著
-- 北京 ：中国水利水电出版社，2017.3
高职高专院校“十三五”精品示范系列教材. 软件技术专业群
ISBN 978-7-5170-5102-2

Ⅰ. ①J… Ⅱ. ①易… ②李… Ⅲ. ①JAVA语言－程序设计－高等职业教育－教材②移动通信－携带电话机－应用程序－程序设计－高等职业教育－教材 Ⅳ. ①TP312.8②TN929.53

中国版本图书馆CIP数据核字(2017)第013560号

责任编辑：周益丹　　加工编辑：陈宏华　　封面设计：李　佳

书　名	高职高专院校“十三五”精品示范系列教材（软件技术专业群） Java ME 手机应用程序开发 Java ME SHOUJI YINGYONG CHENGXU KAIFA
作　者	易　灿　李志勇　编　著 刘彦姝　主　审
出版发行	中国水利水电出版社 （北京市海淀区玉渊潭南路 1 号 D 座　100038） 网址：www.waterpub.com.cn E-mail：mchannel@263.net（万水） sales@waterpub.com.cn 电话：（010）68367658（营销中心）、82562819（万水）
经　售	全国各地新华书店和相关出版物销售网点
排　版	北京万水电子信息有限公司
印　刷	北京瑞斯通印务发展有限公司
规　格	184mm×240mm　16 开本　16.25 印张　357 千字
版　次	2017 年 3 月第 1 版　2017 年 3 月第 1 次印刷
印　数	0001—3000 册
定　价	34.00 元

I

前　言

每一个人都会玩游戏，都喜欢玩游戏；但并不是每一个人都会开发制作游戏。要让游戏无处不在，还需要更多的人学会开发游戏。相信好学的你在玩过几个好玩的游戏后也会问：这个游戏是怎么做出来的？我是否也可以做出同样出色的游戏？我该如何学习游戏的制作？…… 游戏有很多种——本书中会讲到——在当今的 2016 年如果想找一种游戏，它只需要一个人业余花费很少的时间和精力就可以制作出，而且有可能会成为非常受欢迎的游戏，我想非移动游戏莫属！

Java ME 是 Sun 公司提供的移动应用开发平台。自从 Sun 公司发布 Java ME 以来，Java ME 技术便引起了软件开发商、信息服务商的极大关注，超过 500 家公司签订了使用 Java ME 的协议。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 ME 技术的手机。现在，有越来越多的人意识到了 Java ME 技术的开发与应用带来的无限机遇。本书主要面向有一定 Java 基础的开发人员和高校学生。

本书作为 Java ME 移动游戏（主要是手机游戏）制作的入门读物，只要你具备 Java 编程的基础知识并且了解一些 Java ME 的背景知识，通过本书的学习，你就能开发出自己的游戏。当然，最最重要的还是读者你的创意！

本书由三个模块组成，下面分别介绍：

第一个模块为基础模块，包含第 1～2 章，本模块主要介绍 Java ME 手机游戏开发相关的基础知识和编程环境搭建。本模块主要由易灿编写。

第二个模块为基础应用模块，本模块主要介绍 Java ME 在游戏开发方面应用的相关知识。其中第 3、4 章由李志勇编写，第 5、6 章由刘彦姝编写。

第三个模块包含第 7、8、9、10 章，本模块主要通过开发一个完整的手机游戏来阐述 Java ME 手机游戏开发的整体思想和相关技术的应用，由易灿编写。

本书的全部代码均在 JDK1.6+WTK2.2 环境下调试通过，并在 WTK 自带的模拟器上能够正确运行。本书代码仅供学习 Java ME 手机游戏开发的编程人员和学习者使用，欢迎读者对书中不当之处提出批评建议。

本书是国家骨干高等职业院校重点建设项目研究成果之一，由易灿、李志勇编著，刘彦姝主审，适用读者对象是高职高专软件技术、计算机应用及相关专业学生，也可作为具有其他语言或者平台游戏开发经验并且想使用 Java ME 开发游戏的游戏开发者参考。

编　者

2016 年 12 月

II

目 录

1

Java ME 概述

本章主要介绍 Java ME 的相关背景知识。读者需要掌握以下知识点：

- Java ME 的 3 层体系结构。
- CLDC 类库。
- MIDP 2.0 的新特性。
- CLDC/MIDP 总体体系结构。

1.1 Java ME 体系结构

为了适应移动数据的发展，推进无线电子商务等业务的发展，J2ME（Java 2 Micro Edition）即用于嵌入式系统的 Java 被引入无线领域（又称 Java ME）。Java ME 的出现实际上是 Java 技术的回归。作为 Java 2 平台的一部分，Java ME 与 J2SE、J2EE 一起，为无线应用的客户端和服务器端建立了完整的开发和部署环境。随着 Java ME 的应用，它为移动互联引入了一种新的模型，即允许手机从互联网上下载各种应用程序，并在手机上创造可执行环境离线运行这些程序。由于定义了可执行程序下载的标准，并在手机上创立了可执行环境和程序开发语言，由此，在移动通信业第一次为软件开发商创造了巨大的商机，手机用户在得到丰富应用体验的同时，也大大提高了运营商的网络流量。

1.1.1 Java 的版本

Java 在 10 多年的发展历程中，已经成长为一个全面而成熟的面向对象应用程序开发平台，它适用于广泛的、异构的编程环境，这些应用的涉及面非常广，从企业级的服务器应用到传统

的桌面应用以及各式各样面向小型设备的嵌入式应用。

Java 2 平台包括 3 个版本，每个版本都针对不同的用户群，如图 1-1 所示。这 3 个版本具体为：

- Java 2 平台企业版（J2EE）：用于企业级分布式系统的设计与开发。
- Java 2 平台标准版（J2SE）：用于传统 PC 桌面应用程序开发。
- Java 2 平台微型版（J2ME）：主要面向消费电子产品和嵌入式设备的软件开发。

说明：Java SDK 1.2 及以后的版本都统一改名为 Java 2，因此这些名字中都带有 2。

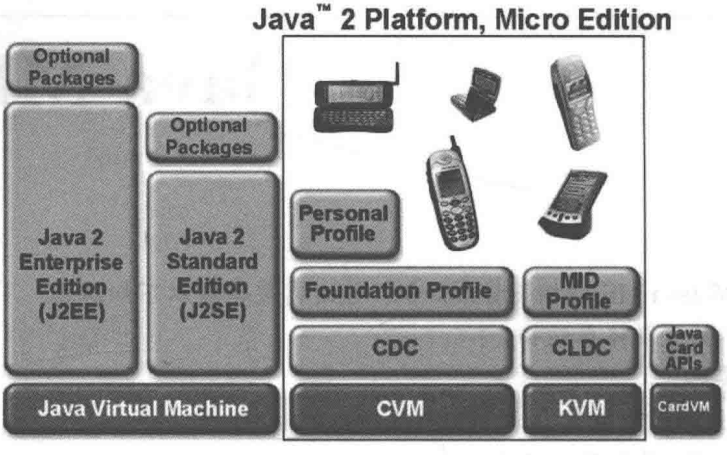

图 1-1　Java 的版本结构

Java 2 面向市场的每一个版本都有其自己的虚拟机，这些虚拟机都为其目标应用做了特别优化。

1.1.2　Java ME 的 3 层体系结构

Java ME 用于为信息家电市场提供应用服务，这些信息家电包括传呼机、移动电话、像 Palm 这样的个人数字助手（PDA）、电视机顶盒、POS 终端以及其他的消费电子设备，而且每一种家电设备又有不同的特性和界面。

为了满足消费者和嵌入式市场不断发展和多样化的需求，Java ME 体系结构采用模块化、可扩展的设计。这种设计是通过一个 3 层软件模型来实现的，该模型构建于本地操作系统之上。

Java ME 的 3 层体系结构依照各种设备的资源特性，将 Java ME 技术架构分为简表（Profile）、配置（Configuration）和 Java Virtual Machine（JVM）3 层，然后再进一步细分，这使 Java ME 能够在每一类设备的限制下工作，同时能够提供最低限度的 Java 语言功能性，如图 1-2 所示。

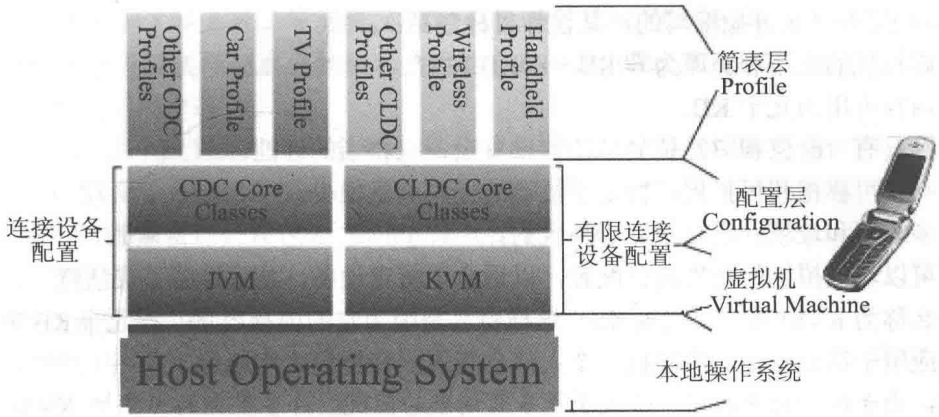

图 1-2 Java ME 的分层结构

- Java 虚拟机（JVM）层：这一层是针对设备本地操作系统定制的 Java 虚拟机的实现，支持特定的 Java ME 配置，就像使用所有 Java 技术一样，Java ME 的核心也在一种虚拟机中。
- 配置（Configuration）层：面对的是大量各种不同的小型嵌入式设备，它们在外观和功能上均各不相同。Java ME 对这些设备进行分类，将一些共性提取出来形成适合于某个范畴中设备可用的规范，称为“配置”。读者也可以将配置理解为对硬件的描述，所以通过定义配置的方法就能够清楚地描述硬件功能。
- 简表（Profiles）层：简表层定义了特定系列设备上可用的应用程序编程接口（API）的最小集。简表在一个特定的配置上面实现。应用程序是针对特定的简表编写的，因此可以移植到支持该简表的任何设备上。另外，一个设备可以同时支持多个简表。用户和开发人员看到最多的就是这一层。

Java 虚拟机是 Java ME 技术的核心，配置和简表则提供特殊环境的类应用程序接口。配置是用于一组通用设备的最小的 Java 平台，而简表则为具体的设备家族或特别的应用程序提供更具体的功能。

Java 虚拟机和构建于此虚拟机之上的配置规范一起代表了某一类设备的基本能力，而更进一步的设备分类上的区别是通过简表层提供的 API 实现的。为了满足更多新的应用的需要，简表可以通过扩充类库来强大自己的功能。

1.1.3 虚拟机（KVM）

KVM 是一个专门为小型、资源受限设备所设计的紧凑的、便携的 Java 虚拟机，是用于 Java ME 平台最小的虚拟机，并且是用于 CLDC 配置的虚拟机。Java ME 应用程序并不一定非要使用 KVM，Java ME 技术支持使用任何虚拟机，不过至少应当有 KVM 这样的功能。

KVM 是完全从头开始编写的，其设计目标包括：

- 虚拟机的大小和类库为 60KB～80KB 左右。
- 内存占用为几十 KB。
- 在具有 16 位和 32 位处理器的设备上，有相当的性能。
- 高度可移植和可扩展，特定于机器的代码总量很少。
- 多线程和垃圾回收是独立于系统的。
- 可以对虚拟机的组件进行配置，以适合于特定设备，从而增强了灵活性。

为什么称为 KVM 呢？K 代表 Kilo。这样命名是因为它的内存容量是用几十 KB 来衡量的。其典型地应用于移动电话、传呼机、个人信息管理器和便携式终端等。

说明：由于这一历史原因，造成了很多名词上的困扰。许多早期的文章把 KVM 与 com.sun.kjava 包合称为 KVM，来共同表示这一技术。而现在所说的 KVM 单指 Sun 的 CLDC 虚拟机引用实现。甚至还有人用 K-Java 作为 Java ME 的代称，表示基于 KVM 的 Java，这很容易和正式的 Java ME 标准以及早期的 com.sun.kjava 包相混淆。读者在看这类文章时需要注意文章的时间和背景，并采用严格的正式名称。

KVM 实现所需的最小内存空间大约是 128KB，包括虚拟机、最小的库和运行 Java 应用所需要的堆空间。一个更加典型的实现需要总共 256KB 内存空间，其中 32KB 作为应用运行时的堆空间，60KB～80KB 用于虚拟机本身，剩余的为类保留。

1.2 有限连接设备配置表（CLDC）

CLDC 提供了一个适合于小型、资源受限、连接受限设备上使用的标准 Java 平台。这种设备有移动电话、PDA 等。为此，它在每个方面都进行了大量的优化。它的虚拟机很小，并且不支持 Java 语言的一些特性。

1.2.1 CLDC 概览

CLDC 起源可以追溯到 1999 年 JavaOne 大会上介绍的 Sun 的第一个袖珍版 Java 和第一个 KVM 以及相关的类库，虽然 CLDC 和所有的配置都满足成为虚拟机的条件，但是它本身还不是虚拟机，CLDC 的引用实现只是包含在当前分布中的 KVM。

根据规范，运行 CLDC 的设备应该有：

- 至少 192KB 的内存空间。
- 16 或 32 位处理器。
- 一个有限的电源供给（通常使用电池）。
- 有限的或间断的网络连接性（9600 bps 或更少）。
- 多样化的用户界面甚至没有用户界面。

说明：由于增加了浮点运算等功能，CLDC 的内存需求从 1.0 的 160KB 上升到 1.1 的 192KB。其中 160KB 用于存储实际的虚拟机和本身的类库，CLDC 规范假定应用程序能够在 32KB 这样小的 Java 堆栈空间中运行。

通常说来，这个配置是为个人化的、移动的、有限连接信息设备而设计的，如传呼机、移动电话和 PDA 等。与 J2SE 相比，CLDC 缺少下列特征：

- AWT（抽象窗口开发包），Swing 或其他图形库。
- 用户自定义类装载器。
- 类实例的最终化。
- 弱的引用（在 CLDC1.1 中已经引入）。
- RMI。
- Reflection（映射）。

如果要理解为什么 CLDC 去除这么多 J2SE 中重要的类和特征，可以回顾一下与 CLDC 相关的两条基本原理。首先，它只有 512KB 的内存空间，而 RMI 和映射需要的内存太大。其次，配置必须满足为一组通用设备提供最小的 Java 平台。在个人移动信息设备领域中，许多系统都不能支持 J2SE 中的众多高级特征。例如，许多系统没有或不提供访问一个文件系统的功能或权限，因此与文件有关的类也被丢弃了。而且错误处理是一个代价非常高的过程处理，在许多消费电子设备中，故障恢复是很难的甚至是不可能的。所以在 CLDC 中，许多错误处理类也被删除了。

1.2.2 CLDC 中使用的 J2SE 类

CLDC 包含一个很小的 J2SE 子集，因为其他的类对虚拟机和本地运行环境的依赖性都比较大，而且会消耗大量的资源，因此被忽略掉了。下面介绍 CLDC 支持的类。

1. java.lang 中的类

（1）系统类

J2SE 类库中包含了几个同 Java 虚拟机密切相关的类，经常使用的几个 Java 工具需要有几个类的支持，如标准的 Java 编译器（javac）要求 String 和 StringBuffer 中的特定方法是可用的。所以 CLDC 中支持的系统类主要包括以下几个：

- java.lang.Object
- java.lang.Class
- java.lang.Runtime
- java.lang.System
- java.lang.Thread
- java.lang.Runnable（接口）
- java.lang.String

- java.lang.StringBuffer
- java.lang.Throwable

（2）数据类型类

CLDC 中同样支持 J2SE 中的基本数据类型，这些类都是 J2SE 相应类型的子集。它们是：

- java.lang.Boolean
- java.lang.Byte
- java.lang.Short
- java.lang.Integer
- java.lang.Long
- java.lang.Float（从 1.1 版本开始支持）
- java.lang.Double（从 1.1 版本开始支持）
- java.lang.Character

（3）错误类

- java.lang.Error
- java.lang.NoClassDeFoundError（从 1.1 版本开始支持）
- java.lang.OutOfMemoryError
- java.lang.VirtualMachineError

（4）异常类

- java.lang.Exception
- java.lang.ArithmeticException
- java.lang.ArrayIndexOutOfBoundsException
- java.lang.ArrayStoreException
- java.lang.ClassCastException
- java.lang.ClassNotFoundException
- java.lang.IllegalAccessException
- java.lang.IllegalArgumentException
- java.lang.IllegalMonitorStateException
- java.lang.IllegalThreadStateException
- java.lang.IndexOutOfBoundsException
- java.lang.InstantiationException
- java.lang.InterruptedException
- java.lang.NegativeArraySizeException
- java.lang.NullPointerException
- java.lang.RuntimeException
- java.lang.NumberFormatException

- java.lang.SecurityException
- java.lang.StringIndexOutOfBoundsException

（5）弱引用

- java.lang.ref.Reference（从 1.1 版本开始支持）
- java.lang.ref.WeakReference（从 1.1 版本开始支持）

2. java.io 中的类

（1）输入/输出类

- java.io.InputStream
- java.io.OutputStream
- java.io.ByteArrayInputStream
- java.io.ByteArrayOutputStream
- java.io.DataInput（接口）
- java.io.DataOutput（接口）
- java.io.DataOutputStream
- java.io.DataInputStream
- java.io.Reader
- java.io.Writer
- java.io.InputStreamReader
- java.io.OutputStreamReader
- java.io.PrintStream

（2）在 java.io 中还有几个相对于输入/输出处理的异常类

- java.io.EOFException
- java.io.InterruptedIOException
- java.io.IOException
- java.io.UnsupportedEncodingException
- java.io.UTFDataFormatException

3. java.util 中的类

（1）集合容器类

- java.util.Vector
- java.util.Stack
- java.util.Hashable
- java.util.Enumeration（接口）

（2）日历和时间类

CLDC 还包含了 J2SE 中相对于时间和日期的类，例如 java.util.Calendar、java.util.Date 和 java.util.TimeZone 类的子集。为了节省空间和其他硬件资源，CLDC 规范中仅要求电子设备支

持一个时区，是否支持其他时区以及具体支持什么时区可以由硬件设备制造商来决定。以下为3个日历和时间类：

- java.util.Calendar
- java.util.Date
- java.util.TimeZone

（3）异常类

CLDC 中的异常除了 java.lang 和 java.io 两个包之外，还包括 java.util 的异常。

- java.util.EmptyStackException
- java.util.EmptyNoSuchElementException

4. 其他附加类

除了上面所描述的几个集合的类外，CLDC 中还提供了两个附加类。其中，java.util.Random 是一个简单的随机数生成类，可以由比较简单的算法生成随机数，而 java.lang.Math 是一个数学函数库的类，其提供的方法如 min、max 等可以用于基本的数学函数运算。在 CLDC 1.1 中，java.lang.Math 类还支持浮点三角函数和平方根运算，并增加了其他一些工具函数，如 ceil 和 floor。

- java.util.Random
- java.lang.Math

1.2.3 CLDC 专用类

J2SE 和 J2EE 类库通过 java.io 和 java.net 包提供的一系列功能可用来处理访问存储和网络系统的输入和输出。尽管如此，保证所有这些功能都适合只有几百 KB 存储空间的小型设备是很困难的。于是 CLDC 针对 J2SE 网络和 I/O 类进行抽象、泛化，这些新系统的一个目标就是 J2SE 类的严格子集。这样才能容易地映射到通用的底层硬件或者任何 J2SE 实现，而且保证具有较好的扩展性、灵活性和一致性，以支持新的设备和协议。

一般来说，资源受限设备在对网络和存储类库的要求上差异非常明显。例如，需要处理包交换网络的设备制造商通常会需要基于数据报的通信机制，而需要处理线路交换网络的设备制造商则需要基于流的连接。考虑到严格的内存限制，如果制造商支持了特定类型的网络功能，通常都不希望再支持其他机制。所有这些都使得设计 CLDC 的网络设施非常具有挑战性，特别是由于 Java ME 配置不允许定义可选的特性。而且，如果存在多种网络机制和协议，则可能会使应用程序开发者非常困扰，特别是当他们不得不自己处理底层协议时。

于是，CLDC 采用“通用连接框架（GCF）”，取代了 J2SE 中的网络 API，为 CLDC 提供网络连接功能。所有 Java ME/CLDC 应用程序都有权访问 HTTPS 功能，但在 MIDP 1.0 规范中并不正式需要 HTTPS 支持。考虑到 HTTPS 在移动商业中显而易见的重要性，许多 MIDP 设备供应商已经将对 HTTPS 的支持添加到它们自己的 MIDP 运行时实现中。Sun Microsystems 也

在其 Java ME Wireless Toolkit 版本 1.0.2 及后续版本中添加了 HTTPS 支持。在 MIDP 2.0 规范中，HTTPS 支持将成为正式需求。CLDC 1.0 规范（CLDCSPEC）为 MIDP 网络定义了一个抽象的通用连接框架。该框架定义了各种网络 API 的接口格式，但没有对任何网络协议作具体实现。

该框架支持的功能包括：

- 各种基于流的连接。
- 各种数据报。

按照 MIDP 1.0 规范（MIDPSPEC），基于 CLDC 通用连接框架的 MIDP 实现至少需要实现 HTTP1.1 上（RFC2616）基于流的各种客户端连接。一个 MIDP 实现也可以实现 CLDC 通用连接框架中其他各种网络 API（例如 TCP/IP 套接字、数据报等）。但是，因为 MIDP 1.0 规范仅将 HTTP 网络协议定义为基本配置，所以为了 MIDlets 的完全可移植性，不应使用 MIDP 规范中除 HTTP 网络 API 之外的其他 API。

1.2.4 CLDC 1.1 的新特性

CLDC 1.1（JSR139）专家组成员对 CLDC Specification 1.0 版基本满意，他们认为在新的规范中不需要做什么根本上的修改。因此，CLDC Specification 1.1 版基本上只是一个增补版，并且是对 CLDC Specification 1.0 版完全向后兼容的。一些重要的新功能，如对浮点的支持，被加入到这个新版本中。

CLDC 1.1 相对于 CLDC 1.0（JSR30）版本本质上没有变化，只是随着硬件水平的不断提高，CLDC 1.1 在兼容性和可用性上做了如下一些改进：

- 增加了对浮点数据的支持。
- CLDC 1.1 支持所有跟浮点运算相关的字节码指令。
- 由于允许使用浮点运算，设备的最小内存从 160KB 提高到 192KB。
- 加入了 Float 和 Double 两个类，其他类库中的类也增加了一些方法以处理浮点数。
- 增加了对弱引用（Weak Reference）的支持（只是 J2SE 的弱引用类的一个小子集）。
- 更加明确了对错误处理的要求，增加了一个新的错误类 NoClassDefFoundError。
- Thread 对象同 J2SE 中的线程一样被命名。
- 引入了 Thread.getName()方法来获知线程的名字，并且 Thread 类从 J2SE 继承了几个新的构造函数。
- 线程可以被 Thread.interrupt()方法中断。
- Calendar、Date 和 TimeZone 类被重新设计。
- 增加了一份更详细的验证器规范作为 CLDC Specification 1.1 版的附录（《CLDC Byte Code Typechecker Specification》）。

类库中还加入了以下方法用于处理浮点数据：