

j2me的基本介绍 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/220/2021\\_2022\\_j2me\\_E7\\_9A\\_84\\_E5\\_9F\\_BA\\_c97\\_220770.htm](https://www.100test.com/kao_ti2020/220/2021_2022_j2me_E7_9A_84_E5_9F_BA_c97_220770.htm) 计算机等级考试训练软件《百宝箱》4。深入理解CLDC 本节的部分内容译自J2ME White Paper，比较枯燥，但请记住，毕竟，正确的观念重于一切。CLDC的目的为小型的，且资源有限，Connected Limited的设备定义一个标准的Java平台。允许动态的向这些设备发布内容与应用程序。方便第三方软件厂商与开发者向这些设备移植内容与应用程序。CLDC的范围 Java语言与虚拟机的特性 核心Java类库（如java.lang.\*，java.util.\*） 输入/输出 对网络的支持 对安全性的支持 对国际化的支持 CLDC不包括的范围（即由MIDP解决的） 对应用程序life-cycle的管理（就是指一个应用程序是如何安装，运行，删除的） UI界面（User Interface） Event处理 高级应用程序模式（这里指用户与应用程序的交互）。CLDC的安全性 低端的基于虚拟机的安全性由下载时对class文件的审核（verification）步骤来保证。（如果你写过并编译过MIDlet程序，就会知道这一步是必须的。这里有一个概念就行了，我会在以后讲述如何创建MIDlet程序时，详细介绍具体的步骤）。基于应用程序的安全性通过使每个应用程序分别运行在相对独立的封闭环境中来保证并且处于被保护的system package中的类不能被应用程序重载（overridden）。对于Java语言支持的特殊的方面 CLDC总的来说是与Java语言完全兼容的，但需注意以下几点：不支持浮点数据类型（即没有float和double）（这实际上主要是由于CLDC所面向的设备，其硬件因其内存限制或是基于安全性的考虑不支持浮

点运算) 不支持类实例 ( class instance ) 的终结 ( finalization ) , 即不存Object.finalize()方法。(注: finalization指的是java中提供的有别与garbage collection的另外的一种对象清理方法。具体可参看侯捷老师译的Bruce Eckel的《Thinking in Java 第二版》, 可以在www.jjhou.com中下载) 有限的错误处理, 即大部分Java.lang.Error的子类都未被支持。与此相反, CLDC包括了相当完备的异常 ( exception ) 类。对于Java虚拟机 ( Java Virtual Machine ) 支持的特殊的方面 实质上CLDC在其严格的内存限制下达到了对Java虚拟机相当程度的兼容性。不过还是有下列不同点。不支持浮点数据类型 ( 没有float和double ) 不支持JNI ( the Java Native Interface ) 不支持用户自定义的Java级的类载入器 ( class loaders ) 没有反射(reflection)特性 ( 注: reflection指java通过java.lang.reflect提供的, 可以对类的能力进行分析的功能。常用来对类的结构进行检查, 在JavaBeans上有广泛运用可参见Gary Cornell的Core Java2 volumel,II, 机工有出中译本 ) 不支持线程组 ( thread groups ) 或守护线程 ( daemon threads ) 不支持类实例 ( class instance ) 的终结 ( finalization ) 没有弱参考 ( weak references ) ( 注: weak reference提供了一种解决指向已经被garbage collector清除的对象的方法。在Java2中被介绍。具体请参考java.lang.ref API 文档, 和sun网站上关于Reference Objects和Garbage Collection的文章。 ) 有限的错误处理 ( error handling )

100Test 下载频道开通, 各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)