Java编程技术中汉字问题的分析及解决 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/273/2021_2022_Java_E7_BC_96_E7_A8_8B_c104_273318.htm 在基于 Java 语言的编程中，我们经常碰到汉字的处理及显示的问题。一大堆看不懂的乱码肯定不是我们愿意看到的显示效果，怎样才能够让那些汉字正确显示呢？Java语言默认的编码方式是UNICODE，而我们中国人通常使用的文件和数据库都是基于GB2312或者BIG5等方式编码的，怎样才能够恰当地选择汉字编码方式并正确地处理汉字的编码呢？本文将从汉字编码的常识入手，结合Java编程实例，分析以上两个问题并提出解决它们的方案。 现在 Java 编程语言已经广泛应用于互联网世界，早在 Sun 公司开发 Java 语言的时候，就已经考虑到对非英文字符的支持了。Sun 公司公布的 Java 运行环境（JRE）本身就分英文版和国际版，但只有国际版才支持非英文字符。不过在 Java 编程语言的应用中，对中文字符的支持并非如同 Java Soft 的标准规范中所宣称的那样完美，因为中文字符集不只一个，而且不同的操作系统对中文字符的支持也不尽相同，所以会有许多和汉字编码处理有关的问题在我们进行应用开发中困扰着我们。有很多关于这些问题的解答，但都比较琐碎，并不能够满足大家迫切解决问题的愿望，关于 Java 中文问题的系统研究并不多，本文从汉字编码常识出发，分析 Java 中文问题，希望对大家解决这个问题有所帮助。 汉字编码的常识 我们知道，英文字符一般是以一个字节来表示的，最常用的编码方法是 ASCII 。但一个字节最多只能区分256个字符，而汉字成千上万，所以现在都以双字节来表示汉字，为了能够与英文

字符分开，每个字节的最高位一定为1，这样双字节最多可以表示64K格字符。我们经常碰到的编码方式有 GB2312、BIG5、UNICODE 等。关于具体编码方式的详细资料，有兴趣的读者可以查阅相关资料。我肤浅谈一下和我们关系密切的 GB2312 和 UNICODE。GB2312 码，中华人民共和国国家标准汉字信息交换用编码，是一个由中华人民共和国国家标准总局发布的关于简化汉字的编码，通行于中国大陆地区及新加坡，简称国标码。两个字节中，第一个字节（高字节）的值为区号值加32（20H），第二个字节（低字节）的值为位号值加32（20H），用这两个值来表示一个汉字的编码。UNICODE 码是微软提出的解决多国字符问题的多字节等长编码，它对英文字符采取前面加“ 0 ”字节的策略实现等长兼容。如“ A ”的 ASCII 码为0x41，UNICODE 就为0x00，0x41。利用特殊的工具各种编码之间可以互相转换。 Java 中文问题的初步认识 我们基于 Java 编程语言进行应用开发时，不可避免地要处理中文。Java 编程语言默认的编码方式是 UNICODE，而我们通常使用的数据库及文件都是基于 GB2312 编码的，我们经常碰到这样的情况：浏览基于 JSP 技术的网站看到的是乱码，文件打开后看到的也是乱码，被 Java 修改过的数据库的内容在别的场合应用时无法继续正确地提供信息。String sEnglish =“ apple ”.String sChinese =“ 苹果 ”.String s =“ 苹果 apple ”. sEnglish 的长度是5，sChinese的长度是4，而 s 默认的长度是14。对于 sEnglish来说， Java 中的各个类都支持得非常好，肯定能够正确显示。但对于 sChinese 和 s 来说，虽然 Java Soft 声明 Java 的基本类已经考虑到对多国字符的支持（默认 UNICODE 编码），但是如果操

作系统的默认编码不是 UNICODE，而是国标码等。从 Java 源代码到得到正确的结果，要经过“ Java 源代码-> Java 字节码-> .虚拟机->操作系统->显示设备 ”的过程。在上述过程中的每一步骤，我们都必须正确地处理汉字的编码，才能够使最终的显示结果正确。“ Java 源代码-> Java 字节码 ”，标准的 Java 编译器 javac 使用的字符集是系统默认的字符集，比如在中文 Windows 操作系统上就是 GBK ,而在 Linux 操作系统上就是ISO-8859-1，所以大家会发现在 Linux 操作系统上编译的类中源文件中的中文字符都出了问题，解决的办法就是在编译的时候添加 encoding 参数，这样才能够与平台无关。用法是 javac ?Cencoding GBK。“ Java 字节码->虚拟机->操作系统 ”， Java 运行环境 （JRE） 分英文版和国际版，但只有国际版才支持非英文字符。 Java 开发工具包 （JDK） 肯定支持多国字符，但并非所有的计算机用户都安装了 JDK 。很多操作系统及应用软件为了能够更好的支持 Java ，都内嵌了 JRE 的国际版本，为自己支持多国字符提供了方便。“ 操作系统-> 显示设备 ”，对于汉字来说，操作系统必须支持并能够显示它。英文操作系统如果不搭配特殊的应用软件的话，是肯定不能够显示中文的。 还有一个问题，就是在 Java 编程过程中，对中文字符进行正确的编码转换。例如，向网页输出中文字符串的时候，不论你是用out.println(string). 还是用，都必须作 UNICODE 到 GBK 的转换，或者手动，或者自动。在 JSP 1.0中，可以定义输出字符集，从而实现内码的自动转换。用法是 但是在一些 JSP 版本中并没有提供对输出字符集的支持，（例如 JSP 0.92），这就需要手动编码输出了，方法非常多。最常用的方法是String s1 = request.getParameter(“ keyword

”).String s2 = new String(s1.getBytes(“ ISO-8859-1 ”), ” GBK ”). getBytes 方法用于将中文字符以“ ISO-8859-1 ”编码方式转化成字节数组，而“ GBK ”是目标编码方式。我们从以ISO-8859-1方式编码的数据库中读出中文字符串 s1，经过上述转换过程，在支持 GBK 字符集的操作系统和应用软件中就能够正确显示中文字符串 s2 。 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