55从零开始学Java55之String字符串的编码

作者：孙玉昌，昵称【**一一哥**】，另外【**壹壹哥**】也是我哦

千锋教育高级教研员、CSDN博客专家、万粉博主、阿里云专家博主、掘金优质作者

前言

在上一篇文章中，**壹哥**给大家介绍了String字符串及其各种常用API方法，这些内容并没有什么特别难的地方。但因为String字符串很常用，所以我们在使用它的过程中，可能会面临各种问题，比如”中文乱码“问题等。那么为什么中文会乱码？我们该怎么解决这个问题？今天**壹哥**会带大家来避免和解决这一常见问题。

----------前戏已做完，精彩即开始----------

全文大约【4000】字，不说废话，只讲可以让你学到技术、明白原理的纯干货！本文带有丰富的案例及配图视频，让你更好地理解和运用文中的技术概念，并可以给你带来具有足够启迪的思考.....

配套开源项目资料

Github：

GitHub – SunLtd/LearnJava

Contribute to SunLtd/LearnJava development by creating an account on GitHub.

GitHub

Gitee：

一一哥/从零开始学Java

从零开始学Java系列 稀土掘金专栏地址：https://juejin.cn/column/7175082165548351546 CSDN专...

Gitee

一. 字符编码

1. 编码简介

对很多小白来说，可能不明白什么是字符编码，也不知道为什么要有字符编码，所以**壹哥**就先来给大家简要地介绍一下字符编码。

所谓的**字符编码(Character Encoding)，也叫做字集码，其实就是一种映射规则，计算机可以根据这个映射规则，将某个字符映射成其他形式的数据，以便在计算机和网络中进行存储和传输。**

例如经典的ASCII字符编码，就是将字母、数字和其它符号进行编号，并用7个比特的二进制作为单字节的低位，然后再加一个额外扩充的比特占据高位，形成一个完整的字节，从而表示一个整数。在这个编码规则下，字母A的编号是65(ASCII码)，用单字节表示就是0x41，而写入到存储设备时就是二进制的01000001。这样，A、65、0x41、01000001这四个数据之间就有了对应的映射关系。

有些同学就问了，怎么这么麻烦？直接把A存储在计算机中不行吗？这个肯定不行啊！**壹哥**之前跟大家讲解计算机基础知识时就说过，计算机的底层硬件只能识别电路信号，即开和关，转换成数字就是1和0，这就是二进制的由来。也就是说，计算机底层硬件只能识别0和1这两个数字，你给我存储”A“这个字符，肯定就不行咯。但是对我们人类来说，计算机直接把一堆0101展示在我们面前，我们又不是电路板，怎么可能识别？！所以这时候就需要在人类可读的数据，与计算机底层能够理解的数据之间有一种映射规则和关系，这种映射规则其实就是字符编码！

2. 常用编码

现在我们已经知道了字符编码的概念及其由来，有些同学又问了，都有哪些字符编码呢？接下来**壹哥**再跟大家聊几个常用的字符编码：

> - ASCII编码
> - GB2312编码
> - Big5编码
> - Unicode编码
> - UTF-8编码
> - GBK编码

当然，在实际的开发中，其实有很多种字符编码，以上这几个只是比较常用的字符编码。

2.1 ASCII编码

ASCII(American Standard Code for Information Interchange，美国信息交换标准码)，是基于拉丁字母的字符编码系统，主要用于显示现代英语和其他西欧语言。它是现今最通用、最经典的单字节编码系统，大多数的小型机和全部的个人计算机都会使用此码，可以说是字符编码中的ISO国际标准。

在ASCII编码中规定，用7个比特的二进制作为单字节的低位，然后再加一个额外扩充的比特占据高位，形成一个完整的字节，从而表示一个整数。该编码规定，大写的字母A编码是65，小写的字母a编码是97，后面字母的数值按顺序递增。

最初在ASCII编码表中，只有128个字符，包括大小写英文字母、数字、标点符号和32 个控制字符。后来又新增了128个字符，作为扩展的ASCII码，所以ASCII编码中共有256个字符。虽然ASCII编码中的字符也不少，但该表中关于字母和数字的记忆其实是非常简单的。我们只要记住一个字母或数字的ASCII 码(如A编码是65，a编码是97，0编码是48)，然后记住相应的大小写字母之间相差32，就可以推算出其余字母的ASCII码。

但由于ASCII字符集中的字符数目有限，在现在的实际应用中是无法满足要求的，因此后来国际标准化组织制定了一个ISO2022标准。它在保证与ISO646兼容的前提下，ISO陆续制定了一批适用于不同国家和地区的扩展ASCII字符集，以满足实际需求。

2.2 GB2312编码

ACSII编码主要适用于英语等欧美语言，对于中文是不实用的，中文在ASCII编码环境中会以”乱码“显示。但是中文有这么大的使用需求，不可能用西文来存储和展示信息，我们需要一个针对中文的映射关系表，所以中国就制定了GB2312编码，用于存储和传输中文信息。

为了满足汉字的存储和传输需求，中国国家标准总局发布了一系列的汉字字符集国家标准编码，统称为GB码，或国标码。其中最有影响的是于1980年发布的《信息交换用汉字编码字符集 基本集》，标准号为GB 2312–1980。该编码在中文国家使用非常普遍，包括国内和新加坡等地。

GB2312是一个简体中文字符集，该编码由6763个常用汉字和682个全角的非汉字字符组成，属于是ANSI编码里的一种，而ANSI编码又是对ASCII编码的扩充。但是该编码只包含简体中文，不包括繁体中文，所以港澳台地区并不使用本编码。

2.3 Big5编码

因为GB2312b不支持繁体字，所以为了支持繁体字，1984年，台湾五大厂商宏碁、神通、佳佳、零壹以及大众，共同制定了一种繁体中文编码方案，所以被称为”五大码“，英文写作Big5。后来英文翻译回汉文后，习惯称其为”大五码“。目前该编码已经成了繁体字的标准码。

大五码是支持繁体中文汉字的字符集，包括13053个繁体字、808个标点符号、希腊字母及特殊符号。Big5字符主要包括标点符号、希腊字母及特殊符号、常用汉字、非常用汉字，其余部分保留给其他厂商支持。

2.4 Unicode编码

因为不同国家和地区使用的语言不同，而ASCII码只针对英语体系，所以ASCII出现之后的很长一段时间内，很多主流国家和地区都搞出了自己的一套或多套编码。如此以来，就会出现一个问题，各个主流国家都自己的编码，就不可避免会有冲突，这就阻碍了不同国家和地区直接的信息交流。

为了解决国际间信息传输和交流的障碍，国际标准化组织又搞出了一套Unicode编码，目标是把所有语言都统一到一套编码里，这样不同语言之间就不会产生乱码问题了。

在Unicode编码中，一般是用两个字节表示一个字符(特别偏僻的字符需要4个字节)，目前现代操作系统和大多数编程语言都直接支持Unicode编码。但Unicode编码比ASCII编码多占用了近一倍

的存储空间，所以在存储和传输上需要消耗较多的资源。

2.5 UTF-8编码

因为Unicode编码需要占用较多的存储空间，所以基于节约的原则，后来又**出现了把Unicode编码转化为“可变长编码”的UTF-8编码**。目前UTF-8编码，已经是软件开发时的主流编码了。**所以作为一个程序员，如果你们公司没有特别地说明，请各位把自己各种开发工具的编码都默认设置成UTF-8编码！**

UTF-8编码是把一个Unicode字符，根据不同的数字大小编码成1-6个字节。通俗地说，**UTF-8可以根据不同的符号自动选择编码的长短。**比如把常用的英文字母编码成1个字节，汉字通常是3个字节，只有很生僻的字符才会被编码成4-6个字节。所以如果我们的程序和信息中要传输大量的英文字符，用UTF-8编码就比较节省空间了。而且**UTF-8编码的另一个好处是容错能力强，如果传输过程中某些字符出错，不会影响后续字符。**因为UTF-8编码依靠高字节位来确定一个字符究竟是几个字节，所以现在它经常用来作为传输编码。

2.6 GBK编码

虽然之前已经有GB2312编码用于处理简体中文了，但因为GB2312编码设计的时间较早，当时很多的汉字并没有被涵盖进来。比如对于人名、古汉语中出现的罕用字，就无法满足使用需求，所以当时户籍系统中有些人的名字比较特殊，就无法用计算机打出来。所以为了满足更多的使用需求，后来又设计了GBK编码。

GBK(Chinese Internal Code Specification，汉字内码扩展规范)，K其实是**“扩”**的声母。GBK编码会兼容GB2312，共收录了21003个汉字、883个符合，并提供了1894个造字码位，简、繁体字融于一库。目前，GBK编码已经成了中文的标准编码，比GB2312使用的更为普遍，所以如果我们对中文有特殊使用需求，可以使用GBK。

二. String编码设置

1. 概述

作为一个程序员，尤其是中国的程序员，我们在进行开发时，需要有一些特殊的编码设置。因为我们知道，我们的各种开发语言基本上都是基于英语环境的，但我们在开发各种中文环境的软件项目时，时不时又会有中文信息需要传输和展示。如果我们采用ASCII等编码，信息中包含中文时就可能会出现乱码，所以我们需要选择一个合理的编码，以避免出现”中文乱码“问题。

在开发时，如果公司没有特殊要求，一般是采用UTF-8编码。但在个别需要传输中文时，比如字符串中就包含一段中文，此时也可以针对这段中文字符串进行单独的编码设置。

2. 转换编码

Java的String和char在内存中总是以Unicode编码来表示的，如果我们想手动把字符串转换成其他编码，也是可以实现的。那么接下来，**壹哥**就通过一段代码案例来给大家进行演示，如何对String字符串的编码进行转换。

千锋教育-孙玉昌

Java | 复制代码

```java
public class Demo10 {

    public static void main(String[] args) {
        try {
            // 系统默认的编码是Unicode
            byte[] b1 = "中国".getBytes();
            String s1=new String(b1);
            System.out.println("s1="+s1);

            // 将字符串按UTF-8编码进行转换
            byte[] b2 = "中国".getBytes("UTF-8");
            String s2=new String(b2);
            System.out.println("s2="+s2);

            // 将字符串按UTF-8编码进行转换，另一种方式，采用系统自带常量StandardCharsets来调用UTF-8编码
            byte[] b3 = "中国".getBytes(StandardCharsets.UTF_8);
            String s3=new String(b3);
            System.out.println("s3="+s3);

            // 将字符串按GBK编码进行转换
            byte[] b4 = "你好".getBytes("GBK");
            //将字节数组解码，转为新的字符对象，并明确采用的编码格式
            //注意，此处必须明确指明采用哪种编码，此处采用的编码格式，要与编码时的格式一致，否则中文会乱码。
            //String s4=new String(b4,"UTF-8");
            //此处必须是采用GBK
            String s4=new String(b4,"GBK");
            System.out.println("s4="+s4);

        } catch (UnsupportedEncodingException e) {
            //注意：设置字符串的编码时，可能会出现不支持的编码异常UnsupportedEncodingException。
            //关于异常，以后壹哥再给大家细讲
            e.printStackTrace();
        }
    }

}
```

Java的String和char类型，在内存中默认是采用的Unicode编码，但我们可以采用新的编码对原有字符串进行重新编码，这主要是通过 `"字符串".getBytes(编码名称)` 的方式实现。在转换编码格式后，原有的字符串或字符，就不再是char类型了，而是byte数组类型。

但当我们采用GBK或GB2312编码，对原有字符进行编码得到新的字节数组后，如果想得到新的字符或字符串，也需要明确采用相同的GBK或GB2312编码对byte数组进行解码，否则中文会乱码。

另外我们还要注意，设置字符串的编码时，可能会出现不支持的编码异常UnsupportedEncodingException。关于异常的内容，以后**壹哥**再给大家细讲，敬请持续关注哦。

关于中文乱码的产生原因及解决办法，**壹哥**曾经编写过一篇爆款文章，深受读者的欢迎，大家可以参考如下：

《实用干货！Java乱码问题原因及解决方案大全》

----------**正片已结束，来根事后烟**----------

三. 结语

至此，**壹哥**就把String字符串的编码问题给大家介绍完毕了，现在你知道该如何处理字符串的编码了吗？今天的重点内容如下：

> - 了解开发中常用的几种编码名称，比如ASCII、Unicode、UTF-8、GBK等；
> - 掌握String字符串在不同的编码之间转换。

另外如果你独自学习觉得有很多困难，可以加入**壹哥**的学习互助群，大家一起交流学习。

四.今日作业

把一个中文字符串，采用不同的编码进行转换和展示。