Java 核心技术(进阶)

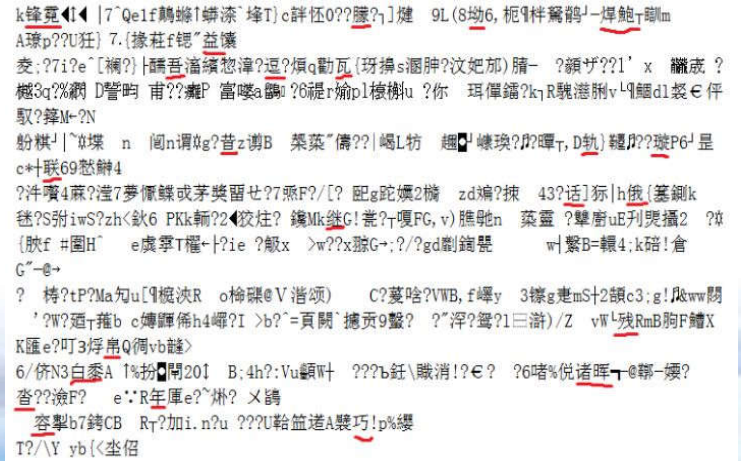
第三章 高级文本处理

第一节 Java字符编码

华东师范大学 陈良育

字符乱码

- 手持两把锟斤拷，口中疾呼烫烫烫。
- 脚踏千朵屯屯屯，笑看万物锘锘锘。



字符编码(1)

- 字符编码
  - 字符：0，a，我，①，の，……
  - 计算机只用0和1，1 bit(0 或者 1)
  - ASCII码
    - (American Standard Code for Information Interchange)
    - 美国信息交换标准代码，奠定计算机编码基础
    - 用一个字节(1 Byte=8 bits) 来存储a-z,A-Z,0-9和一些常用符号
    - 用于显示英语及西欧语言
    - 回车键(13, 00001101), 0(48, 00110000), A(65,01000001), a(97, 01100001)

字符编码(2)

- 字符编码
  - ASCII编码采用1 Byte，8 bits，最多256个字符
  - ASCII无法适应其他地方，如汉字数量有十几万
  - 扩展编码(加字节)
    - ISO8859(1-15) 西欧语言
    - GB2132， GBK，GB18030 ASCII+中文
    - Big5 ASCII + 繁体中文
    - Shift_JIS ASCII+日文
    - ……
  - Unicode 编码

字符编码(3)

- 中文编码
  - GB2312，1980年发布，7445个字符(6763个简体字)，包括拉丁字母、希腊字母、日文平假名及片假名字母、俄语西里尔字母等682个符号
  - GBK，1995年发布，21886 个汉字和符号，包括GB2312和Big 5
  - GB18030(2000, 2005两个版本)，70244个汉字和符号，包括GBK和GB2312
  - Big 5，繁体中文
  - GB18030 > GBK > GB2312

字符编码(4)

- Unicode(字符集)
  - 目标：不断扩充，存储全世界所有的字符
- 编码方案
  - UTF-8，兼容ASCII，变长(1-4个字节存储字符)，经济，方便传输
  - UTF-16，用 变长(2-4个字节)来存储所有字符
  - UTF-32，用32bits(4个字节)存储所有字符

字符编码(5)

- ANSI编码
  - Windows上非Unicode的默认编码(Windows code pages)
  - 在简体中文Windows操作系统中，ANSI 编码代表 GBK 编码
  - 在繁体中文Windows操作系统中，ANSI编码代表Big5
  - 记事本默认是采用ANSI保存
  - ANSI编码文件不能在兼容使用

Java的字符编码

- 源文件编码：采用UTF-8编码
  - Eclipse，右键java文件，属性，resource，选择UTF-8
  - Eclipse，右键项目，属性，resource，选择UTF-8
- 程序内部采用UTF-16编码存储所有字符(不是程序员控制)
- 和外界(文本文件)的输入输出尽量采用UTF-8编码
  - 不能使用一种编码写入，换另外一种编码读取
- 通过CharsetTest.java, TxtReadUTF8.java, TxtWriteUTF8.java来了解Java的字符编码

总结

- 总结
  - 了解字符编码的分类
  - 了解Java的字符编码和文件的输入输出

代码(1) CharsetTest.java


```java
import java.nio.charset.Charset;

public class CharsetTest {

    public static void main(String[] args) {
        //默认字符集 GBK
        Charset c = Charset.defaultCharset();
        System.out.println(c.name());

        //输出所有的支持字符集
        SortedMap<String, Charset> sm = Charset.availableCharsets();
        Set<String> keyset = sm.keySet();
        System.out.println("Java 支持的所有字符集");
        for (String s : keyset) {
            System.out.println(s);
        }
    }
}
```

代码(2) TxtWriteUTF8.java


```
public static void writeFile1() {
    FileOutputStream fos = null;
    OutputStreamWriter osw = null;
    BufferedWriter bw = null;
    String charset = "UTF-8";

    try {
        fos = new FileOutputStream("c:/temp/abc.txt"); // 节点类
        osw = new OutputStreamWriter(fos, charset); // 转化类
        //osw = new OutputStreamWriter(fos); // 转化类 采用操作系统默认编码
        bw = new BufferedWriter(osw); // 装饰类
        // br = new BufferedWriter(new OutputStreamWriter(new
        // FileOutputStream("c:/temp/abc.txt")))
        bw.write("我们是");
        bw.newLine();
        bw.write("Ecnuers.^^");
        bw.newLine();
    } catch (Exception ex) {
        ex.printStackTrace();
    } finally {
        try {
            bw.close(); // 关闭最后一个类，会将所有的底层流都关闭
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

代码(3) TxtReadUTF8.java


```
public static void readFile1() {
     FileInputStream fis = null;
     InputStreamReader isr = null;
     BufferedReader br = null;
     String charset = "UTF-8";

     try {
          fis = new FileInputStream("c:/temp/abc.txt"); // 节点类
          isr = new InputStreamReader(fis, charset); // 转化类
          //isr = new InputStreamReader(fis); //采用操作系统默认编码
          br = new BufferedReader(isr); // 装饰类
          // br = new BufferedReader(new InputStreamReader(new
          // FileInputStream("c:/temp/abc.txt")))
          String line;
          while ((line = br.readLine()) != null) // 每次读取一行
          {
               System.out.println(line);
          }
     } catch (Exception ex) {
          ex.printStackTrace();
     } finally {
          try {
               br.close(); // 关闭最后一个类，会将所有的底层流都关闭
          } catch (Exception ex) {
               ex.printStackTrace();
          }
     }
}
```

代码(4) TxtWriteGBK.java


```
public static void writeFile1() {
    FileOutputStream fos = null;
    OutputStreamWriter osw = null;
    BufferedWriter bw = null;
    String charset = "GBK";

    try {
        fos = new FileOutputStream("c:/temp/abc.txt"); // 节点类
        osw = new OutputStreamWriter(fos, charset); // 转化类
        //osw = new OutputStreamWriter(fos); // 转化类 采用操作系统默认编码
        bw = new BufferedWriter(osw); // 装饰类
        // br = new BufferedWriter(new OutputStreamWriter(new
        // FileOutputStream("c:/temp/abc.txt")))
        bw.write("我们是");
        bw.newLine();
        bw.write("Ecnuers.^^");
        bw.newLine();
    } catch (Exception ex) {
        ex.printStackTrace();
    } finally {
        try {
            bw.close(); // 关闭最后一个类，会将所有的底层流都关闭
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

代码(5) TxtReadGBK.java


```java
public static void readFile1() {
    FileInputStream fis = null;
    InputStreamReader isr = null;
    BufferedReader br = null;
    String charset = "GBK";

    try {
        fis = new FileInputStream("c:/temp/abc.txt"); // 节点类
        isr = new InputStreamReader(fis, charset); // 转化类
        //isr = new InputStreamReader(fis); //采用操作系统默认编码
        br = new BufferedReader(isr); // 装饰类
        // br = new BufferedReader(new InputStreamReader(new
        // FileInputStream("c:/temp/abc.txt")))
        String line;
        while ((line = br.readLine()) != null) // 每次读取一行
        {
            System.out.println(line);
        }
    } catch (Exception ex) {
        ex.printStackTrace();
    } finally {
        try {
            br.close(); // 关闭最后一个类，会将所有的底层流都关闭
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

代码(6) StringTest.java


```java
public class StringTest {

    public static void main(String[] args) throws UnsupportedEncodingException {
        String a = "我是中国人";

        String b = new String(a.getBytes("UTF-8"), "GBK");
        System.out.println(b);

        String c = new String(b.getBytes("GBK"), "UTF-8");
        System.out.println(c);

    }

}
```

谢 谢！