1. 轉換流涉及到的類：屬於字符流

*InputStreamReader：将一个字节的输入流转换为字符的输入流*

*解码：字节、字节数组 --->字符数组、字符串*

*OutputStreamWriter：将一个字符的输出流转换为字节的输出流*

*编码：字符数组、字符串 ---> 字节、字节数组*

說明：編碼決定了解碼的方式

1. 作用：*提供字节流与字符流之间的转换*
2. 圖示：



1. 典型實現：

*/\*  
此时处理异常的话，仍然应该使用try-catch-finally  
InputStreamReader的使用，实现字节的输入流到字符的输入流的转换  
\*/*

@Test  
 public void test1() throws IOException {  
 FileInputStream fis = new FileInputStream("dbcp.txt");  
*// InputStreamReader isr = new InputStreamReader(fis);//使用系统默认的字符集  
//参数2指明了字符集，具体使用哪个字符集，取决于文件dbcp.txt保存时使用的字符集* InputStreamReader isr = new InputStreamReader(fis,"UTF-8");*//使用系统默认的字符集* char[] cbuf = new char[20];  
 int len;  
 while((len = isr.read(cbuf)) != -1){  
 String str = new String(cbuf,0,len);  
 System.*out*.print(str);  
 }  
 isr.close();  
 }

*/\*  
此时处理异常的话，仍然应该使用try-catch-finally  
综合使用InputStreamReader和OutputStreamWriter  
 \*/*@Test  
public void test2() throws Exception {  
 *//1.造文件、造流* File file1 = new File("dbcp.txt");  
 File file2 = new File("dbcp\_gbk.txt");  
 FileInputStream fis = new FileInputStream(file1);  
 FileOutputStream fos = new FileOutputStream(file2);  
 InputStreamReader isr = new InputStreamReader(fis,"utf-8");  
 OutputStreamWriter osw = new OutputStreamWriter(fos,"gbk");  
 *//2.读写过程* char[] cbuf = new char[20];  
 int len;  
 while((len = isr.read(cbuf)) != -1){  
 osw.write(cbuf,0,len);  
 }  
 *//3.关闭资源* isr.close();  
 osw.close();  
}

1. 說明：

*文件編碼的方式(比如：ＧＢＫ)，決定了解析時使用的字符集（也只能是ＧＢＫ）*

1. 常見的編碼表



1. 對後面學習的啟示

客戶端/瀏覽器 <-----> 後台(java，go…) <---> 數據庫

要求前前后后使用的字符集都要統一：ＵＴＦ－８