1、转换流：属于字符流

1. InputStreamReader

1. 将一个字节的输入流转换为字符的输入流
2. 需要和InputStream “套接”
3. 构造器
    - public InputStreamReader InputStream in)
    - public InputSreamReader InputStream in,String charsetName
    - 如：Reader isr = new InputStreamReader(System.in ,” gbk “) ;

"gbk": 指定字符集

```
import java.io.File;
import java.io.FileInputStream;
```

```java
import java.io.InputStream;
import java.io.InputStreamReader;

/**
 * @author wangyimu
 * @Program InputStreamReader的使用
 * @create 2021-11-04-23:21
 */
public class InputStreamReaderTest {
 // InputStreamReader的使用，实现字节的输入流到字符的输入流的转换
 public static void main(String[] args) {
 File file = new File("dbcp.txt");
 try (InputStream is = new FileInputStream(file);
 InputStreamReader isr = new InputStreamReader(is,"utf-8")) {
 char[] buffer = new char[4096];
 int len;
 while((len = isr.read(buffer)) != -1){
 String str = new String(buffer,0,len);
 System.out.println(str);
 }
 }catch(Exception e){
 e.printStackTrace();
 }
 }
}
```

2. OutputStreamWriter

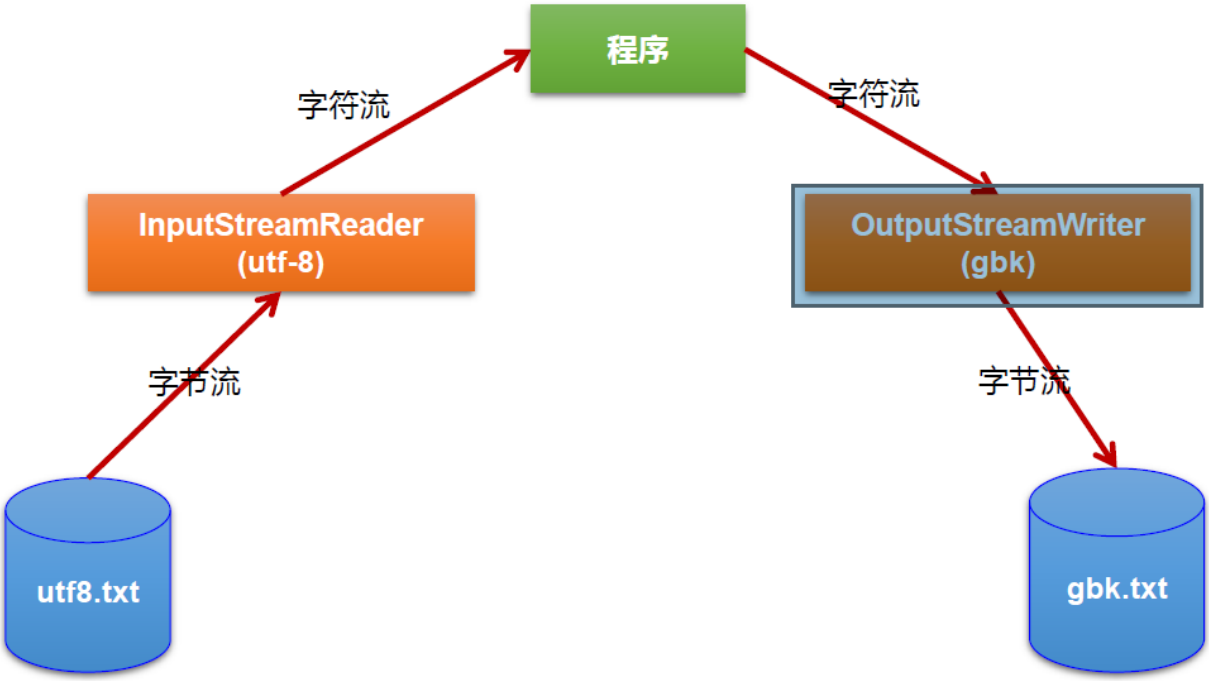
1.将一个字符的输出流转换为字节的输出流

2. 需要 和 OutputStream “套接”

3. 构造器

- public OutputStreamWriter(OutputStream out)

- public OutputSreamWriter (OutputStream out,String charsetName)




```java
import java.io.*;

/**
 * @author wangyimu
 * @Program 转换流的综合使用
 * @create 2021-11-04-23:35
 */
public class Test {
 public static void main(String[] args) {
  // 1.造文件，造流
  File file1 = new File("dbcp.txt");
  File file2 = new File("dbcp_gbk.txt");
  try (InputStream fis = new FileInputStream(file1);
  OutputStream fos = new FileOutputStream(file2);
  InputStreamReader isr = new InputStreamReader(fis,"utf-8");
  OutputStreamWriter osw = new OutputStreamWriter(fos, "gbk")) {
  // 2.读写过程
  char[] cubf = new char[40];
```

```
19  int len;
20  while((len = isr.read(cubf)) != -1){
21  osw.write(cubf,0,len);
22  osw.flush();
23  }
24  }catch(Exception e){
25  e.printStackTrace();
26  }
27  }
28 }
```

2、作用

提供字节流与字符流之间的转换

3、字符编码

1. 编码表的由来

计算机只能识别二进制数据，早期由来是电信号。为了方便应用计算机，让它可以识别各个国家的文字。就将各个国家的文字用数字来表示，并一一对应，形成一张表。这就是编码表。

2. 常见的编码表

1. ASCII: 美国标准信息交换码。

用一个字节的7位可以表示。

2. ISO8859-1: 拉丁码表。欧洲码表

用一个字节的8位表示

3. GB2312：中国的中文编码表。最多两个字节编码所有字符。

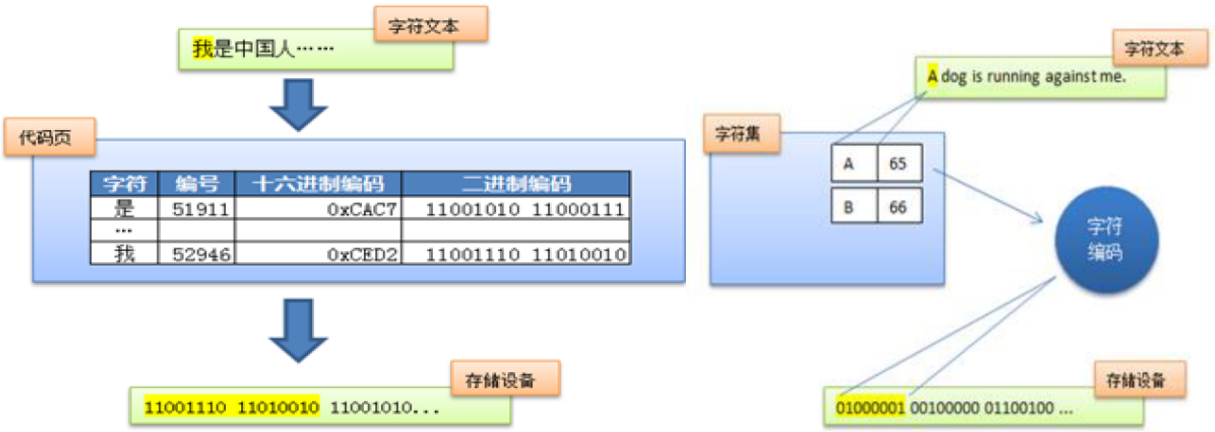
4. GBK：中国的中文编码表升级，融合了更多的中文 文字符号 。最多两个字节编码

5. Unicode：国际标准码， 融合 了目前人类使用的所有字符。为每个字符分配唯一的字符码。所有的文字都用两个字节来表示。

6. UTF-8：变长的编码方式，可用 1-4 个字节 来表示一个字符。

3. 说明

1. 在 Unicode 出现之前，所有的字符集都是和具体编码方案绑定在一起的（即字符集二分≈编码方式）），都是直接将字符和最终字节流绑定死 了



2. GBK等双字节编码方式， 用最高位是1或0表示两个字节和一个字节。

3. 面向 传输的众多 UTF UCS Transfer Format ）标准出现了，顾名思义 UTF-8 就是每次 8 个位传输数据，而 UTF-16 就是每次 16 个位 。 这 是为传输而设计的编码，并使编码无国界，这样就可以显示全世界上所有文化的字符了

4. Unicode 只是定义了一个庞大的、全球通用的字符集，并为每个字符规定了唯一确定的编号，具体存储成什么样的字节流，取决于字符编码方案。推荐的Unicode 编码是 UTF-8 和 UTF-16 。

```
Unicode符号范围   | UTF-8编码方式
(十六进制)          |      （二进制）
————————————————————————————
0000 0000-0000 007F | 0xxxxxxx（兼容原来的ASCII）
0000 0080-0000 07FF | 110xxxxx 10xxxxxx
0000 0800-0000 FFFF | 1110xxxx 10xxxxxx 10xxxxxx
0001 0000-0010 FFFF | 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
```

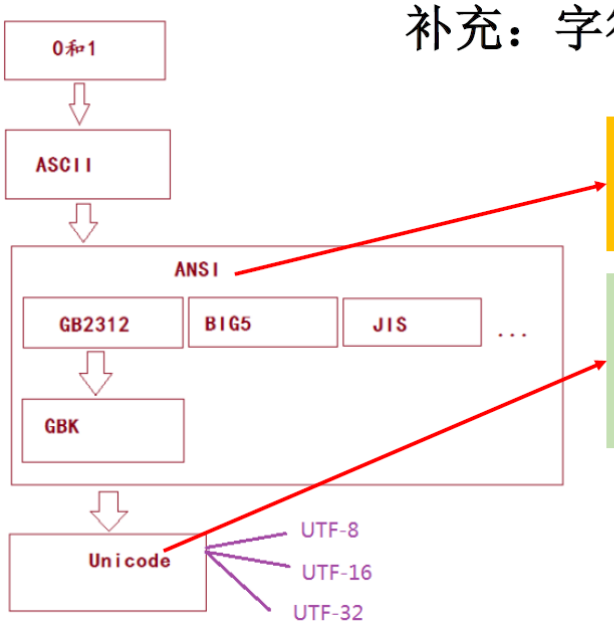
尚

Unicode编码值：23578　　十六进制 5C1A　　二进制　0101 1100 0001 1010

```
1110xxxx 10xx xxxx 10xx xxxx
1110 0101 1011 0000 1001 1010
```

UTF-8编码：　　1110 0101 1011 0000 1001 1010　　e5 b0 9a　　[-27, -80, -102]

补充：字符编码



ANSI编码，通常指的是平台的默认编码，例如英文操作系统中是ISO-8859-1，中文系统是GBK

Unicode字符集只是定义了字符的集合和唯一编号，Unicode编码，则是对UTF-8、UCS-2/UTF-16等具体编码方案的统称而已，并不是具体的编码方案。

4. 编码和解码

   编码：字符串=》字节数组

   解码：字节数组 =》字符串

5. 转换流的编码应用

   a. 可以将字符按指定编码格式存储

   b. 可以对文本数据按指定编码格式来解读

   c. 指定编码表的动作由构造器完成