1.所有字符流都是处理流封装字节流.

2.基本字符流:可以设置字符编码,解决乱码问题.还可以实现内容追加.

2.1:基本字符输入流:Reader->InputStreamReader

eg:public static void main(String[] args) throws IOException {

//声明流对象

InputStreamReader isr=null;

try {

//1.创建流对象

isr=new InputStreamReader(new FileInputStream("d:\\从前有座灵剑山.txt"),"utf-8");

/*2.用流来读取内容*/

//声明一个数

char[] c=new char[100];

//声明一个变量存读取的长度

int len;

//边读取边输出,返回读取的长度,读取的内容存在数组中了

while ((len=isr.read(c))!=-1) {

//将读取的内容转换字符串输出

String s1=new String(c, 0, len);

System.out.println(s1);

}

System.out.println("读取成功!");

} catch (Exception e) {

e.printStackTrace();

}finally {

//3.关流

if (isr!=null) {

isr.close();

}

}

```
    }
```

2.2:基本字符输出流:Writer->OutputStreamWriter

```
    eg:public static void main(String[] args) throws IOException {
        //声明流对象
        OutputStreamWriter osw=null;

        try {
            //1.创建流对象,可以实现追加内容,设置编码
            osw=new OutputStreamWriter(new
FileOutputStream("aa\\a2.txt",true), "utf-8");
            //2.用流对象向文件中写入内容
            osw.write("我是千锋人");
            osw.append("我爱千锋");
            System.out.println("写入成功");
        } catch (Exception e) {
            e.printStackTrace();
        }finally {
            //关流
            if (osw!=null) {
                osw.close();
            }
        }
    }
```

3.富二代字符流:FileReader,FileWriter,特点:使用方便,不没设置字符编码.

```
    eg:public static void main(String[] args) throws IOException {
        //创建流对象
        FileReader fr=null;
        FileWriter fw=null;

        try {
            //1.创建流对象
            fr=new FileReader("d:\\从前有座灵剑山.txt");
```

```
                fw=new FileWriter("aa\\a3.txt");
                /*2.用流对象调用方法边读取边写入*/
                //准备一个数组
                char[] c=new char[100];
                //声明变量存读取的长度
                int len;
                //返回读取的长度,读取的内容存到数组中
                while ((len=fr.read(c))!=-1) {
                    //将读取写入到文件中
                    fw.write(c,0,len);
                    //刷新
                    fw.flush();
                }
                System.out.println("拷贝成功");
            } catch (Exception e) {
                e.printStackTrace();
            }finally {
                //关流
                if (fr!=null) {
                    fr.close();
                }
                if (fw!=null) {
                    fw.close();
                }
            }
        }
```

4.字符缓冲流

4.1:字符缓冲输出流:BufferedWriter

```
        public static void main(String[] args) throws IOException {
        //声明流对象
        BufferedWriter bw=null;
        try {
            //1.创建流对象
```

```
                //bw=new BufferedWriter(new OutputStreamWriter(new
FileOutputStream("aa\\a4.txt",true), "utf-8"));
                bw=new BufferedWriter(new FileWriter("aa\\a4.txt",true));
                /*2.用流来向文件中写入内容*/
                bw.write("我是\n中国人");
                //换行,有两种方式,一种:\n换行,newLine()换行
                bw.newLine();
                //刷新
                bw.flush();
                //接着写入
                bw.append("我爱中国");
                //换行
                bw.newLine();
                //刷新
                bw.flush();
                System.out.println("写入成功");
            } catch (Exception e) {
                e.printStackTrace();
            }finally {
                //关流
                if (bw!=null) {
                    bw.close();
                }
            }
        }
```

4.2:字符缓冲输入流:BufferedReader

```
    eg:public static void main(String[] args) throws IOException {
            //声明流
            BufferedReader br=null;
            try {
                //1.创建流对象
                //br=new BufferedReader(new InputStreamReader(new
FileInputStream("aa\\a4.txt"), "utf-8"));
```

```
            br=new BufferedReader(new FileReader("aa\\a4.txt"));
            //2.用流对象一行一行的读取内容,前提写入文件中内容要有换行
            //声明一个变量存每次读取的内容
            String content;
            while ((content=br.readLine())!=null) {
                //读取一行输出一行
                System.out.println(content);
            }
            System.out.println("读取成功");
        } catch (Exception e) {
            e.printStackTrace();
        }finally {
            //3.关流
            if (br!=null) {
                br.close();
            }
        }
}
```