InputStreamReader/OutputStreamWriter用来实现将字节流转化成字符流。

System.in是字节流对象，代表键盘的输入，如果我们想按行接收用户的输入时，就必须用到缓冲字符流BufferedReader特有的方法readLine()，但是经过观察会发现在创建BufferedReader的构造方法的参数必须是一个Reader对象，这时候我们的转换流InputStreamReader就派上用场了。

/**

readLine()使用注意：

1.readLine()是一个阻塞函数。

2.readLine()只有在数据流发生异常或者另一端被close()掉时，才会返回null值。

3.如果不指定buffer大小，则readLine()使用的buffer有8192个字符。在达到buffer大小之前，只有遇到"/r"、"/n"、"/r/n"才会返回。

**/

而System.out也是字节流对象，代表输出到显示器，按行读取用户的输入后，并且要将读取的一行字符串直接显示到控制台，就需要用到字符流的write(String str)方法，所以我们要使用OutputStreamWriter将字节流转化为字符流。

* 编码：字符—-编码字符集———-》二进制

* 解码：二进制—解码字符集————》字符

出现乱码问题的原因：

1.编码和解码字符集不一致造成了乱码

2.字节的缺失，长度的丢失

使用InputStreamReader接收用户的输入，并输出到控制台

```
import  java.io.BufferedReader;
import  java.io.BufferedWriter;
```

```java
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;

public class TestConvertStream {
    public static void main(String[] args) {
        // 创建字符输入和输出流:使用转换流将字节流转换成字符流
        BufferedReader br = null;
        BufferedWriter bw = null;
        try {
            br = new BufferedReader(new InputStreamReader(System.in));
            bw = new BufferedWriter(new OutputStreamWriter(System.out));
            // 使用字符输入和输出流
            String str = br.readLine();
            // 一直读取，直到用户输入了exit为止
            while (!"exit".equals(str)) {
                // 写到控制台
                bw.write(str);
                bw.newLine();// 写一行后换行
                bw.flush();// 手动刷新
                // 再读一行
                str = br.readLine();
            }
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            // 关闭字符输入和输出流
            if (br != null) {
                try {
                    br.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
            if (bw != null) {
                try {
                    bw.close();
```

```
                } catch  (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```