****

****

**信息学院软件工程系**

**《JAVA程序设计》实验报告**

实验1

**姓名：马嘉昱**

**学号：37220222203719**

**学院：信息学院**

**专业：软件工程**

**完成时间：3。4**

**一、实验目的及要求**

实验目的：

1. 熟悉JDK安装、设置
2. 熟悉Eclipse/IDEA安装和使用
3. 写一个简单的Java程序

实验要求：

按照要求完成实现，并完成实验报告(参看《实验报告（参考格式）》）

**二、实验题目及实现过程**

题目1：安装并配置Java SE17

1. 实现过程：
   1. 进入浏览器直接搜索Java SE17，点击链接；
   2. 找到符合自己电脑架构的安装包，我的电脑是Mac M系列，所以下载Arm64版本；
   3. 之后按照默认安装完成安装即可。

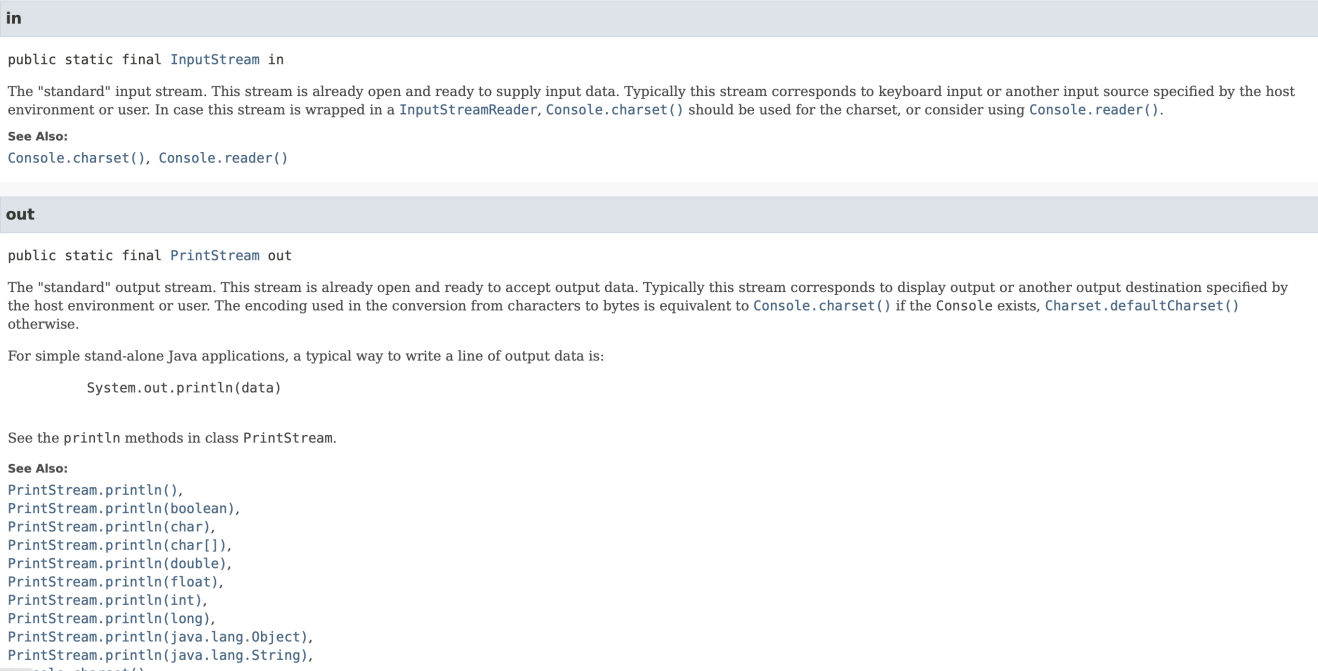
题目2：安装IDEA

1. 实现过程：
   1. 直接进入jetbrain官网下载IDEA即可。



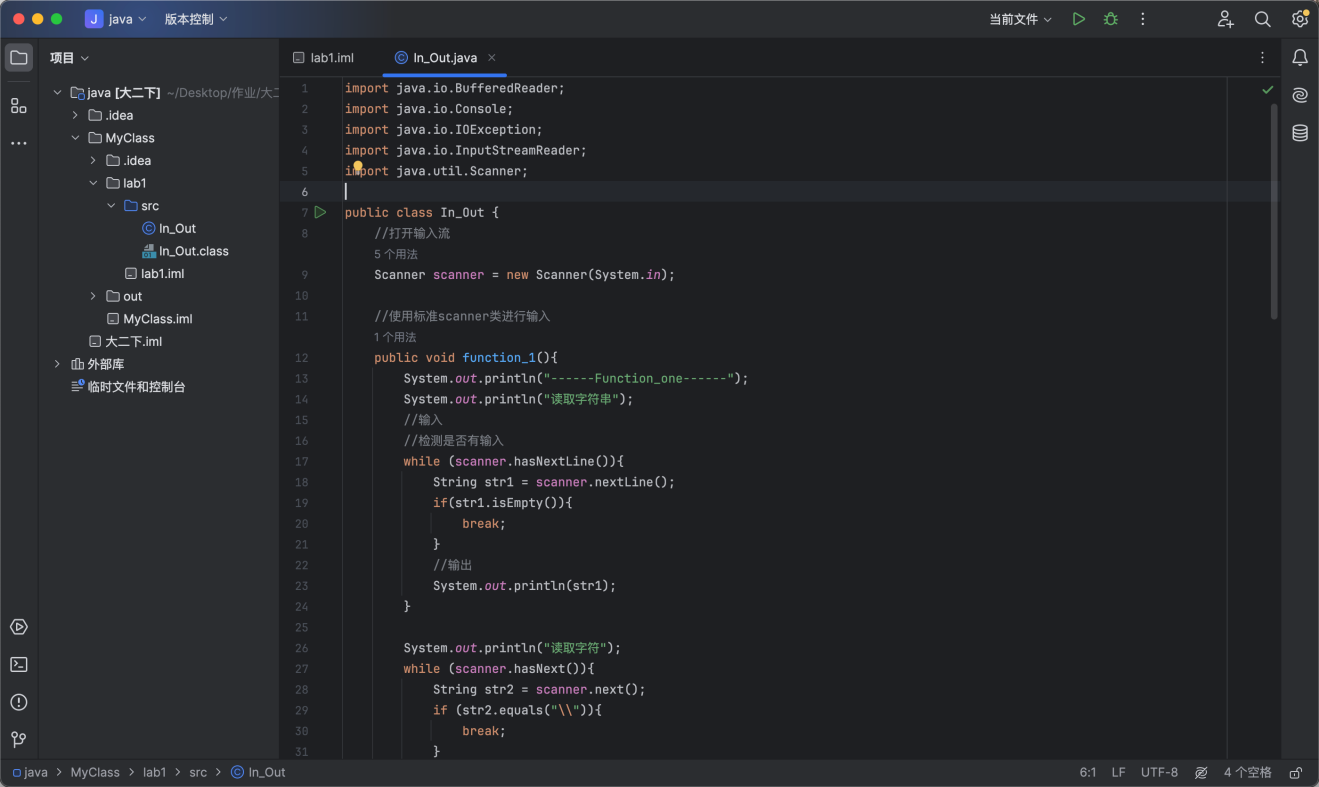
题目3:写一个简单的输入输出程序（尝试至少两种方式输入和输出）

1. 实验环境
   1. 操作系统：macOS
   2. 集成开发环境：IDEA
   3. jdk版本：java 17
2. 实现过程
   1. 查阅System文档https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/System.html

有关输入输出的内容如下：

* 1. 根据查阅到的资料，了解到与输入有关的方法有Scanner; System.in.read(); Console; BuffterRead这四种；输出只找到System.out.print这一种。将所有方法写成函数，通过对象调用。

1. 过程截图



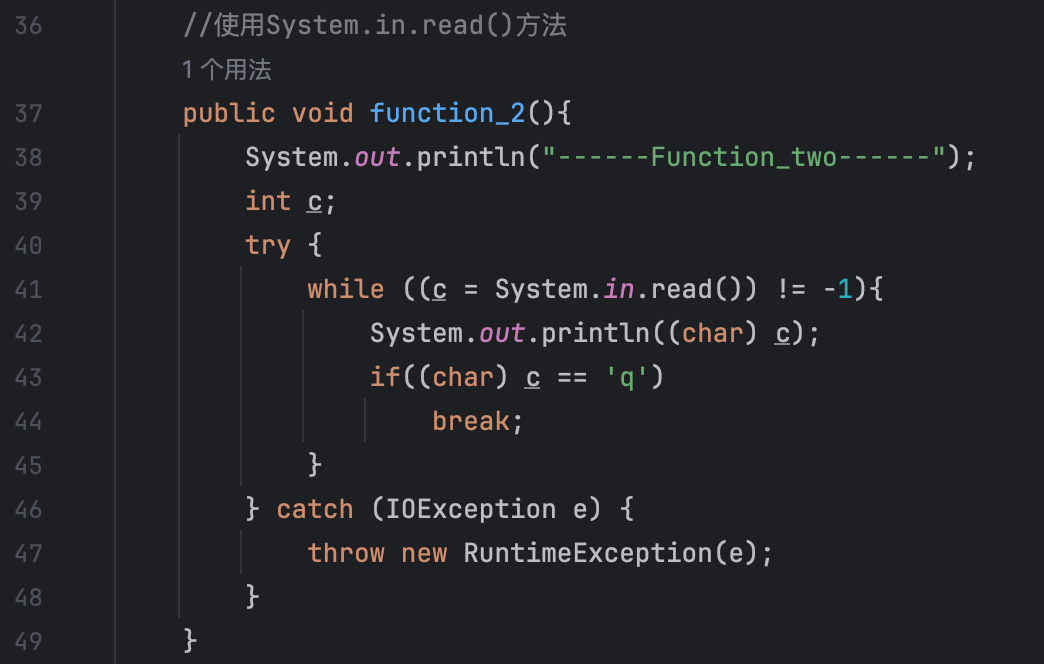
1. **Scanner方法：**

* Scanner直接从控制台或文件读取各种类型的数据。
* 直接创建scanner对象，使用nextline()方法读取行，next()读取字符。在该方法中分别写了读取字符串和读取字符的循环。当输入数据为空时跳出第一个循环；当输入数据为\时跳出第二个循环。
* nextLine()方法可以读取空格，next()不读取空格，将空格视为结束的标识。
* 运行结果截图：



1. **System.in.read()方法**

* 该方法是Java中最基本的输入方法之一，从标准输入流中读取单个字节的数据。一般用于处理字节流输入，对于处理文本文件等情况并不适用。当输入字符q时退出循环。



* 运行结果截图：

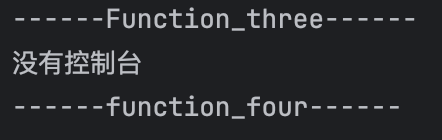


1. **Console方法：**

console方法是直接在控制台上运行的，在IDEA中会显示错误。Console用于从控制台读取密码或其他敏感信息，比如输入密码不想显示在屏幕上。函数中先判断是否在控制台环境下，之后再输入内容，当输入为空时退出循环。



IDEA运行结果：

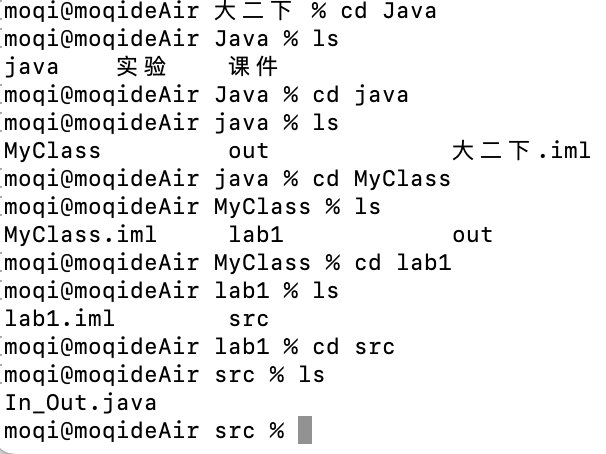
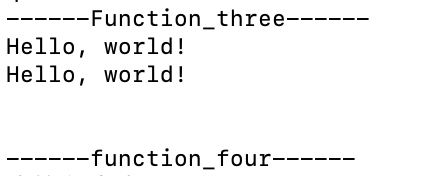


用命令行工具运行java文件：

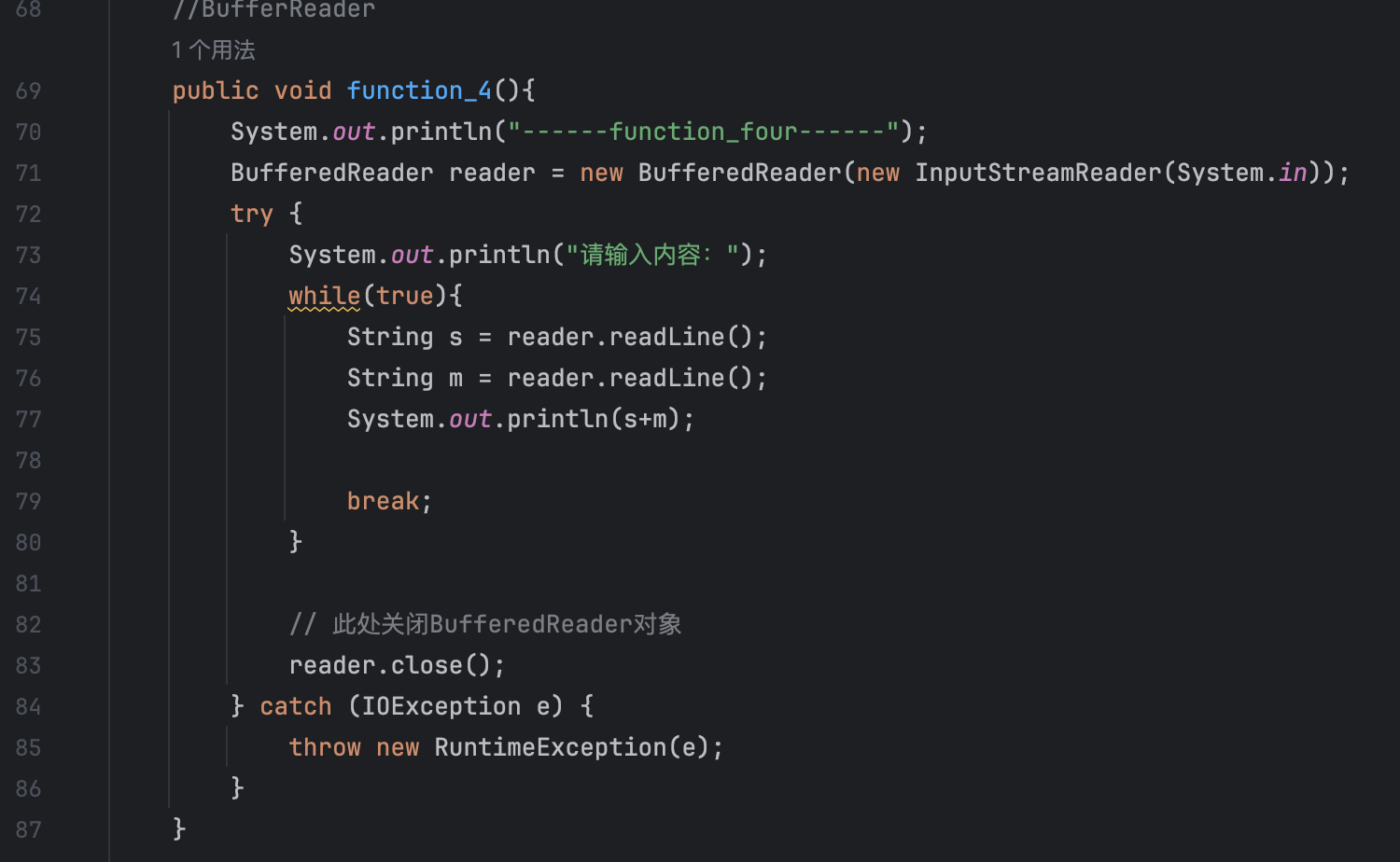
查阅相关资料：<https://blog.csdn.net/qq_42650988/article/details/81366388>

因为文件中没有package语句，所以直接：

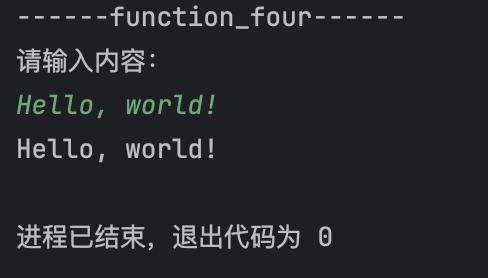


1. 先在命令行中找到java文件  
   
2. 输入javac In\_Out.java，可以看到同目录下出现In\_Out.class文件  
   
3. 输入java In\_Out.java运行文件，查看运行结果  
   
4. **BufferRead方法：**

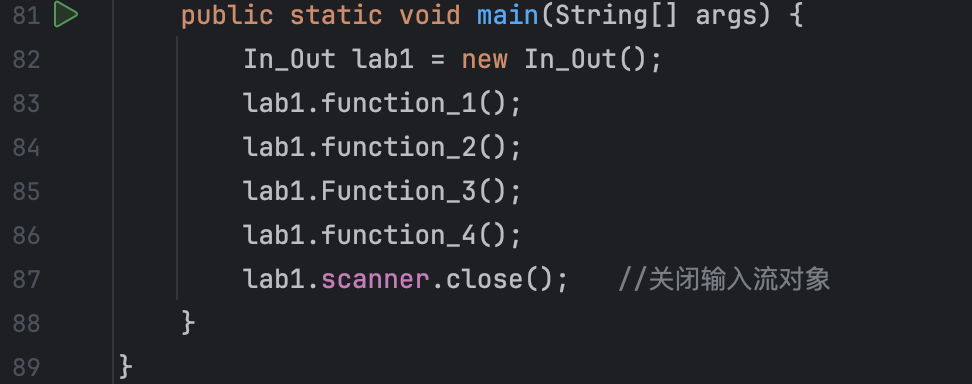
BufferedReader用于从字符输入流中读取文本，提供了缓冲功能，因此可以更有效地从输入流中读取数据。适合读取大量文本数据，例如从文件中读取文本，或者需要处理文本行的情况。这里因为上面使用了System.in.read()方法使得换行符保留在缓冲区中，所以第一个s读取的是换行符，第二个s才会读取输入的内容。



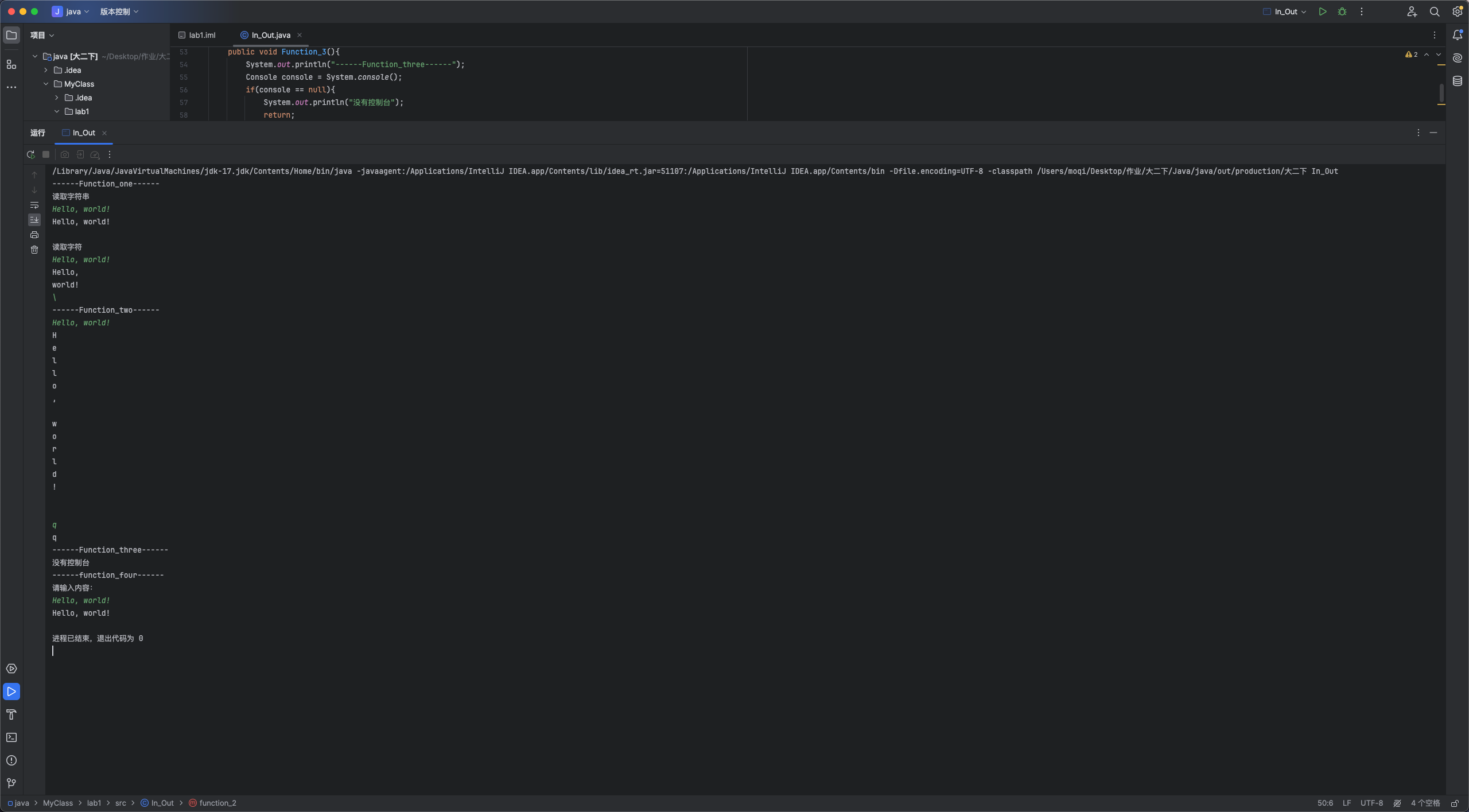
运行结果截图：

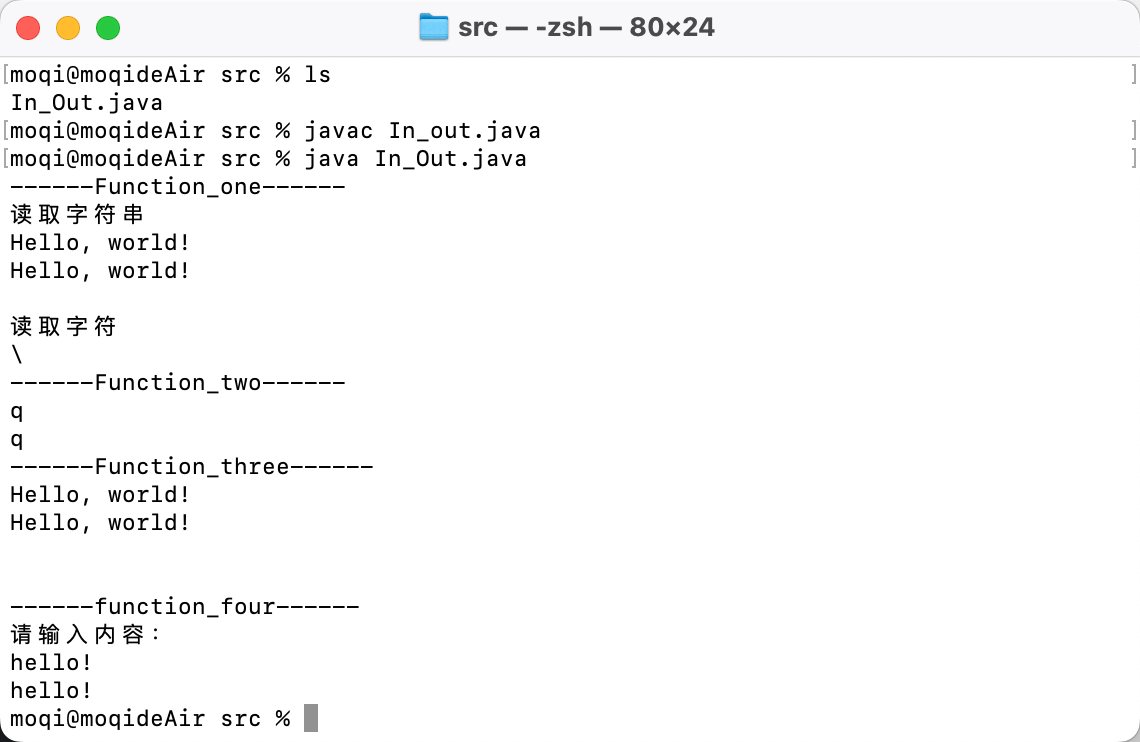


最后在主函数中调用这些方法：



运行结果截图：





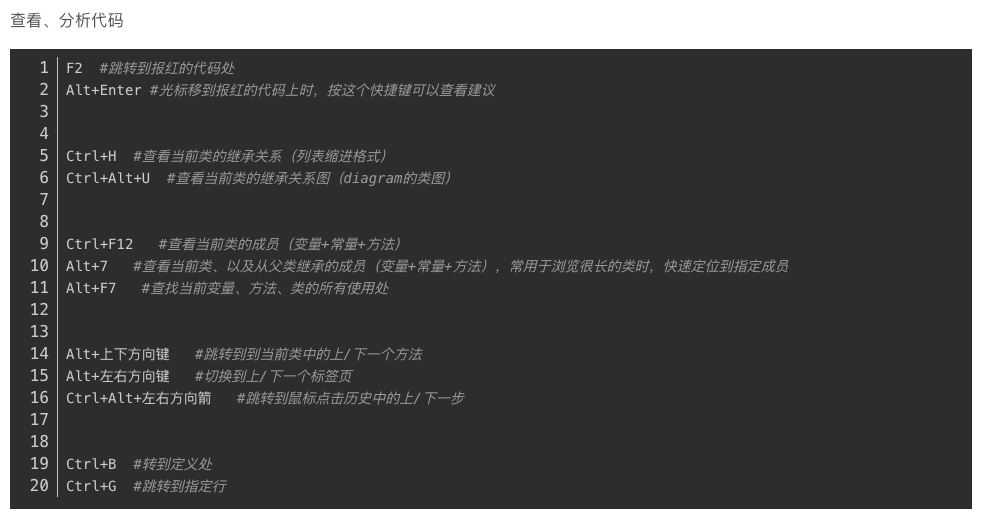
题目四：熟悉IDEA的使用

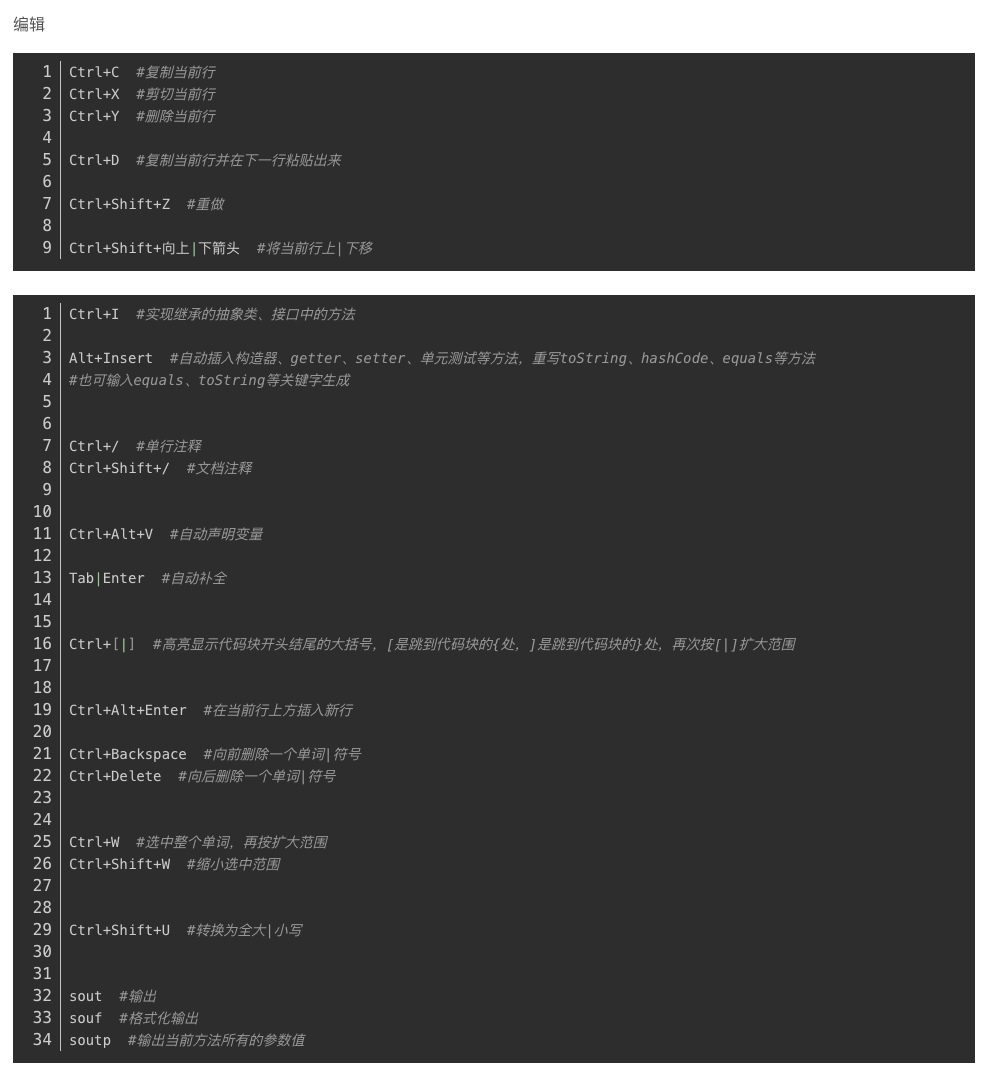
官方教程：<https://www.jetbrains.com/help/idea/getting-started.html>

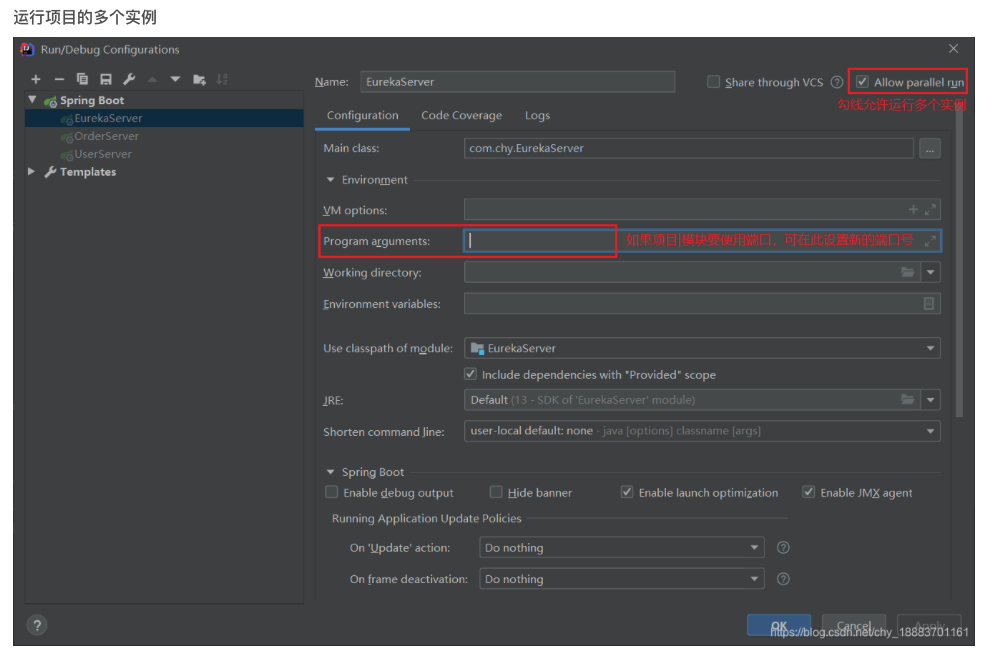
图文教学：<https://zhuanlan.zhihu.com/p/377258715>

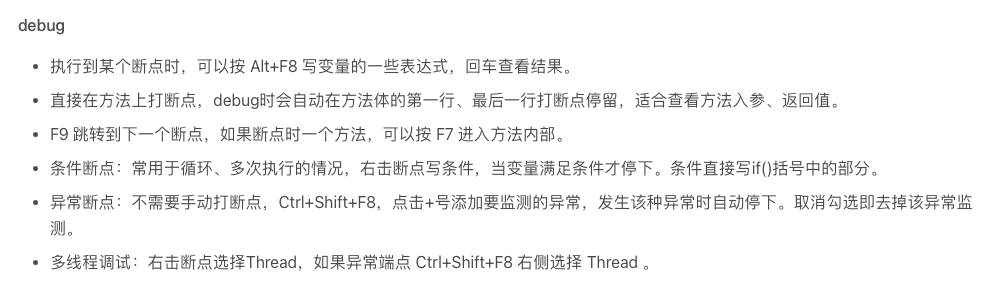
快捷指南：https://blog.csdn.net/chy\_18883701161/article/details/106342191?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522170955838816800226554009%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request\_id=170955838816800226554009&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~sobaiduend~default-1-106342191-null-null.142^v99^control&utm\_term=IDEA使用指南&spm=1018.2226.3001.4187



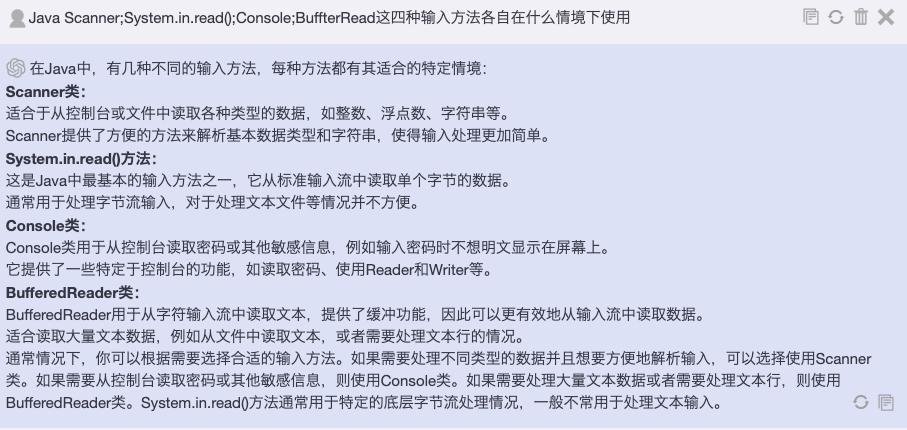








**三、实验总结与心得记录**

本次实验中，我了解到了除Scanner以外的其他输入方法，它们各自有各自适用的场景。gpt给出的特定情境如下：  


在控制台上运行java程序也是第一次，了解到了Java在控制台上运行的过程。虽然对于Console的学习浅尝辄止，但是它属于和密码等敏感信息有关的方法，相信之后肯定会或多或少进行了解。

其实在查找资料中是找到了第二种输出方法，BuffterWrite()方法。该方法自带缓冲区，查到的资料说明该方法一般用于写入文件而不是在控制台上显示，给出的示例代码也都是写入文件，所以此处没有过多研究。

另外，在运行第四个方法时，开始运行结果都为空。虽然知道这种情况一般是读到了缓冲区的换行符，但是一直以为是console方法。后面想明白console方法甚至都没有运行，而System.in.read()和BuffterRead()方法看起来就很像，应该都和什么缓冲区输入流有关，而且function2输入之后出来一堆换行，所以就确定是方法二的换行。

备注：

建议附带代码提交的方式：导出工程压缩包。

平时实验成绩以考查参与度为主，所有实验要求自己完成，一旦发现抄袭或者其他投机取巧，取消所有平时成绩