# 实验十： 输入输出流

## 实验目的

1、掌握java I/O的基本原理。

2、掌握标准输入输出流和Scanner类的基本使用方法。

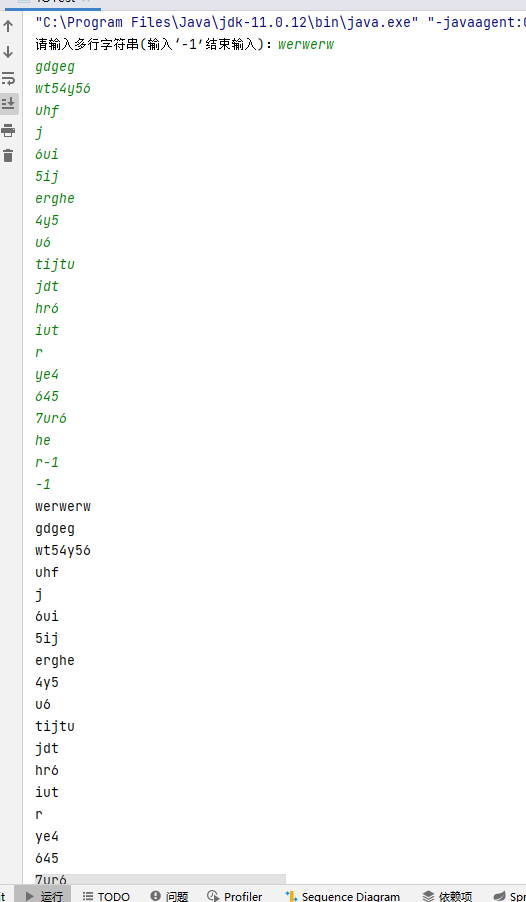
3、掌握FileInputStream、FileOutStream、FileReader、FileWriter、BufferedReader 、BufferedWriter类的常用方法。

## 实验内容

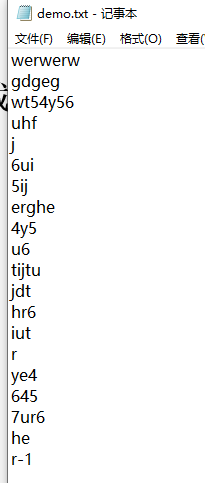
从键盘输入多行字符串（以单行“#”结束输入），存入指定文件，再从文件中读取内容输出到屏幕。

## 实验代码

package com.shf.demo11;  
  
import java.io.\*;  
import java.util.Scanner;  
  
public class IOTest *{* public static void main*(*String*[]* args*) {  
 writeFile(*"D:\\Develop\\upload\\demo.txt"*)*;  
  
 *readFile(*"D:\\Develop\\upload\\demo.txt"*)*;  
 *}* public static void readFile*(*String filePath*){  
 //创建一个读取文件的流对象* FileReader fileReader = null;  
 try *{* fileReader = new FileReader*(*filePath*)*;  
 *}* catch *(*FileNotFoundException e*) {* e.printStackTrace*()*;  
 *}  
 //bufferreader进行了一下封装，是读取的效率更高* BufferedReader bufferedReader = null;  
 if *(*fileReader != null*) {* bufferedReader = new BufferedReader*(*fileReader*)*;  
 *}* String str;  
 *//bufferedReader.readLine()是读取文件的一行，如果有多行，会逐行读取。  
 //当每一行不为空时则把内容打印到控制台中，也可以存到写入的流中，把内容写到文本里* while*(*true*) {* try *{* if *(*!*((*str = bufferedReader.readLine*())*!= null*)) {* break;  
 *}* System.*out*.println*(*str*)*;  
 *}* catch *(*IOException e*) {* e.printStackTrace*()*;  
 *}  
 }  
 //最后不要忘记关流* try *{* fileReader.close*()*;  
 *}* catch *(*IOException e*) {* e.printStackTrace*()*;  
 *}  
 }* public static void writeFile*(*String filePath*){* System.*out*.print*(*"请输入多行字符串(输入‘-1’结束输入)："*)*;  
 Scanner scanner = new Scanner*(*System.*in)*;  
 String str = "";  
 while*(*true*){* String s = scanner.nextLine*()*;  
 if *(*s.equals*(*"-1"*)){* break;  
 *}* str += s;  
 str += "\r\n";  
 *}* File f=new File*(*filePath*)*;  
  
 *//字节流输出* FileOutputStream fos=null;  
 try *{* fos=new FileOutputStream*(*f*)*;  
 fos.write*(*str.getBytes*())*;  
 *}* catch *(*Exception e*) {* e.printStackTrace*()*;  
 *}* finally *{* try *{* if *(*fos != null*) {* fos.close*()*;  
 *}  
 }* catch *(*IOException e*) {* e.printStackTrace*()*;  
 *}  
 }  
 }  
}*



## 实验截图



## 实验小结

通过本次实验我学会使用Java中的输入输出流对文件的读取和写入的操作，基本的步骤是①创建源或目标对象(文件)②创建IO流对象③具体的IO操作（读取写入）④关闭资源。并封装成为写入方法和读取方法，调用时传入文件地址即可读取写入。