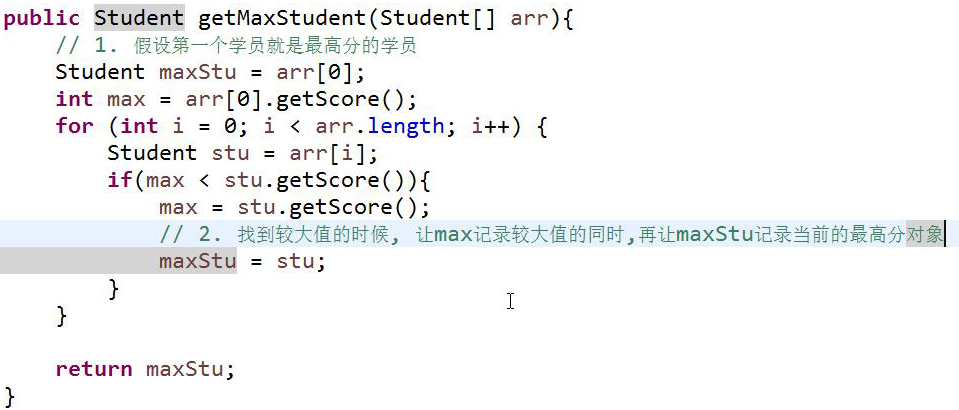
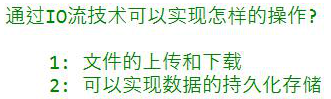
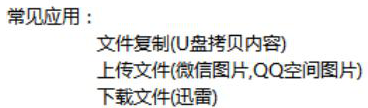
1 获取最高分对应的学生：



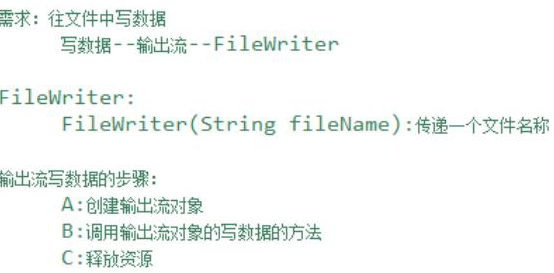
2 IO流的作用：

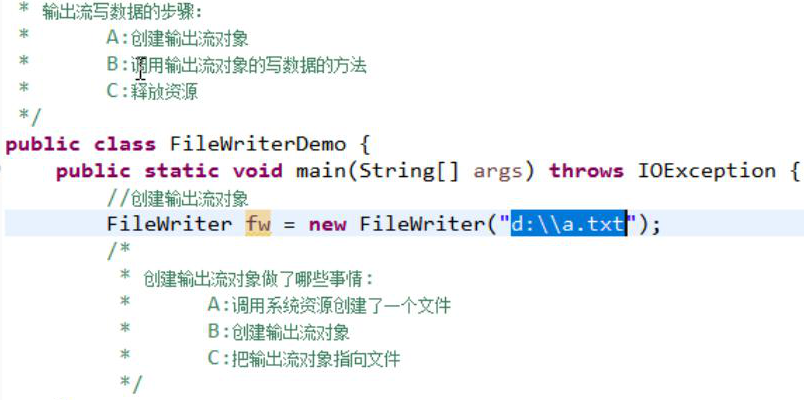


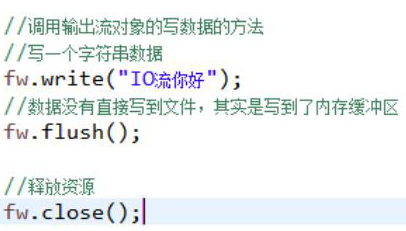
3 IO流的常见应用：



4 输出流





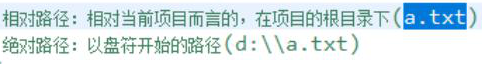


5 注意要写两个反斜线或一个正斜线：



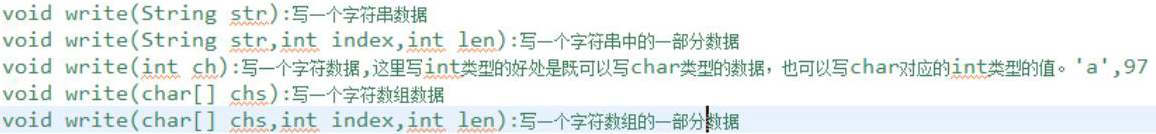
6 对着io流的红线ctrl+1可以**自动显示抛出异常选项**。（或点击左边红色错号也行）

7 文件不写绝对路径的话（即不写盘符）会存放在当前项目的根目录。



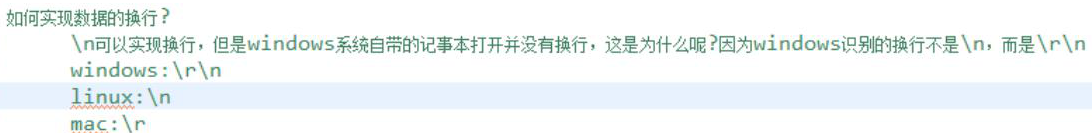
8 **流关闭之后不能再写数据**，但是刷新之后可以继续使用。

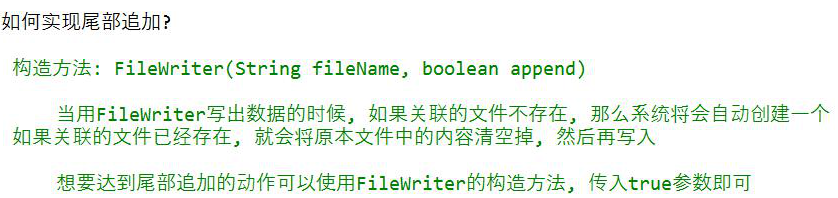
9 五种write()方法：（第一种和最后一种是重点）（**总结起来就是可以写入字符串（字符串的一部分）和字符数组（或字符数组的一部分）**）



注意第三个参数是长度而不是索引。（不管是字符串还是字符数组都是长度而不是索引）

10 F**ileWriter有清空再写入的效果。（假如文件已经存在的话。）如果想要尾部追加或换行：**





11



12 读取文件的两种方法：



当读取完毕再读取时，会读到-1.（**read读出的数据是字符所对应的数字而不是一个字符**）

13 换行和空格也是符号，因此也可以用FileReader读到。

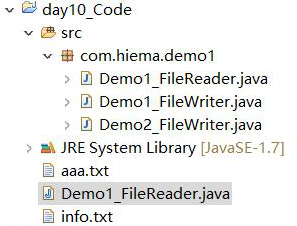
14 输入流貌似不用flush()。

15 被读取的文件不能不存在。

16 抛出的异常一般是IOException

17

项目的根目录是JRE system......：

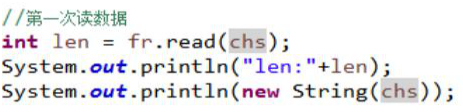


18 IO流核心代码：



**注意不用转换成字符，直接用整型就行。读出来的是整型，写进去的也是整型。**

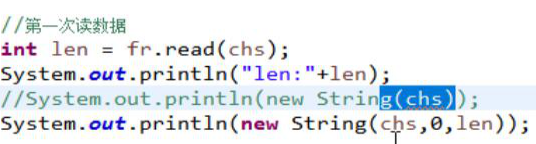
19 **read方法传入字符数组：（下面的构造方法可以将字符和字符数组转换成字符串）**



Fr.read()方法返回的是实际读取的字符个数而不是字符数组的长度，chs字符数组里面才是读取到的内容。打印时用new String()构造方法转换成字符串。如上面所示。



20 为了防止多读取：（文件已经结束还在读取）

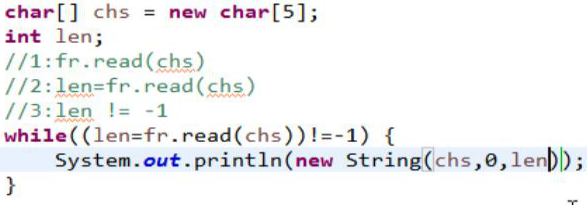


21 当实际返回长度为-1的时候代表未读取。

22 在当前项目下可以将文件拖到项目名字上。

23 **本地变量就是局部变量。**

24 应该除了最后一组之外全是len=5：



23 一般写1024，因为进制是1024：

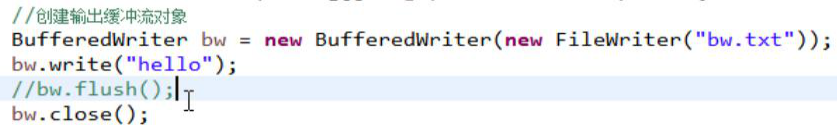


24 当文件的剩余字符数少于字符数组的长度时，由于字符数组是覆盖数据，所以剩下的位置是上一次的字符元素。例如abcde就返回ab,cd,ed，即出现了两个d。

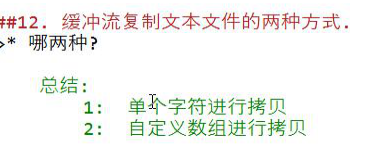
25 输入流和输出流都能缓冲：



26 缓冲流的输出：



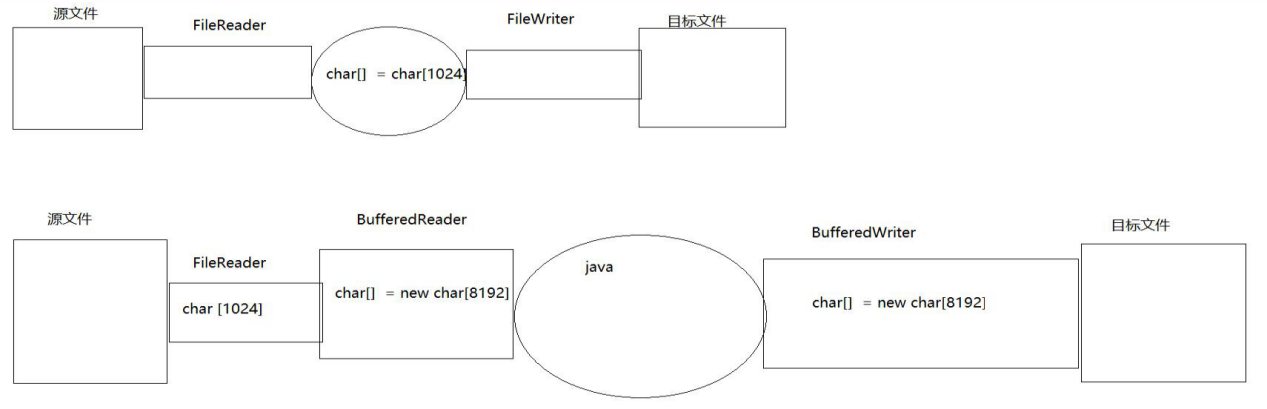
27



28 为什么复制前后不一样？

29 为什么用buffer高效流之后还有两种方法？

30 普通流和高效流图解：



31 buffer的特殊功能：（重点）



32 newLine()的跨平台性使得他可以在记事本中正确显示换行。

33 readLine()不读换行符，因此要用println()

34 用newLine()和readLine()拷贝文件：



注意当没有东西可读时，readLine()会返回null。

35 一共五种拷贝方式，标\*\*\*的效率最高（**总之就是两个有自定义数组的效率最高**）：



36 输入流和输出流都要记得close()。**用buffer之后只需要关闭buffer不需要关闭原来的普通流**。

37 如果一个方法抛出异常，调用它的方法也要抛出异常或者在方法内处理。

38 read()方法的返回值总结：**readLine()返回null或读取的内容**，read()返回-1或者字符的int类型数值（**不管是FileReader对象还是BufferedReader对象**）。当read()里面传入字符数组时，返回的是字符个数。**FileReader对象和BufferedReader对象都有read()方法。**

**缓冲流的四个对象：BufferedReader,BufferedWriter,BufferedInputStream,BufferedOutputStream。**

39 计算程序的运行时间：



40 输入流和输出流不一定同时存在。

41 不能只写路径，还要写文件名。

42  **Arrays.toString()打印出来的是集合那种形式。（数组的字符串化）**

43 集合中删除元素不仅可以根据索引删除，还可以直接传入要删除的字符串。

44 第九天作业第四题第二问的两种方法

45 加减乘除要加转义。（\\）**（减号不用转义，但转义了也行）**

46 在根目录添加新文件的方法是：项目名--右键==new file。

47 **集合有contains方法**。集合还有什么方法 **字符串也有contains方法**

48 buw快捷键

49 readLine()不会读取到换行符。

50 向集合中添加不重复的元素需要每次都遍历集合。当想向文件中添加不重复的信息而不是向集合中添加信息时可以将信息暂时的存放在集合中并用集合来检测信息是否存在，避免了使用输出流。集合还有一个好处是可以保存对象，因此在查询和比较时可以用对象的属性，而不用在文件中进行切割字符串的操作。

51 集合中不一定存储字符串，还可以存储对象：

//创建学生对象

Student s = new Student();

s.setId(id);

s.setName(name);

s.setAge(age);

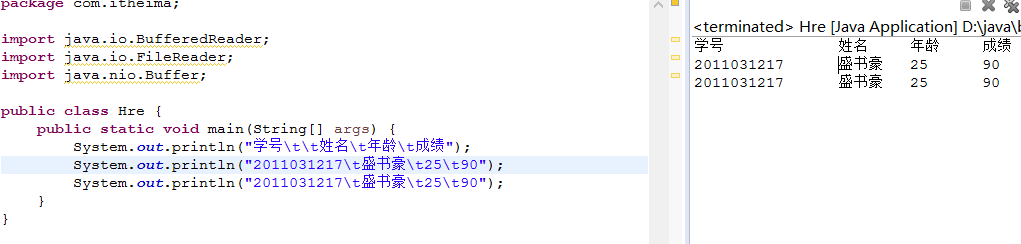
s.setAddress(address);

//把学生对象作为元素添加到集合

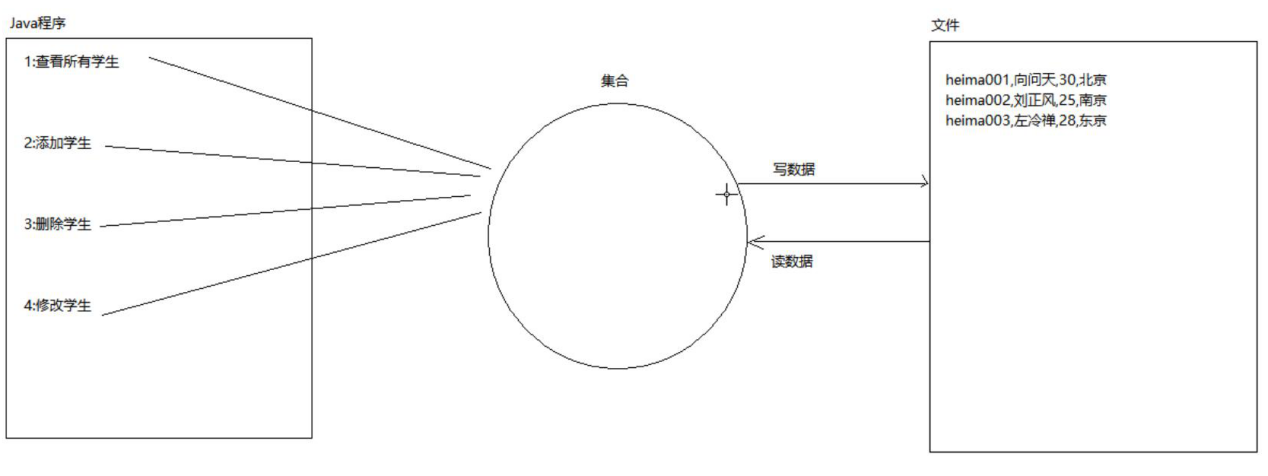
array.add(s);

52 StringBuilder可以提高拼接字符串的效率。

53 关于\t：每个字符串长度是8字节。超过八字节就会变成十六字节。



54 做学生管理系统IO版。（增删改查学生）（用集合做中间操作）

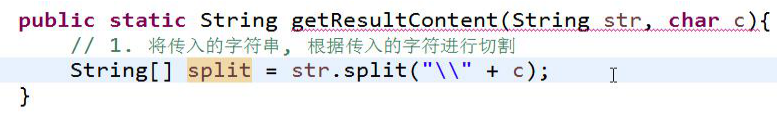


可以看出增删改查都要用到写数据和读数据，都是先将文件数据写到集合再将集合中的对象保存到文件。因此把这两个方法封装出来。其中查看学生只需要读文件到集合。

55 字符串，集合，io流。（考试重点）

56 文件读取到集合，集合修改之后再写回到文件。（考试重点）

57



58 switch的break和continue

59 **StringBuilder有翻转reverse()方法。**

60 StringBuilder必须先转换成String再写入到文件。

61 输入流和输出流操作的是同一个文件时要将文件保存在一个中间存储处，例如集合。

62 第十天最后一题不仅每行翻转，行的序号也要改变。

63 **是Character而不是Charactor。**

64 **去重复的原理是创建一个新的集合，保证该集合中不重复。很多问题可以先去重，例如求某数组中每个字符的出现次数。**

**65 获得元素出现次数的两种方法：**

**1.**

//用于获得传入元素在数组中出现的次数

public static int getcount(int[] arr ,int number){

int count=0;

for (int i = 0; i < arr.length; i++) {

if(number==arr[i]){

count++;

}

}return count;

}

//定义一个方法传入一个int类型数组，输出这个数组中只出现一次的数字及个数

public static void get4(int[] arr){

int number=0;

for (int i = 0; i < arr.length; i++) {

int c=getcount(arr,arr[i]);

if(c==1){//也只能检验1了，反复出现就会输出多次，出现几次就输出几次，虽然数值是对的但是反复输出。

System.out.println(arr[i]);

number++;

}

}System.out.println("只出现一次的次数"+number);

}

**2.**

**package** com.itheima;

**import** java.util.ArrayList;

/\*第二题：

定义一个字符串”zenmoyaok,a.o;;sddhia123”

1. 分别统计每个字符出现的次数？并打印在控制台上？\*/

**public** **class** Test2 {

**public** **static** **void** main(String[] args) {

String str="zenmoyaok,a.o;;sddhia123";

//首先去重

**char**[] cha=str.toCharArray();

ArrayList<Character> array=**new** ArrayList<>();//是er不是or>。用于保存不重复的字符。

**for** (**int** i = 0; i < cha.length; i++) {

**boolean** flag=**false**;

**for** (**int** j = 0; j < array.size(); j++) {//集合的长度为0时不会执行

**if**(array.get(j)==cha[i]){

flag=**true**;

**break**;

}

}

**if**(flag==**false**){

array.add(cha[i]);

}//太复杂了，直接用集合的contains方法。

}

**for** (**int** i = 0; i < array.size(); i++) {

**int** sum=0;

**for** (**int** j = 0; j < cha.length; j++) {

**if**(cha[j]==array.get(i)){

sum++;

}

}

System.***out***.println("字符"+array.get(i)+"的出现次数是"+sum);

}

}

}

66 double类型的数字初始化时要加.0

67 **remove(“字符串”)**只能去除第一个。如要去除全部需要while循环：（**注意contains()和remove（）方法**）

ArrayList<String> arr=**new** ArrayList<>();

arr.add("abc");

arr.add("abc");

arr.add("awe");

**while**(arr.contains("abc")){

arr.remove("abc");

}

System.***out***.println(arr);

}

**68 字符缓冲序列(StringBuilder):它里面可以添加任意类型数据,但是最终所有的数据都是以字符存储的。**

**69 delete(int start,int end)//将[start,end-1]之间的字符全部删掉。**

**70 总结所有截取的方法是数量还是长度：**

**substring()和delete()都不是长度，write()和new String(char[])是传入长度。后两者都和字符有关。**

**71**

**//定义一个方法传入一个int类型数组，输出这个数组中只出现一次的数字及个数**

**public static void geShu(int[] arr){//该方法的缺点是只判断是否出现，无法判断出现了几次，否则会重复输出。**

**int nuum=0;**

**System.out.print("只出现一次的数字是：");**

**for (int i = 0; i < arr.length; i++) {**

**boolean flag=true;**

**for (int j = 0; j < arr.length; j++) {**

**if(i!=j&&arr[i]==arr[j]){**

**flag=false;**

**break;**

**}**

**}**

**if(flag==true){**

**System.out.print(arr[i]+",");**

**nuum++;**

**}**

**}**

**System.out.println("只出现一次的数字个数是："+nuum);**

**}**

**72 直接传入的话必须要新建对象：**

/\*int[] aa=new int[]{1,2,3,4,1};

geShu(aa);\*/

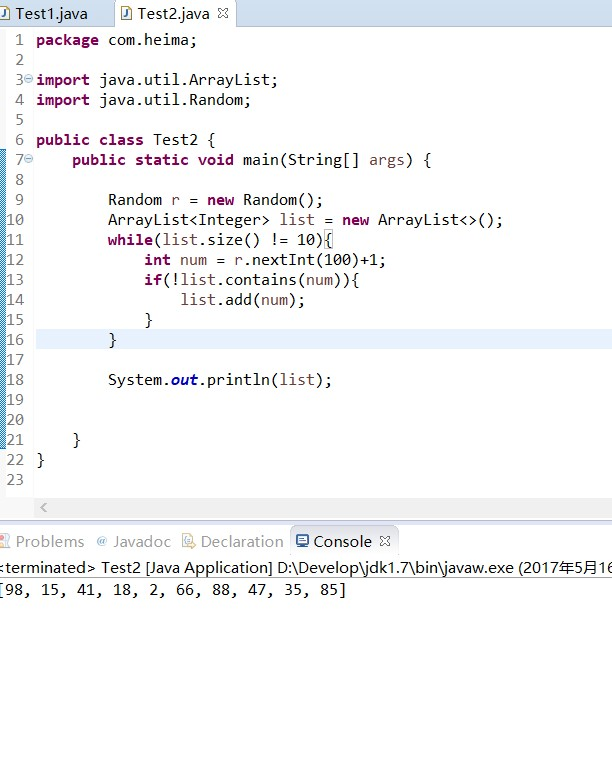
*geShu*(**new** **int**[]{1,2,3,4,1});

73 生成不重复的数组元素可以用到判断一个数组中是否存在某个数组元素的方法。

74 成员方法不加static，构造函数既没有static也没有返回值。

75 面向对象的方法中还可以传入本对象做实参。。。。

76



75 数组元素是否重复可以转化成出现了几次。

76 判断字符串中单词出现的次数要结合substring()和indexOf()

77

// 1. 定义长度为26，元素值为26个大写英文字母的数组chs

char[] chs = new char[26];

int index = 0;

for(char c = 'A'; c <= 'Z'; c++){

chs[index] = c;

index++;

}

78 string有contains（）方法，别的类没有。

79 只有当存在输入流且输入流用字符数组时，输出流才能用缓冲数组。