实验内容

1. 第一部分

编写一个Java源程序，其中包含编写一个包含主方法main的公共类（访问权限为public的类），主方法main中完成的任务是：

（1）从键盘上输入两个文件的路径名及其文件名。

（2）从键盘上输入若干学生的信息（姓名，数学成绩，计算机成绩，各项之间用空格隔开），以finish结束，将这些学生的信息写入第一个文件。

（3）将第一个文件作为输入流的源，找出姓“张”的学生，将他们的信息写入第二个文件。

将输入的每一行学生的信息作为一个字符串来处理。

1. 第二部分

编写一个Java源程序，其中的文件读写部分使用RandomAccessFile类来完成，程序实现的功能如下：

编写一个包含主方法main的公共类（访问权限为public的类），主方法main中完成的任务是：

（1）将第一部分产生的文件中的姓名“李善”改为“王山”。

（2）之后，在文件的最后继续添加新的学生信息。

（如果没有保存第一部分的文件，也可以在记事本中输入若干同学的姓名和成绩等信息作为第二部分的原文件。）

提示：

（1）用RandomAccessFile来对文件进行操作时，读写模式应该是“rw”。

（2）假设有RandomAccessFile类的对象raf，则

String s=raf.readLine();

表示从文件中读取一行，将读取的内容放入字符串对象s中。但是，如果读入的内容有汉字时，可能会读入乱码，导致读入的内容出错。通常采用如下语句

byte[] b=s.getBytes("ISO-8859-1");

String s1=new String(b);

进行转换，之后，用数组b创建的新字符串s1就没有乱码问题。

（3）在改名字的程序段中，应该存取两个指针的位置，一个是读完上一行之后的文件指针的位置，另一个是读完当前行之后的文件指针的位置（想一想，为什么？）。使用RandomAccessFile类的对象调用getFilePointer()和seek()方法实现上述任务。

（4）向文件中添加信息时，假设从键盘上输入的学生信息存放在字符串s中，则raf.write(s.getBytes());

可以向文件写入字符串。

实现方法

第一个程序从用户获取两个文件的路径，将学生信息写入第一个文件，然后按照姓氏过滤并将姓“张”的学生信息写入第二个文件。

第二个程序要求用户提供一个文件路径，首先替换文件中名为“李善”的学生姓名为“王山”，然后允许用户在文件末尾继续添加新的学生信息。

1. 第一部分：

主要通过基本的输入输出流完成文件的读写，通过BufferedReader从键盘上获取文件路径，并通过FileWriter将学生信息写入第一个文件。使用BufferedReader和FileReader搭配按行读取第一个文件，将姓“张”的学生信息用FileWriter写入第二个文件。

在读取第一个文件中的时候，先按行读取，再用String类的split方法获取学生姓名，而后将这个子串拿出来判断是否以“张”开头。

1. 第二部分：

使用BufferedReader获取文件路径，创建RandomAccessFile以实现对文件的随机读写。在替换学生姓名时，使用RandomAccessFile按行读取和修改文件内容。

getFilePointer()方法用于获取RandomAccessFile流中的当前指针。在代码中在每次遍历一行后pointer变量获取getFilePointer（）的返回值获取当前独读到的文件位置，方便下次匹配到需更改的文件内容时回溯到该行的前一个位置。

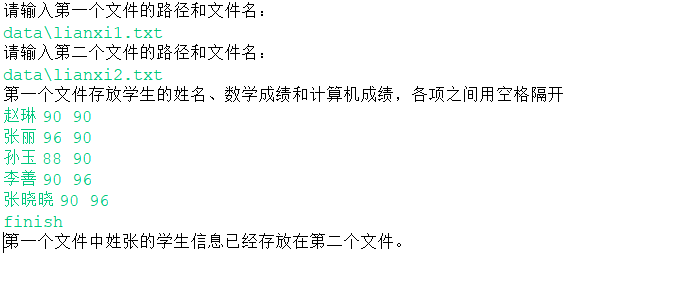
通过seek()方法定位文件指针，可以实现不按线性的移动，比如和上面的pointer配合回到上一行的起始位置，再比如直接移动到文件末尾，实现文件尾部的追加写入。

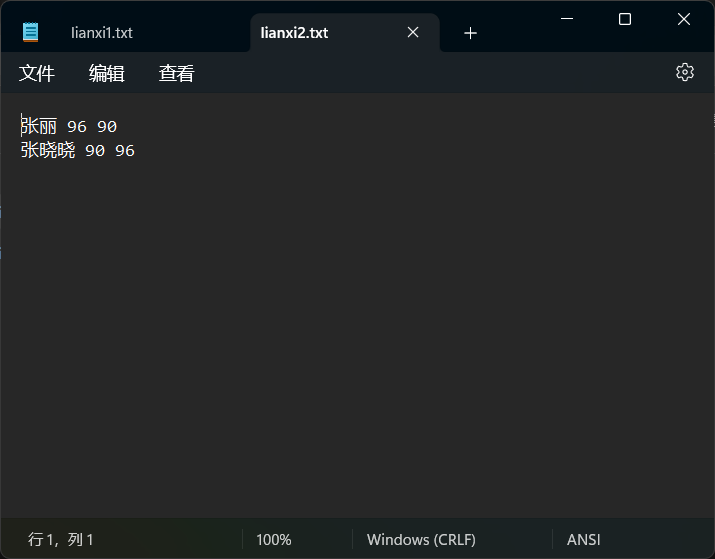
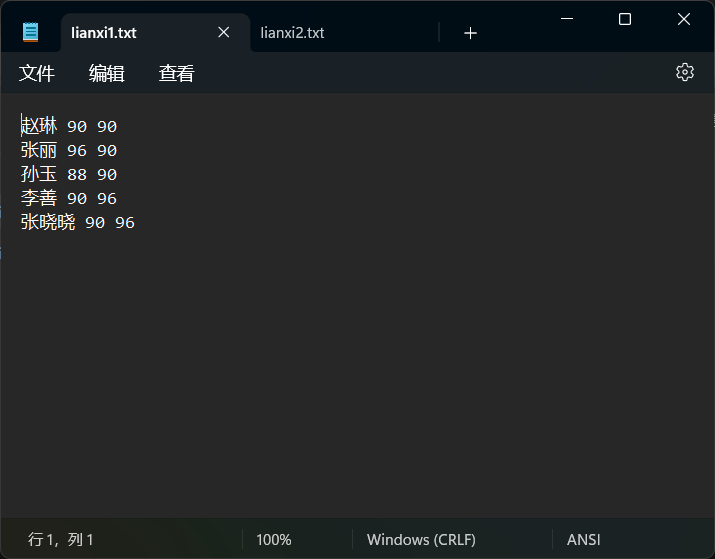
替换学生姓名的时候，将要替换的新名字字串和原来的字串的后半段子串拼接起来写入。

1. 注意：
2. 路径的正确：在eclipse中，当前路径在与src同级的文件夹下，事先要在该目录下创建data文件夹。
3. 使用异常处理保证文件操作的可靠性。
4. 在写文件的时候不需要按行写入，只需要用FileWriter写入并在行末加上换行符即可。
5. 在用字节流及其继承类的时候，要将String转换成byte数组，必要的时候需小心处理字符编码转换，确保不产生乱码。

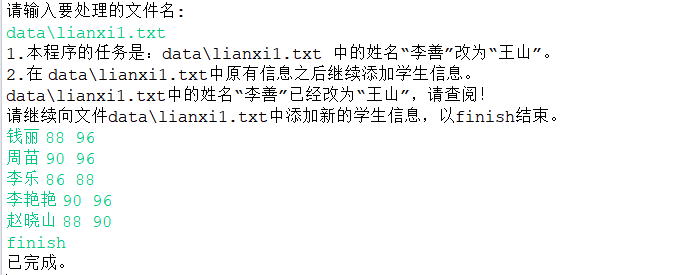
实验结果

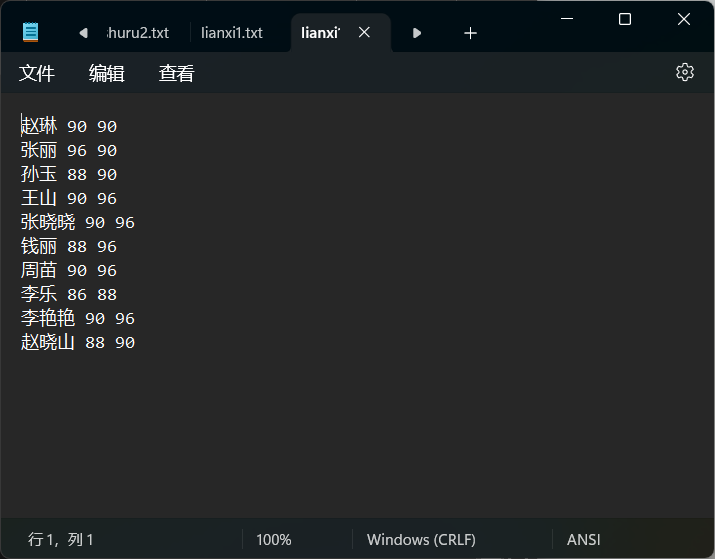
第一部分：





第二部分：





结论分析

通过不同的文件处理方式展现了Java中多样化的输入输出流和文件操作技术。

在第一部分，程序采用了缓冲字符流的方式，通过BufferedReader和FileWriter实现文件的逐行读写。这种方法适用于顺序读写的场景，不适用于随机访问文件的需求。

在第二部分，程序采用了随机访问文件的方式，通过RandomAccessFile实现对文件的随机读写。这种方法的灵活性和效率较高，适用于需要直接跳转到文件中特定位置进行读写的场景。通过seek()方法，程序可以定位文件指针并实现对文件内容的精确修改。