定义数组，向数组中存储4名学生，遍历数组判断学生性别，如果是男则存入man.txt。具体要求如下：  
 定义学生类Student,要求包括学生的姓名name(字符串类型)和学生性别sex（字符类型）两    种属性。  
1:在main方法中创建学生数组，并赋值，其中赋值内容如下  
张三，男；  
李四，女；  
王五，男；  
赵六，女；  
2:写一个方法   
adMan(Student[] stus) 将男生姓名添加到 man.txt中    
答案：

参考代码：  
 **class** Student {

//学生属性

**private** String name;

**private** String sex;

//构造方法

**public** Student() {

}

**public** Student(String name, String sex) {

**this**.name = name;

**this**.sex = sex;

}

//set get

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getSex() {

**return** sex;

}

**public** **void** setSex(String sex) {

**this**.sex = sex;

}

}

**public** **class** Test {

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

//创建学生对象数组 长度是4

Student[] stus = **new** Student[4];

//添加学生到数组中

stus[0]=**new** Student("张三", "男");

stus[1]=**new** Student("李四", "女");

stus[2]=**new** Student("王五", "男");

stus[3]=**new** Student("赵六", "女");

//将男生姓名添加到文件中

*addMan*(stus);

}

//将男生姓名添加到文件中

**public** **static** **void** addMan(Student[] stus) **throws** Exception {

//创建字符输出流

FileWriter fw = **new** FileWriter("man.txt");

//遍历数组

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

//得到每一个学生对象

Student stu = stus[i];

//判断学生是男是女

**if**(stu.getSex().equals("男")){

//学生名字写入 文件中

String name = stu.getName();

fw.write(name);

}

}

//关闭资源 释放流

fw.close();

}

}

已知存在一个金州勇士队的球员信息表，从表中提取球员名字  
player.txt文件内容：  
1,贾维尔-麦基,中锋,2.13米,122公斤,29岁,1988-01-19,8年  
2,布里安特-韦伯,后卫,1.88米,75公斤,25岁,1992-12-29,1年  
3,大卫-韦斯特,前锋,2.06米,113公斤,37岁,1980-08-29,13年  
5,凯文-卢尼,前锋,2.06米,100公斤,21岁,1996-02-06,1年  
9,安德烈-伊戈达拉,后卫/前锋,1.98米,98公斤,33岁,1984-01-28,12年  
11,克雷-汤普森,后卫,2.01米,98公斤,27岁,1990-02-08,5年  
20,詹姆斯-迈克尔-麦卡杜,前锋,2.06米,109公斤,24岁,1993-01-04,2年  
21,伊恩-克拉克,后卫,1.91米,79公斤,26岁,1991-03-07,3年  
23,德雷蒙德-格林,前锋,2.01米,104公斤,27岁,1990-03-04,4年  
27,扎扎-帕楚里亚,中锋,2.11米,125公斤,33岁,1984-02-10,13年  
30,斯蒂芬-库里,后卫,1.91米,86公斤,29岁,1988-03-14,7年  
34,沙恩-利文斯顿,后卫,2.01米,87公斤,32岁,1985-09-11,11年  
35,凯文-杜兰特,前锋,2.06米,109公斤,29岁,1988-09-29,9年

注意： 第  第二列是球员姓名  
 根据文件中的信息完成下列操作：  
  1:在main方法中创建存储球员信息集合  
2:写一个方法，读取文件并将球员信息添加至集合中  
3:写一个方法：遍历球员姓名  
获取  
代码：

 **public** **class** Test {

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

//创建存储球员的集合

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

// 添加球员的方法

*add*(list);

System.*out*.println("金州勇士队球员名字如下：");

// 展示球员信息

*print*(list);

}

**public** **static** **void** add(ArrayList<String> list) **throws** Exception {

// 创建高效字符流并关联文件

BufferedReader br = **new** BufferedReader(**new** FileReader("player.txt"));

// 定义字符串接收变量

String line;

// 读取文件

**while**((line=br.readLine())!=**null**){

// 解析并进行切割 根据观察按照 英文逗号 进行切割

String[] split = line.split(",");

// 球员信息是第二列 并存储到集合中

list.add(split[1]);

}

br.close();

}

**public** **static** **void** print(ArrayList<String> list) {

// 遍历集合

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

// 输出信息

System.*out*.println(list.get(i));

}

}

}

/\*

\* 1.项目根目录下已存在文件player.txt，文件中存放球员和球队

\* 存储格式如下(原始球员和球队不用程序录入，可手动建文件并在文件中录入数据)：

\* 球员,球队（中间用,分隔）,如：

\* 杜兰特，勇士队

\* 威少,雷霆队

\* 库里,勇士队

\* 石破天,湖人队

\* 科比,湖人队

\* 2.程序启动时：提示用户输入球员名 然后检索出球员多对应的球队

\* 2.1 键盘录入球员名

\* 2.2 在player.txt中获取球员名和所属球队，验证键盘录入的球员是否存在

\* 是：控制台将球员的所属球队信息打印

\* 否：控制台提示：不存在该球员，并结束程序

\*/

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;

import java.util.Scanner;

public class Test3 {

public static void main(String[] args) throws IOException {

//创建Scanner对象

Scanner sc = new Scanner(System.in);

System.out.println("请录入您要查找的球员名字:");

String s = sc.nextLine();//s---录入要查找的球员名字

//需要去文件中找 读取文件信息

FileReader fr = new FileReader("player.txt");

BufferedReader br = new BufferedReader(fr);

//获取每一行的信息

String line;

while((line=br.readLine())!=null){

//line 杜兰特，勇士队

String[] player = line.split(",");

String name = player[0];

String team = player[1];

if(s.equals(name)){//判断键盘录入的名字是否存在

//存在 就进来了

System.out.println(name+",来自"+team);

//如果查到了 就不用再查询了 可以结束这个方法

br.close();

fr.close();

return;//执行到这里 方法结束

//在这里 只要找到了 就没有再继续查找的必要

}

}

//如果 方法没有结束 说明没有该球员

System.out.println("没有该球员");

br.close();

fr.close();

}

}