import java.io.File;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.RandomAccessFile;

import java.util.Arrays;

import java.util.Scanner;

import java.util.Vector;

interface Addition {

public void save();

public void load();

}

public class Worker implements Comparable{ //class rabotnik, s implementirasht Comparable

private String Name; //skriti chlen promenlivi, => ime,pol, obrazovanie

private String Sex;

private int Education;

//metodi

public Worker() {

this.setName("N/A");

this.setSex("N/A");

this.setEducation(0);

}

public Worker(String Name1, String Sex1, int Edu1) {

this.setName(Name1);

this.setSex(Sex1);

this.setEducation(Edu1);

}

public int getEducation() {

return Education;

}

public void setEducation(int education) {

Education = education;

}

public String getSex() {

return Sex;

}

public void setSex(String sex) {

Sex = sex;

}

public String getName() {

return Name;

}

public void setName(String name) {

Name = name;

}

public String toString() {

return "Name: " +Name+ "Sex: " +Sex+ "Education: " +Education+ "\n";

}

public boolean ravno(Object ed1)

{

Worker ed2 = (Worker) ed1;

return Education == ed2.Education;

}

public boolean pomalko(Object ed4)

{

Worker ed3 = (Worker) ed4;

return Education < ed3.Education;

}

@Override

public int compareTo(Object o) {

// TODO Auto-generated method stub

return 0;

}

}

class Firma implements Addition{

protected Vector<Firma> kolekciq= new Vector<Firma>();

protected String vhod;

protected String izhod;

public void save()

{

try {

RandomAccessFile fout = new RandomAccessFile(izhod, "rw");

for (Worker current : kolekciq) {

fout.writeBytes(current.getName()+" ");

fout.writeBytes(current.getSex()+" ");

fout.writeBytes(current.getEducation()+"\r\n");

}

fout.close();

} catch (FileNotFoundException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

public Firma (String file1, String file2) {

vhod = file1;

izhod = file2;

}

public void load() {

try {

Scanner iStream = new Scanner(new File(vhod));

while(iStream.hasNext()) {

Worker red = new Worker(iStream.next(),iStream.next(),iStream.nextInt());

kolekciq.add(red);

}

iStream.close();

} catch (FileNotFoundException e) {

e.printStackTrace();

}

}

public void toString1() {

System.out.println(kolekciq.toString());

}

}