Program.java

import java.lang.\*;

import java.util.Scanner;

import java.io.File;

import java.io.FileNotFoundException;

public class Program

{

/\*\*

\* This is the main entry point for the application

\*/

public static void main(String args[]) {

Person []residents = new Person[100];

AuthorizedPerson []authorizedPerson = new AuthorizedPerson[100];

int totalResidents = 0;

int totalAuthorized = 0;

try {

Scanner s = new Scanner(new File("inp.dat"));

while(s.hasNextLine()) {

residents[totalResidents] = new Person();

residents[totalResidents].setName(s.nextLine());

boolean isAuthority = false;

if (s.nextLine().endsWith("yes"))

isAuthority = true;

residents[totalResidents].setSex(s.nextLine());

residents[totalResidents].setAge((int)Integer.parseInt(s.nextLine()));

residents[totalResidents].setAddress(s.nextLine());

if (isAuthority) {

authorizedPerson[totalAuthorized] = new AuthorizedPerson();

authorizedPerson[totalAuthorized].setName(residents[totalResidents].getName());

authorizedPerson[totalAuthorized].setAge(residents[totalResidents].getAge());

authorizedPerson[totalAuthorized].setSex(residents[totalResidents].getSex());

authorizedPerson[totalAuthorized].setAddress(residents[totalResidents].getAddress());

totalAuthorized++;

}

totalResidents++;

}

} catch (FileNotFoundException e) {

e.printStackTrace();

}

// Done with data inputting.

System.out.println(totalResidents+", "+totalAuthorized);

// Query 1 - Residents elder than the requester - We query over all authorized females since

// the answer is dependent on who the asker is.

for (int i = 0; i < totalAuthorized; i++) {

if (authorizedPerson[i].getSex().equals("female")) {

int q1 = authorizedPerson[i].residentsElderThanMe(residents, totalResidents);

System.out.println("Query 1: "+q1+", asked by - "+authorizedPerson[i].getName());

}

}

// Query 2 - oldest male age. We use break here since the query output doesn't change based on

// who is asking it.

for (int i = 0; i < totalAuthorized; i++) {

if (authorizedPerson[i].getSex().equals("male")) {

int q2 = authorizedPerson[i].oldestMaleAge(residents, totalResidents);

System.out.println("Query 2: "+q2+", asked by - "+authorizedPerson[i].getName());

break;

}

}

// Query 3 - Find name provided an address

for (int i = 0; i < totalAuthorized; i++) {

if (authorizedPerson[i].getAge() < 20) {

String q3 = authorizedPerson[i].getNameGivenAddress(residents, totalResidents, "B-14 James, Avenue ");

System.out.println("Query 3: "+q3+", asked by - "+authorizedPerson[i].getName());

break;

}

}

}

}

Person.java

import java.lang.Math;

public class Person {

private String name;

private String sex;

private int age;

private String address;

public Person() {

}

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 int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String toString() {

return "Name: "+this.getName()+"\nSex: "+this.getSex()+"\nAge: "+this.getAge()+"\nAddress: "+this.getAddress();

}

}

AuthorisedPerson.java

import java.lang.Math;

public class AuthorizedPerson {

private String name;

private String sex;

private int age;

private String address;

public AuthorizedPerson() {

}

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 int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

/\*\*

\* Return the number of residents elder than me.

\* This should only happen

\*/

public int residentsElderThanMe(Person[] residents, int totalResidents) {

int numResidents = 0;

for (int i = 0; i < totalResidents; i++) {

if (residents[i].getAge() > this.age)

numResidents++;

}

return numResidents;

}

public int oldestMaleAge(Person[] residents, int totalResidents) {

int oldestAge = this.age;

for (int i = 0; i < totalResidents; i++) {

if (residents[i].getSex().equals("male")) {

oldestAge = Math.max(oldestAge, residents[i].getAge());

}

}

return oldestAge;

}

public String getNameGivenAddress(Person[] residents, int totalResidents, String address) {

for (int i = 0; i < totalResidents; i++) {

if (residents[i].getAddress().equals(address))

return residents[i].getName();

}

return null;

}

public String toString() {

return "Name: "+this.getName()+"\nSex: "+this.getSex()+"\nAge: "+this.getAge()+"\nAddress: "+this.getAddress();

}

}