



نظام إدارة مستشفى (أتمتة بيانات)

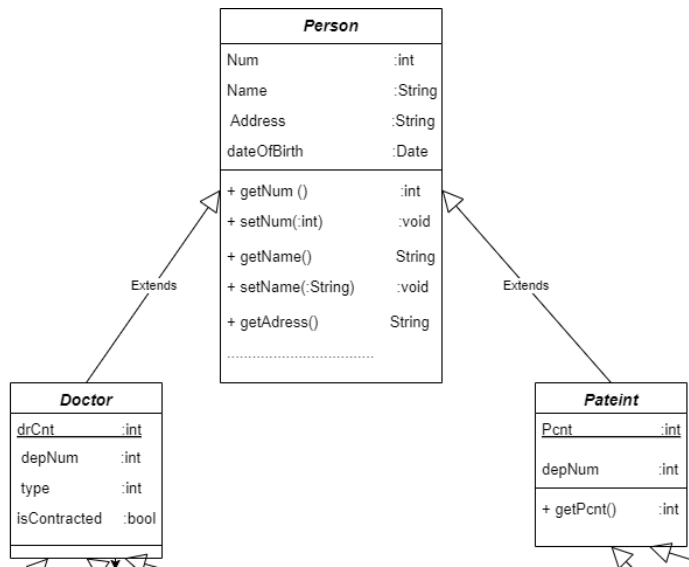
المشروع الفصلي (برمجة 2)

بنى المعطيات المستخدمة:

(1) الصفوف:

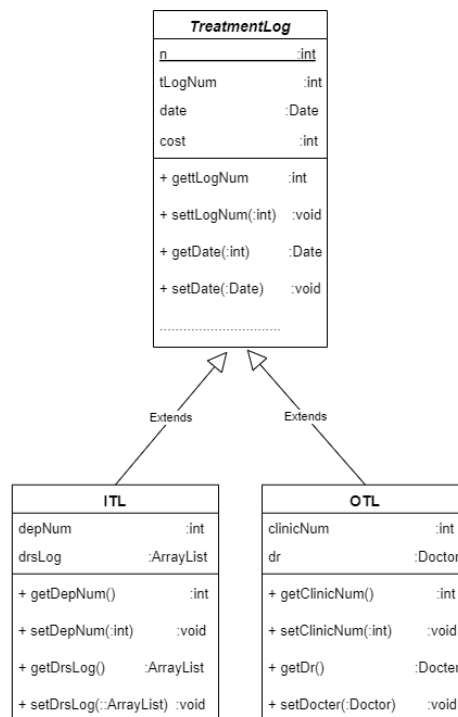
Date	
d	:int
m	:int
y	:int
diff(Date, Date) :int	
setDate(int, int, int) :Date	
isOk(Date) :boolean	

في البداية أنشأنا صف "تاريخ" لتنسيق التواريخ والعمليات عليها في البرنامج.



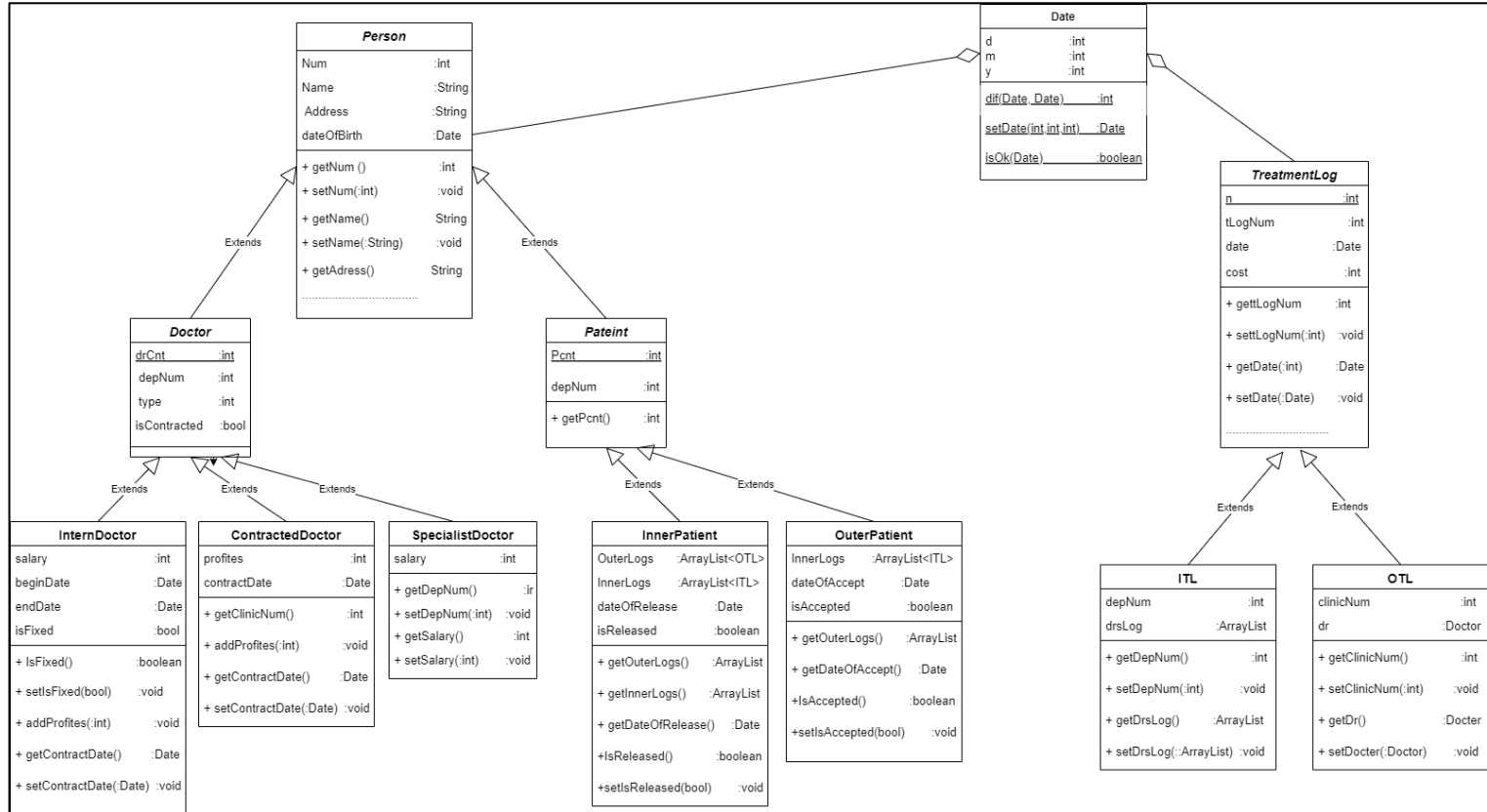
ثم قمنا بإنشاء صف باسم "شخص" وقمنا بتوريثه لصفين، طبيب ومريض (وهي صفوف مجردة) ثم قمنا بإنشاء صفوف ترث الطبيب (متعاقد – متدرب – اختصاصي) ومن المريض (داخلي – خارجي) وهي الصفوف التي سنخلق منها كائنات.

بعدها أنشأنا صف سجل العلاج (مجرد) وورثنا منه صفان (سجل داخلي – سجل خارجي).



لا ننسى أننا قمنا بكتابة جميع البواني وال (Getters & setters) في الصفوف السابقة.

مخطط الصفوف:



(2) المصفوفات:

ضمن صف سجل العلاج الداخلي قمنا بإنشاء مصفوفة لتخزين سجلات الأطباء المشرفين على المعالجة.

في صف المريض الخارجي أنشأنا مصفوفة لتخزين سجلات العلاج الخارجية، وفي صف المريض الداخلي أنشأنا مصفوفتان لتخزين سجلات العلاج الخارجية والداخلية.

وفي الكلاس الأساسي "إدارة المستشفى" أنشأنا مصفوفة لتخزين سجلات الأطباء وأخرى لتخزين سجلات المرضى.

```

10 // Doctors & Patients Lists-----
11 static ArrayList<Doctor> drs = new ArrayList(); //---
12 static ArrayList<Patient> patients = new ArrayList(); //---
13 //-----

```

البرامج الجزئية:

في التابع الرئيسي وضعنا

Flag while loop

يكون الدخل متحول محرف

ويغلق البرنامج إذا كان الدخل q=

يتم استدعاء التابع func الذي

يستقبل المحرف المدخل.

```
1 package hospitalmanaging;
2 import java.util.ArrayList;
3 import static java.util.Arrays.asList;
4 import java.util.Scanner;
5
6 public class HospitalManaging{
7
8     static Date today = new Date(7, 6, 2024);
9
10    // Doctors & Patients Lists-----
11    static ArrayList<Doctor> drs = new ArrayList(); //---
12    static ArrayList<Pateint> patients = new ArrayList(); //---
13    //-----
14
15    public static Scanner in = new Scanner(System.in);
16    public static void main(String[] args) {
17
18        System.out.println("-----Welcome To Hospital Managment System-----");
19        while(true){
20            System.out.println("exit: 'q' || add: 'a' || delete: 'd' ");
21            System.out.println("fix: 'f' || count: 'c' || info: 'i'");
22            System.out.println("release patient: 'r'");
23            char input = in.next().charAt(0);
24            if(input == 'q'){
25                System.out.println("Closing System");
26                return;
27            }
28            func(input);
29        }
30    }
```

```
32 public static void func(char i){
33     switch (i) {
34         case 'a':{
35             System.out.println("What do you want to add?");
36             System.out.println("Doctor log: 'd' | Patient Log: 'p' | Treatment: 't'");
37             add();
38             break;}
39
40         case 'd':{
41             System.out.println("Delete doctor log");
42             delete();
43             break;}
44
45         case 'f':{fix();break;}
46
47         case 'c':{count();break;}
48
49         case 'i':{info();break;}
50
51         case 'r':{release();break;}
52
53         default:System.out.println("WRONG INPUT");break;
54     }
55 }
```

يقوم التابع func باستدعاء

التوابع الجزئية (إضافة/حذف/

تنبيت/إحصاء/عرض معلومات/

تخريج).

نقوم في تابع الإضافة بإدخال محرف لتحديد فيما إذا كنا نريد إضافة طبيب أو مريض أو سجل معالجة لمريض.

في حال إضافة طبيب ندخل المتحولات الخاصة به واحدة تلو الأخرى، وفي النهاية دخل نوع الطبيب (محرف) لخلق الكائن وإضافته لسجل الأطباء

```
72 System.out.println("Enter if contracted 'c', fixed 'f' or intern 'i':");
73 char ch = in.next().charAt(0);
74 switch(ch){
75     case 'c':
76         System.out.println("(Clinics) Gastrointestinal'5' | Ophthal '6' | Internal '7' ...");dn =
77         ContractedDoctor doc1 = new ContractedDoctor(dn, nm, ad, bd, Begin);
78         drs.add(doc1);
79         System.out.println("Doctor was added succesfully!\nhis serial number is: "+doc1.Num);
80         break;
81     case 'f':
82         System.out.println("(Departments) Ophthal'1' | Surgery '2' | Thoracic '3'| Ambulance'4' ")
83         System.out.println("Enter Salary:");int s1 = in.nextInt();
84         SpecialistDoctor doc2 = new SpecialistDoctor(s1, dn, nm, ad, bd);
85         drs.add(doc2);
86         System.out.println("Doctor was added succesfully!\nhis serial number is: "+doc2.Num);
87         break;
88     case 'i':
89         System.out.println("(Departments) Ophthal'1' | Surgery '2' | Thoracic '3'| Ambulance'4' ")
90         System.out.println("Enter Salary:");int s2 = in.nextInt();
```

وفي حال إضافة مريض فإننا ندخل المتحولات، وفي النهاية نحدد إذا كان المريض خارجي أم داخلي.

```
System.out.println("Enter if inner'i' or outer pateint'o:");
char type = in.next().charAt(0);
if(type=='i'){
    System.out.println("(Departments) Ophthal'1' | Surgery '2' | Thoracic '3'| Ambulance'4'");
    InnerPatient p1 = new InnerPatient(nm, ad, bd);patients.add(p1);
    p1.setDepNum(dep);
}
else if (type=='o'){
    System.out.println("(Clinics) Gastrointestinal'5' | Ophthal '6' | Internal '7' :...");
    OuterPatient p2 = new OuterPatient(nm, ad ,bd);patients.add(p2);
    p2.setDepNum(dep);
}
```

وفي حال إضافة سجل علاج ندخل المتحولات من تاريخ وكلفة، ثم نختار المريض ونختار القسم او العيادة الذي تمت بها المعالجة ونضيف الأطباء المشرفين إلى السجل (باستخدام الرقم التسلسلي)

```
case 't':
    int cost;
    System.out.println("Enter the date of the treatment(day/month/year):");d = in.nextInt();m = in.nextInt();
    Date tDate = new Date(d,m,y);
    if(Date.isOk(tDate)==false){System.out.println("Bad input!!!"); return;}
    System.out.println("Enter the cost:");cost = in.nextInt();
    System.out.println("give me the serial number of the patient:");
    int n = in.nextInt();
    for(int i=0; i<patients.size(); i++){
        if(patients.get(i).Num == n){
            if(patients.get(i).depNum<=4){
                System.out.println("Enter the department number:");
                System.out.println("(Departments) Ophthal'1' | Surgery '2' | Thoracic '3'| Ambulance'4' ");
                System.out.println("(Clinics) Gastrointestinal'5' | Ophthal '6' | Internal '7' :...");
                dn = in.nextInt();
                if(dn>=1 && dn <=4){
                    ArrayList<Doctor> doctors =new ArrayList();
                    System.out.println("Enter the serial numbers of engaged doctors(type '0' when stop):");
                    for(int j=0; j<drs.size(); j++){
                        int drNum = in.nextInt();
                        if(drNum==0) break;
                        else if(drs.get(j).Num == drNum){
                            doctors.add(drs.get(j));
                            System.out.println("Doctor added sucessfully!");
                        }
                    }
                }
                ITL log = new ITL(dn, doctors, tDate, cost);
                ((InnerPatient) patients.get(i)).InnerLogs.add(log);
            }
        }
    }
}
```

```
ITL log = new ITL(dn, doctors, tDate, cost);
((InnerPatient) patients.get(i)).InnerLogs.add(log);
}
else if(dn>=5 && dn <=7){
    Doctor x = null;
    System.out.println("Enter the serial number of engaged doctor:");
    int drNum = in.nextInt();
    for(int j=0; j<drs.size(); j++){
        if(drs.get(j).Num == n)
            x = drs.get(j);
    }
    if(x.type == 2){
        OTL log = new OTL(dn, x, tDate, cost);
        ((InnerPatient) patients.get(i)).OuterLogs.add(log);
        ((ContractedDoctor) x).addProfites(cost/2);
    }
    else{
        System.out.println("Not a contracted doctor!!!");return;
    }
}
}
else if(patients.get(i).depNum<=7){
    System.out.println("Enter the department number:");
    System.out.println("(Clinics) Gastrointestinal'5' | Ophthal '6' | Internal '7' :...");dn = in.nextInt();
    Doctor x = null;
    System.out.println("Enter the serial number of engaged doctor:");
    int drNum = in.nextInt();
    for(int j=0; j<drs.size(); j++){
        if(drs.get(j).Num == drNum)
            x = drs.get(j);
    }
}
```

ملاحظة: قمنا باستخدام ال Down Casting في عدة أماكن في توابع الإضافة لتخزين البيانات بطريقة صحيحة دون مواجهة مشاكل Errors.

نقوم في تابع الحذف بإدخال الرقم التسلسلي للطبيب الذي نريد حذف سجله،
أنشأنا حلقة لإيجاد الطبيب من مصفوفة الأطباء، وإذا لم يكن موجوداً نطبع

```
201 public static void delete(){
202     System.out.println("give me the serial number of the doctor you want to delete");
203     int n = in.nextInt();
204     for(int i=0; i<drs.size(); i++){
205         if(drs.get(i).Num == n){
206             System.out.println("Doctor was removed sucessfully");
207             drs.remove(drs.get(i));
208             return;
209         }
210     }
211     System.out.println("Out of range!!!");
```

Out Of Range

نقوم في تابع التثبيت بإدخال الرقم التسلسلي للطبيب الذي نريد تثبيته، أي يجب
أن يكون متدرب، وعند تثبيته سيتقاضى 100% من الراتب

```
public static void fix(){
    System.out.println("give me the serial number of the doctor you want to fix");
    int n = in.nextInt();
    if(n>drs.size()){
        System.out.println("Out of range!!!");
    }
    else{
        for(int i=0; i<drs.size(); i++){
            if(drs.get(i).Num == n){
                if(drs.get(i).type==3){
                    if(((InternDoctor) drs.get(i)).IsFixed()){
                        System.out.println("Doctor is Fixed already!");
                    }
                    else{
                        ((InternDoctor) drs.get(i)).setIsFixed(true);
                        System.out.println("Doctor was fixed succesfully");
                    }
                }
                else{
                    System.out.println("He is not an intern!!!");
                }
            }
        }
    }
}
```

يوجد تابع

IsFixed()

عندما تصبح القيمة داخله

True

يقوم بتغيير الراتب

نقوم في تابع التخرج بإدخال الرقم التسلسلي للمريض الذي نريد تخرجه،
ويجب أن يكون المريض داخلي.

```
public static void release(){
    System.out.println("give me the serial number of the pateint you want to release(Inner Patient): ");
    int pnun= in.nextInt();
    for(int i=0; i<patients.size(); i++){
        if(patients.get(i).Num == pnun){
            ((InnerPatient) patients.get(i)).isReleased = true;
            System.out.println("Patient was released succesfully");
            return;
        }
    }
    System.out.println("Out of range!!!");
}
```

نقوم في تابع الإحصاء بإدخال نوع البيانات التي نريد إحصاءها (محرف).

```
236 public static void count() {
237     System.out.println("What do you want to count?");
238     System.out.println("Contracted Doctors: 'd' | All Doctors: 'D'");
239     System.out.println("Pateints in a clinic or department: 'p' | All Pateints : 'P'");
240     char input = in.next().charAt(0);
241     switch (input) {
242         case 'd':
243             int contractedCnt=0;
244             for(int i=0; i<drs.size(); i++){
245                 if(drs.get(i).isContracted)
246                     contractedCnt++;
247             }
248             System.out.println("The number of contracted doctors is: "+contractedCnt);
249             break;
250
251         case 'D':
252             System.out.println("The number of doctors is: ");
253             System.out.println(drs.size());
254             break;
255
256         case 'p':
257             int dep;int cnt = 0;
258             System.out.println("Enter department number(1 -> 7)");
259             System.out.println("(Departments) Ophthal'1' | Surgery '2' | Thoracic '3'| Ambulance'4' ");
260             System.out.println("(Clinics) Gastrointestinal'5' | Ophthal '6' | Internal '7' :...");
261             dep = in.nextInt();
262             for(int i=0; i<patients.size(); i++){
263                 if(patients.get(i).depNum == dep) cnt++;
264             }
265             System.out.println("The number of patients is : "+cnt);
266             break;
267     }
```

نقوم في تابع إظهار المعلومات بإدخال نوع البيانات التي نريد أن نعرف معلومات عنها (محرف).

```
275 public static void info() {
276     System.out.println("What do you want to see details about?");
277     System.out.println("All Doctors(duration): 'D' | All Tretments (pateint):'T'");
278     System.out.println("All Patients(duration): 'P'");
279     char input = in.next().charAt(0);
280     System.out.println("=====");
281     switch (input) {
282         case 'T':
283             System.out.println("Enter the serial number of the patient:");int n=in.nextInt();
284             for(int i=0; i<patients.size(); i++){
285                 if(patients.get(i).Num == n){
286                     if(patients.get(i).depNum <= 4){
287                         ArrayList<ITL> iLogs = ((InnerPatient) patients.get(i)).getInnerLogs();
288                         ArrayList<OTL> oLogs = ((InnerPatient) patients.get(i)).getOuterLogs();
289                         for(ITL x: iLogs){System.out.println(x.toString());}
290                         for(OTL x: oLogs){System.out.println(x.toString());}
291                         break;
292                     }
293                     else if(patients.get(i).depNum <= 7){
294                         ArrayList<OTL> oLogs = ((OuterPatient) patients.get(i)).getOuterLogs();
295                         for(OTL x: oLogs){System.out.println(x.toString());}
296                         break;
297                     }
298                 }
299             }
300             System.out.println("=====");
301             break;
302         case 'D':
303             for(int i=0; i<drs.size(); i++){
304                 System.out.println(drs.get(i).toString());
305                 System.out.println("=====");
306             }
307             break;
308         case 'P':
309             for(int j=0;j<patients.size(); j++){
310                 if(patients.get(j).depNum <= 4){
311                     System.out.println(((InnerPatient) patients.get(j)).toString());
312                 }
313                 else if(patients.get(j).depNum <= 7){
314                     System.out.println(((OuterPatient) patients.get(j)).toString());
315                 }
316                 System.out.println("=====");
317             }
318             break;
319         default:
320             System.out.println("WRONG INPUT!!!");break;
321     }
322 }
```

ملاحظة: قمنا باستعمال

تابع toString لإظهار

البيانات الخاصة بكل نوع،

إضافة لأننا قمنا بعملية

Override

للتابع في الأنواع التي

تستخدم الوراثة.

قمنا أيضاً بتعبئة المصفوفات بعدد من الأطباء والمرضى:

```
338 //FILLING THE ARRAYLISTS-----
339
340 static{
341     InternDoctor dr1 = new InternDoctor(3000, 3, "Ahmad Kerdi", "Syria/Damascus", Date.setDate(5, 4, 2000), Date.setDate(15, 3, 2020));
342     InternDoctor dr2 = new InternDoctor(3500, 2, "Malek Arab", "Syria/Daraa", Date.setDate(9, 11, 1999), Date.setDate(6, 3, 2022), 1);
343     InternDoctor dr3 = new InternDoctor(4000, 4, "Yamen Ali", "Syria/Damascus", Date.setDate(16, 3, 2001), Date.setDate(15, 3, 2021));
344     ContractedDoctor dr4 = new ContractedDoctor(5, "Manar Abdo", "Syria/Daraa", Date.setDate(26, 5, 1992), Date.setDate(10, 10, 2018));
345     ContractedDoctor dr5 = new ContractedDoctor(6, "Mazen Ashqar", "Syria/Homs", Date.setDate(6, 10, 1987), Date.setDate(17, 1, 2017));
346     ContractedDoctor dr6 = new ContractedDoctor(7, "Yara Rayyes", "Syria/Damascus", Date.setDate(4, 7, 1996), Date.setDate(2, 4, 2017));
347     ContractedDoctor dr7 = new ContractedDoctor(7, "Sama Sandouk", "Syria/Damascus", Date.setDate(24, 5, 1992), Date.setDate(8, 4, 2017));
348     SpecialistDoctor dr8 = new SpecialistDoctor(6000, 1, "Ahmad Youcef", "Syria/Damascus", Date.setDate(15, 4, 1980));
349     SpecialistDoctor dr9 = new SpecialistDoctor(6500, 1, "Amal AlKhateeb", "Syria/Damascus", Date.setDate(13, 9, 1988));
350     SpecialistDoctor dr10 = new SpecialistDoctor(7500, 2, "Anas Alloush", "Syria/Daraa", Date.setDate(27, 6, 1975));
351     SpecialistDoctor dr11 = new SpecialistDoctor(7000, 2, "Kareem Alaswad", "Syria/Damascus", Date.setDate(1, 1, 1971));
352     SpecialistDoctor dr12 = new SpecialistDoctor(6500, 2, "Marwa Haddad", "Syria/Damascus", Date.setDate(12, 7, 1990));
353     SpecialistDoctor dr13 = new SpecialistDoctor(8500, 3, "Nuha Haddad", "Syria/Damascus", Date.setDate(2, 5, 1960));
354     SpecialistDoctor dr14 = new SpecialistDoctor(6500, 3, "Kamal Shareef", "Syria/Damascus", Date.setDate(19, 6, 1984));
355     SpecialistDoctor dr15 = new SpecialistDoctor(7000, 4, "Yasser Aallam", "Syria/Homs", Date.setDate(8, 8, 1989));
356     SpecialistDoctor dr16 = new SpecialistDoctor(6000, 4, "Ghazal Mohammad", "Syria/Damascus", Date.setDate(18, 11, 1996));
357     drs.addAll(asList(dr1, dr2, dr3, dr4, dr5, dr6, dr7, dr8, dr9, dr10, dr11, dr12, dr13, dr14, dr15, dr16));
358 //-----
359     InnerPatient p1 = new InnerPatient("Baraa AlAli", "Syria/Latakia", Date.setDate(5, 8, 2000)); p1.setDepNum(1);
360     InnerPatient p2 = new InnerPatient("Rama Khalaf", "Syria/AlHasaka", Date.setDate(15, 2, 1995)); p2.setDepNum(1);
361     InnerPatient p3 = new InnerPatient("Majed AlNaeemat", "Jordan/AlZargaa", Date.setDate(3, 11, 1993)); p3.setDepNum(2);
362     InnerPatient p4 = new InnerPatient("Gabriel Eid", "Syria/Damascus", Date.setDate(22, 9, 1996)); p4.setDepNum(2);
363     InnerPatient p5 = new InnerPatient("Saleh Rasheed", "Egypt/Cairo", Date.setDate(14, 1, 1985)); p5.setDepNum(3);
364     InnerPatient p6 = new InnerPatient("Hayat Najem", "Syria/Hamah", Date.setDate(7, 11, 1975)); p6.setDepNum(3);
365     InnerPatient p7 = new InnerPatient("Boshra Marta", "Syria/Homs", Date.setDate(29, 5, 1977)); p7.setDepNum(4);
366     OuterPatient p8 = new OuterPatient("Kareem Ahmad", "Syria/Tartous", Date.setDate(25, 7, 2002)); p8.setDepNum(5);
367     OuterPatient p9 = new OuterPatient("Hamza AlHaj", "Syria/Daraa", Date.setDate(1, 1, 1967)); p9.setDepNum(6);
368     OuterPatient p10 = new OuterPatient("Ghazi Mounir", "Jordan/Amman", Date.setDate(1, 6, 1989)); p10.setDepNum(6);
369     OuterPatient p11 = new OuterPatient("Mahmoud AlHelo", "Syria/Damascus", Date.setDate(9, 9, 1963)); p11.setDepNum(7);
370     OuterPatient p12 = new OuterPatient("Yaman AlHalak", "Syria/Aleppo", Date.setDate(16, 1, 2007)); p12.setDepNum(7);
371     OuterPatient p13 = new OuterPatient("Qusay Najjar", "Syria/Damascus", Date.setDate(6, 4, 1983)); p13.setDepNum(7);
372     patients.addAll(asList(p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13));
373 //-----
374 }
375 }
```

تم بعونه تعالى.....