

Ministerul Educației și Cercetării al Republicii Moldova
Centrul de Excelență în Informatică și Tehnologii Informaționale
Catedra Informatică II

RAPORT

STUDIU INDIVIDUAL

NR. 1

Unitatea de curs: **PROGRAMAREA ORIENTATĂ PE
OBIECTE**

Tema: Crearea unei aplicații consolă. Clase și obiecte.

Grupa: W-2223

Elev(ul/a): Belii Alexandra

Profesor: MUSTEAȚĂ Victoria

Chișinău, 2024

Cuprins

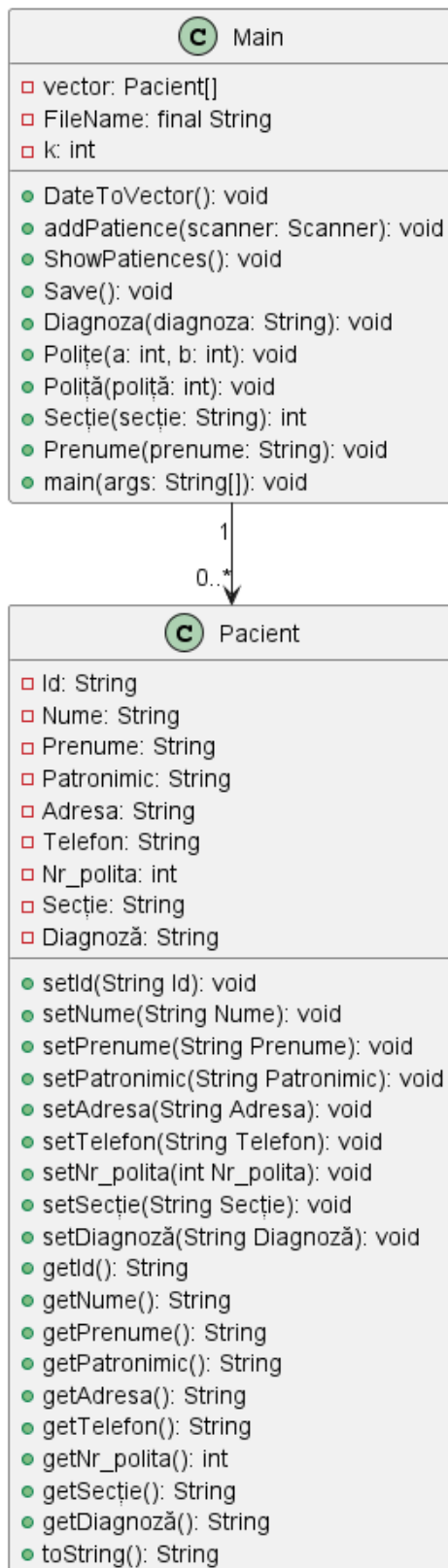
I.Enunțul problemei	3
II. Schema proiectului	4
III. Codul sursă repartizat pe clase	5
IV.Teste	10
V.Concluzie.....	13
Bibliografie	14

I.Enunțul problemei

Pacient. Elaborați un program Java care va colecta date despre un grup de pacienți. Despre pacienți este necesar de colectat următoarea informație: id, Nume, Prenume, Patronimic, Adresa, Telefon, NumarPolitaAsigurare, Sectie, Diagnoza. Operați cu colecții de obiecte de tip Pacient. Prevedeți și tratați situațiile excepționale ce pot apărea la operarea cu datele de prelucrat. Datele să se salveze într-un fișier în care deja există 10 pacienți. Aplicația să permită efectuarea următoarelor operații:

- a) Introducerea unui nou pacient în fișier;
- b) Afișarea datelor din fișier;
- c) Afișarea listei pacienților cu o diagnoză anumită;
- d) Afișarea datelor pacienților cu numărul poliței de asigurare cuprinse într-un interval anumit;
- e) Afișarea datelor pacientului cu un număr anumit al poliței de asigurare;
- f) Afișarea numărului de pacienți dintr-o secție anumită;
- g) Afișarea datelor pacienților după un prenume anumit

II. Schema proiectului



III. Codul sursă repartizat pe clase

- Clasa Main(unde are loc toată distracția)

```
import java.io.*;
import java.util.*;

public class Main {
    public static Pacient[] vector = new Pacient[1000];
    public static final String FileName = "pacienti.txt";
    private static int k = 0;

    static public void DateToVector() {
        try(ObjectInputStream cin = new ObjectInputStream(new
FileInputStream(FileName))) {
            Pacient pacient1;
            while ((pacient1 = (Pacient) cin.readObject()) != null){
                vector[k] = pacient1;
                k++;
            }
        }
        catch (EOFException eof){}
        catch (IOException | ClassNotFoundException ex){}
    }

    public static void addPatience(Scanner scanner){
        Pacient pacient = new Pacient();
        try {
            System.out.print("Id(ex.:001):");
            pacient.setId(scanner.nextLine());
            System.out.print("Nume(ex.:Belîi):");
            pacient.setNume(scanner.nextLine());
            System.out.print("Prenume(ex.:Alexandra):");
            pacient.setPrenume(scanner.nextLine());
            System.out.print("Patronimic(ex.:Leonid):");
            pacient.setPatronimic(scanner.nextLine());
            System.out.print("Adresa(ex.:Livezilor_1):");
            pacient.setAdresa(scanner.nextLine());
            System.out.print("Telefon(ex.:078064842):");
            pacient.setTelefon(scanner.nextLine());
            System.out.print("Numar Polita Asigurare(ex.:123):");
            pacient.setNr_polita(scanner.nextInt());
            scanner.nextLine();
            System.out.print("Sectie(ex.:Chirurgie):");
            pacient.setSecție(scanner.nextLine());
            System.out.print("Diagnoza(ex.:Panic attacks):");
            pacient.setDiagnoză(scanner.nextLine());

            vector[k] = pacient;
            k++;
            System.out.println("\u001B[33mPacientul a fost adăugat cu
succes în fișier.\u001B[0m");
        }
        catch (InputMismatchException e){
            System.out.println("Ai introdus un tip de date nevalabil");
        }
    }
}
```

```

    public static void ShowPatiences () {
        if (k == 0) {
            System.out.println("\u001B[35mÎn fișier nu sunt înscriși
pacienți\u001B[0m");
            return;
        }
        try {

            System.out.println("Lista pacienților din fișier:");
            for(int i = 0; i < vector.length; i++)

System.out.println(vector[i].toString()); //System.out.println(studenti[i])
;
        }
        catch (NullPointerException e) {}
    }

    static void Save() {
        try (ObjectOutputStream outputStream = new ObjectOutputStream(new
FileOutputStream(fileName))) {
            for (int i = 0; i < k; i++) {
                outputStream.writeObject(vector[i]);
            }
        }
        catch (IOException e) {}
    }

    public static void Diagnoza(String diagnoza) {
        if (k == 0) {
            System.out.println("\u001B[35mÎn fișier nu sunt înscriși
pacienți\u001B[0m");
            return;
        }
        else {
            System.out.println("Lista pacienților cu " + diagnoza + ":");
            for(int i=0; i < k; i++)
                if (vector[i].getDiagnoză().equals(diagnoza))
                    System.out.println(vector[i].toString());
        }
    }

    public static void Polițe(int a, int b) {
        boolean ok = false;
        if (k == 0) {
            System.out.println("\u001B[35mÎn fișier nu sunt înscriși
pacienți\u001B[0m");
            return;
        }
        else {
            for(int i=0; i < k; i++)
                if (vector[i].getNr_polita() >= a &&
vector[i].getNr_polita() <= b)
                    ok = true;
            if (ok == false)
                System.out.println("Nu sunt în fișier pacienți cu numărul
poliței de asigurare cuprins între " + a + " și " + b);
            else {
                System.out.println("Lista pacienților cu numărul poliței
de asigurare de la " + a + " la " + b);
                for(int i=0; i < k; i++)

```

```

        if (vector[i].getNr_polita() >= a &&
vector[i].getNr_polita() <= b)
            System.out.println(vector[i].toString());
    }

}

public static void Poliță(int poliță){
    boolean ok = false;
    if (k == 0){
        System.out.println("\u001B[35mÎn fișier nu sunt înscriși
pacienți\u001B[0m");
        return;
    }
    else {
        for(int i=0; i < k; i++)
            if (vector[i].getNr_polita() == poliță){
                System.out.println("Pacientul cu polița " + poliță +
": \n" + vector[i].toString());
                ok = true;
                return;
            }
        if (ok == false)
            System.out.println("În fișier nu există pacient cu numărul
poliței " + poliță);
    }

}

public static int Secție(String secție){
    int cnt = 0;
    if (k == 0){
        System.out.println("\u001B[35mÎn fișier nu sunt înscriși
pacienți\u001B[0m");
        return 0;
    }
    else {
        for(int i=0; i < k; i++)
            if (vector[i].getSecție().equals(secție))
                cnt++;
    }
    return cnt;
}

public static void Prenume(String prenume){
    if (k == 0)
        System.out.println("\u001B[35mÎn fișier nu sunt înscriși
pacienți\u001B[0m");
    else {
        System.out.println("Lista pacienților cu prenumele " + prenume
+ ":");
        for(int i=0; i < k; i++)
            if (vector[i].getPrenume().equals(prenume))
                System.out.println(vector[i].toString());
    }
}

public static void main(String[] args) throws IOException {
    Scanner scanner = new Scanner(System.in);

```

```

        DateToVector();
        while (true){
            System.out.println("\u001B[32mMeniu\u001B[0m \n" +      "1)
Introducerea unui nou pacient în fișier;\n" +                      "2)
Afișarea datelor din fișier;\n" +                                  "3)
Afișarea listei pacienților cu o diagnoză anumită;\n" +           "4)
Afișarea datelor pacienților cu numărul poliței de asigurare cuprinse
într-un interval anumit;\n" +                                     "5)
Afișarea datelor pacientului cu un număr anumit al poliței de
asigurare;\n" +                                                  "6)
Afișarea numărului de pacienți dintr-o secție anumită;\n" +      "7)
Afișarea datelor pacienților după un prenume anumit");
            System.out.println("Alegerea ta:");
            int n = scanner.nextInt();
            scanner.nextLine();
            switch (n){
                case 1:
                    addPatience(scanner); Save(); break;

                case 2:
                    ShowPatiences();
                    break;

                case 3:
                    System.out.print("Ce diagnoză vă interesează:
");
                    String diagnoza = scanner.nextLine();
                    Diagnoza(diagnoza); break;

                case 4:
                    System.out.print("Te interesează pacienții cu
numărul poliței de asigurare cuprinse între: ");
                    int a = scanner.nextInt();
                    System.out.print(" și ");
                    int b = scanner.nextInt();
                    Polițe(a, b);
                    break;

                case 5:
                    System.out.print("Numărul poliței: ");
                    int poliță = scanner.nextInt();
                    scanner.nextLine();
                    Poliță(poliță); break;

                case 6:
                    System.out.println("În care secție te
interesează câți pacienți sunt: ");
                    String secție = scanner.nextLine();
                    System.out.println("În secția " + secție + "
sunt " + Secție(secție) + " pacienți"); break;

                case 7: System.out.print("Lista pacienților cu ce
prenume: ");

```



```

        String prenume = scanner.nextLine();
        Prenume(prenume);        break;
    default:
        System.err.println("Numarul introdus trebuie
să aparțină intervalului [1-7].");
    }

    System.out.println("\n=====");

    System.out.println("Introduceti \u001B[36mSTOP\u001B[0m
pentru" + " a opri executia sau orice pentru a continua");
    String mesaj = scanner.nextLine();
    if(mesaj.equalsIgnoreCase("stop")) break;
}
}
}

```

- Clasa Pacient

```

• import java.io.Serializable;

public class Pacient implements Serializable {
    private String Id, Nume, Prenume, Patronimic, Adresa, Telefon,
    Secție, Diagnoză;
    private int Nr_polita;

    Pacient() {}

    //setters
    public void setId(String Id){ this.Id = Id;}
    public void setNume(String Nume){ this.Nume = Nume;}
    public void setPrenume(String Prenume){ this.Prenume = Prenume;}
    public void setPatronimic(String Patronimic){ this.Patronimic =
Patronimic;}
    public void setAdresa(String Adresa){ this.Adresa = Adresa;}
    public void setTelefon(String Telefon){ this.Telefon = Telefon;}
    public void setNr_polita(int Nr_polita){ this.Nr_polita =
Nr_polita;}
    public void setSecție(String Secție){ this.Secție = Secție;}
    public void setDiagnoză(String Diagnoză){ this.Diagnoză =
Diagnoză;}

    //getters
    String getId(){ return Id;}
    String getNume(){ return Nume;}
    String getPrenume(){ return Prenume;}
    String getPatronimic(){ return Patronimic;}
    String getAdresa(){ return Adresa;}
    String getTelefon(){ return Telefon;}
    int getNr_polita(){ return Nr_polita;}
    String getSecție(){ return Secție;}
    String getDiagnoză(){ return Diagnoză;}

    @Override
    public String toString() {
        return getId() + " " + getNume() + " " + getPrenume() + " " +
getPatronimic() + " " + getAdresa() + " " + getTelefon() + " " +
getNr_polita() + " " + getSecție() + " " + getDiagnoză();
    }
}

```

IV.Teste

Meniu

- 1) Introducerea unui nou pacient în fișier;
 - 2) Afișarea datelor din fișier;
 - 3) Afișarea listei pacienților cu o diagnoză anumită;
 - 4) Afișarea datelor pacienților cu numărul poliței de asigurare cuprinse într-un interval anumit;
 - 5) Afișarea datelor pacientului cu un număr anumit al poliței de asigurare;
 - 6) Afișarea numărului de pacienți dintr-o secție anumită;
 - 7) Afișarea datelor pacienților după un prenume anumit
- Alegerea ta:

1

Id(ex.:001):012

Nume(ex.:Belii):Madan

Prenume(ex.:Alexandra):Marin

Patronimic(ex.:Leonid):Petrică

Adresa(ex.:Livezilor_1):Mănăstirea_Piatra_Neamț

Telefon(ex.:078064842):078934215

Numar Polita Asigurare(ex.:123):123

Sectie(ex.:Chirurgie):Psihiatrie

Diagnoza(ex.:Panic attacks):Psihoză

Pacientul a fost adăugat cu succes în fișier.

=====

Introduceti STOP pentru a opri executia sau orice pentru a continua

|

Alegerea ta:

7

Lista pacienților cu ce prenume: Ion

Lista pacienților cu prenumele Ion:

005 Tudor Ion Constntin Victoriei_30 072314657 115 Chirurgie_generală Apendicită

010 Gheorghe Ion Dan Decebal_7 073482954 120 Urologie Litiază_renală

Meniu

- 1) Introducerea unui nou pacient în fișier;
 - 2) Afișarea datelor din fișier;
 - 3) Afișarea listei pacienților cu o diagnoză anumită;
 - 4) Afișarea datelor pacienților cu numărul poliței de asigurare cuprinse într-un interval anumit;
 - 5) Afișarea datelor pacientului cu un număr anumit al poliței de asigurare;
 - 6) Afișarea numărului de pacienți dintr-o secție anumită;
 - 7) Afișarea datelor pacienților după un prenume anumit
- Alegerea ta:

2

Lista pacienților din fișier:

001 Delia Matache Ion Mihai_Viteazu_10 079685473 111 Ortopedie Fractură_de_femur

002 Ionescu Andrei Vasile Unirii_20 0765987654 112 Cardiologie Infarct_miocardic

003 Stan Ana Maria Libertății_5 0789456123 113 Ginecologie Sarcină_ectopică

004 Dragomir Andrei Mihai Dorobanților_15 0741852963 114 Neurologie Accident_vascular_cerebral

005 Tudor Ion Constntin Victoriei_30 072314657 115 Chirurgie_generală Apendicită

006 Popa Elena Vasile Mihai_Eminescu_25 072132456 116 Dermatologie Eczemă

007 Marinescu Andrei Alexandru Avram_Iancu_12 078934526 117 Oftalmologie Cataractă

008 Dumitrescu Maria George Republicii_18 067483295 118 Psihiatrie Depresie

009 Rădulescu Ioana Mihai Calea_Dorobanților_4 078987654 119 Endocrinologie Diabet_zaharat

010 Gheorghe Ion Dan Decebal_7 073482954 120 Urologie Litiază_renală

012 Madan Marin Petrică Mănăstirea_Piatra_Neamț 078934215 123 Psihiatrie Psihoză

Meniu

- 1) Introducerea unui nou pacient în fișier;
- 2) Afișarea datelor din fișier;
- 3) Afișarea listei pacienților cu o diagnoză anumită;
- 4) Afișarea datelor pacienților cu numărul poliței de asigurare cuprinse într-un interval anumit;
- 5) Afișarea datelor pacientului cu un număr anumit al poliței de asigurare;
- 6) Afișarea numărului de pacienți dintr-o secție anumită;
- 7) Afișarea datelor pacienților după un prenume anumit

Alegerea ta:

3

Ce diagnoză vă interesează: *Depresie*

Lista pacienților cu Depresie:

008 Dumitrescu Maria George Republicii_18 067483295 118 Psihiatrie Depresie

Meniu

- 1) Introducerea unui nou pacient în fișier;
- 2) Afișarea datelor din fișier;
- 3) Afișarea listei pacienților cu o diagnoză anumită;
- 4) Afișarea datelor pacienților cu numărul poliței de asigurare cuprinse într-un interval anumit;
- 5) Afișarea datelor pacientului cu un număr anumit al poliței de asigurare;
- 6) Afișarea numărului de pacienți dintr-o secție anumită;
- 7) Afișarea datelor pacienților după un prenume anumit

Alegerea ta:

4

Te interesează pacienții cu numărul poliței de asigurare cuprinse între: *114*
și *119*

Lista pacienților cu numărul poliței de asigurare de la 114 la 119

004 Dragomir Andrei Mihai Dorobanților_15 0741852963 114 Neurologie Accident_vascular_cerebral

005 Tudor Ion Constantin Victoriei_30 072314657 115 Chirurgie_generală Apendicită

006 Popa Elena Vasile Mihai_Eminescu_25 072132456 116 Dermatologie Eczemă

007 Marinescu Andrei Alexandru Avram_Iancu_12 078934526 117 Oftalmologie Cataractă

008 Dumitrescu Maria George Republicii_18 067483295 118 Psihiatrie Depresie

009 Rădulescu Ioana Mihai Calea_Dorobanților_4 078987654 119 Endocrinologie Diabet_zaharat

Alegerea ta:

5

Numărul poliței: *112*

Pacientul cu polița 112:

002 Ionescu Andrei Vasile Unirii_20 0765987654 112 Cardiologie Infarct_miocardic

Alegerea ta:

6

În care secție te interesează câți pacienți sunt:

Neurologie

În secția Neurologie sunt 1 pacienți

Meniu

- 1) Introducerea unui nou pacient în fișier;
- 2) Afișarea datelor din fișier;
- 3) Afișarea listei pacienților cu o diagnoză anumită;
- 4) Afișarea datelor pacienților cu numărul poliței de asigurare cuprinse într-un interval anumit;
- 5) Afișarea datelor pacientului cu un număr anumit al poliței de asigurare;
- 6) Afișarea numărului de pacienți dintr-o secție anumită;
- 7) Afișarea datelor pacienților după un prenume anumit

Alegerea ta:

8

Numarul introdus trebuie să aparțină intervalului [1-7].

V.Concluzie

Mă numesc Belîi Alexandra și am avut e realizat primul studiu la Java.

Inițial eram sigură că studiul va fi ușor. Am vrut să-i adaug complexitate și într-o seară am început a crea GUI, apoi am realizat că în primul rând am nevoie de programul brut. Pentru că studiasem tema fișiere destul de bine, dar la laborator nu implementasem serializarea am decis că studiul e mediul perfect unde poți să experimentezi și pierzi vreo doi neuroni. Codul meu inițial încă nu îmi merge și versiuni la acest studiu oficial sunt 2, neoficial am pierdut numărătoarea.

Încă mă supără că în varianta aceasta lucrez cu vectori statici, ăia dinamici mi se pare că dau un aer de profesionist.

Fiecare studiu poate fi dezvoltat ca o mică aplicație cu puțin GUI, dar asta e ceva mai multă muncă și după toată frustrarea pe care o adus-o studiul ăsta, double it and give it to the next person.

Nu zic că a fost un studiu greu, dar anume deserializarea mi-a făcut feri suri și faptul că mai mereu schimbând ceva cât de mic, duceam la a ucide programul. Git nu ajută, face mai rău în cazul meu la moment, încercat.

Pe viitor trebuie să îmi organizez lucrul mai bine și să văd cum fac să nu am zece mii de copii și nici una funcțională total, probabil deadline-ul e soluția (scriu asta la 3:20).

Ce ține de funcționalități noi în proiect – orice dorește clientul e poruncă(dacă plătește bine).

În concluzie studiile sunt chestii benefice, dar uneori stresante (dacă ai prea multe idei pe care vrei să le impementezi, s-ar putea să rămâi la versiunea simplă doar pentru că ești presat de timp și/sau obosit de unul și același proiect)

Bibliografie

<https://www.geeksforgeeks.org/java-util-vector-class-java/>

<https://www.geeksforgeeks.org/how-to-print-colored-text-in-java-console/>

<https://docs.oracle.com/en/java/javase/22/>

[https://drive.google.com/drive/folders/1Rs3PIHuLQONXDQeJymcKFC63yl31bhjE?usp=drive link](https://drive.google.com/drive/folders/1Rs3PIHuLQONXDQeJymcKFC63yl31bhjE?usp=drive_link)