Họ và tên: Trần Thị Duyên

MSSV: K185480106004

Lớp: K54KMT.01

**BÁO CÁO BÀI TẬP THỰC HÀNH**

**MÔN LẬP TRÌNH JAVA**

**Bài tập 1:** Nhập một danh sách gồm n sinh viên bao gồm các thuộc tính: MSSV, họ tên SV, địa chỉ, giới tính, quê quán, email, điểm 1, điểm 2, điểm 3, điểmTB. Yêu cầu:

-Tính điểm trung bình cho danh sách sinh viên

-In ra màn hình thông tin sinh viên có điểm trung bình cao nhất

-In ra danh sách sinh viên đạt học bổng. biết rằng sinh viên đạt học bổng là sinh viên có điểmTB>7.0 và không có điểm nào dưới 5

-Sắp xếp danh sách sinh viên theo thứ tự tăng dần của điểm TB

-Sắp xếp danh sách sinh viên theo thứ tự từ điển

-Cho biết thông tin của những sinh viên nữ, có điểm TB>7.0 và điểm 2 >6

**a.Ý tưởng:**

-Tạo 1 Class SinhVien:

+ có các thuộc tính như đề cho, tạo các **getters, setters**, hàm **inpuInfo()** để nhập thông tin, **showInfo()** để xuất thông tin sinh viên.( có kết hợp nhập có điều kiện đối với 1 số thuộc tính như: giới tính(nam/nữ), email(có @ và k có khoảng trắng), điểm(0-10)).

+ phương thức **checkHocBong()** để kiểm tra đủ điều kiện đạt học bổng không

-Class ListSinhVien: Chứa hàm main giải quyết các yêu cầu bài toán

+Sử dụng ArrayList để lưu danh sách sinh viên

+Đối với các ý yêu cầu sắp xếp thì sử dụng Collections.sort() là nhanh nhất

**b. Code:**

\*Class Sinh Viên:

**package** duyen.com.model;

**import** java.util.Scanner;

**public** **class** SinhVien {

**private** String mssv;

**private** String hoTen;

**private** String diaChi;

**private** String sex;

**private** String que;

**private** String email;

**private** **double** diem1;

**private** **double** diem2;

**private** **double** diem3;

**private** **double** diemTB;

**public** SinhVien() {

**super**();

}

**public** SinhVien(String mssv, String hoTen, String diaChi, String sex, String que, String email, **double** diem1,

**double** diem2, **double** diem3, **double** diemTB) {

**super**();

**this**.mssv = mssv;

**this**.hoTen = hoTen;

**this**.diaChi = diaChi;

**this**.sex = sex;

**this**.que = que;

**this**.email = email;

**this**.diem1 = diem1;

**this**.diem2 = diem2;

**this**.diem3 = diem3;

**this**.diemTB = diemTB;

}

**public** String getMssv() {

**return** mssv;

}

**public** **void** setMssv(String mssv) {

**this**.mssv = mssv;

}

**public** String getHoTen() {

**return** hoTen;

}

**public** **void** setHoTen(String hoTen) {

**this**.hoTen = hoTen;

}

**public** String getDiaChi() {

**return** diaChi;

}

**public** **void** setDiaChi(String diaChi) {

**this**.diaChi = diaChi;

}

**public** String getSex() {

**return** sex;

}

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

**if**(sex.contentEquals("nữ")==**true**|| sex.contentEquals("nam") == **true**) {

**this**.sex = sex;

**return** **true**;

}

**else** {

System.***err***.println("=>Nhập lại giới tính('nam' hoặc 'nữ'): ");

**return** **false**;

}

}

**public** String getQue() {

**return** que;

}

**public** **void** setQue(String que) {

**this**.que = que;

}

**public** String getEmail() {

**return** email;

}

**public** **boolean** setEmail(String email) {

**if**(email!=**null** && email.contains("@")&& !email.contains(" ")) {

**this**.email = email;

**return** **true**;

}

**else** {

System.***err***.println("=>Nhập lại email(email#null và chứa @): ");

**return** **false**;

}

}

**public** **double** getDiem1() {

**return** diem1;

}

**public** **boolean** setDiem1(**double** diem1) {

**if**(diem1>=0 && diem1<=10) {

**this**.diem1 = diem1;

**return** **true**;

}

**else** {

System.***err***.println("Nhập lại điểm(0<=điểm<=10): ");

**return** **false**;

}

}

**public** **double** getDiem2() {

**return** diem2;

}

**public** **boolean** setDiem2(**double** diem2) {

**if**(diem1>=0 && diem1<=10) {

**this**.diem2 = diem2;

**return** **true**;

}

**else** {

System.***err***.println("Nhập lại điểm(0<=điểm<=10): ");

**return** **false**;

}

}

**public** **double** getDiem3() {

**return** diem3;

}

**public** **boolean** setDiem3(**double** diem3) {

**if**(diem3>=0 && diem3<=10) {

**this**.diem3 = diem3;

**return** **true**;

}

**else** {

System.***err***.println("Nhập lại điểm(0<=điểm<=10): ");

**return** **false**;

}

}

**public** **void** setDiemTB(**double** d1, **double** d2, **double** d3) {

diemTB = (d1+d2+d3)/3;

}

**public** **double** getDiemTB() {

**return** diemTB;

}

**public** **void** inputInfo()

{

Scanner input= **new** Scanner(System.***in***);

System.***out***.print("-MSSV: ");

mssv=input.nextLine();

System.***out***.print("-Tên sinh viên: ");

hoTen=input.nextLine();

System.***out***.print("-Địa chỉ: ");

diaChi=input.nextLine();

System.***out***.print("-Giới tính: ");

**while**(**true**)

{

String sexInput=input.nextLine();

**boolean** check =setSex(sexInput);

**if**(check)

**break**;

}

System.***out***.print("-Quê quán: ");

que=input.nextLine();

System.***out***.print("-Email: ");

**while**(**true**)

{

String emailInput=input.nextLine();

**boolean** check =setEmail(emailInput);

**if**(check)

**break**;

}

System.***out***.print("-Điểm 1: ");

**while**(**true**) {

Double diem1Input=Double.*parseDouble*(input.nextLine());

**boolean** check1 =setDiem1(diem1Input);

**if**(check1)

**break**;

}

System.***out***.print("-Điểm 2: ");

**while**(**true**) {

Double diem2Input=Double.*parseDouble*(input.nextLine());

**boolean** check2 =setDiem2(diem2Input);

**if**(check2)

**break**;

}

System.***out***.print("-Điểm 3: ");

**while**(**true**) {

Double diem3Input=Double.*parseDouble*(input.nextLine());

**boolean** check3 =setDiem3(diem3Input);

**if**(check3)

**break**;

}

setDiemTB(diem1, diem2, diem3);

}

**public** **void** showInfo()

{

System.***out***.print("\n-MSSV: "+mssv);

System.***out***.print("\n-Tên sinh viên: "+hoTen);

System.***out***.print("\n-Địa chỉ: "+diaChi);

System.***out***.print("\n-Giới tính: "+sex);

System.***out***.print("\n-Quê quán: "+que);

System.***out***.print("\n-Email: "+email);

System.***out***.print("\n-Điểm 1: "+diem1);

System.***out***.print("\n-Điểm 2: "+diem2);

System.***out***.print("\n-Điểm 3: "+diem3);

System.***out***.print("\n-Điểm TB = "+ diemTB+"\n\n");

}

**public** **boolean** checkHocBong()

{

**if**(diemTB>=7 && diem1>5 && diem2>5 &&diem3>5) {

**return** **true**;

}

**return** **false**;

}

**public** **boolean** checkGioiTinh() {

**if**(sex.contentEquals("nữ")==**true**) {

**return** **true**;

}

**else** **return** **false**;

}

}

**package** duyen.com.test;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.Comparator;

**import** java.util.Scanner;

**import** duyen.com.model.SinhVien;

**public** **class** ListSinhVien {

**static** **void** menu()

{

System.***out***.println("\n>>>>>>>>>>>>>>> MENU <<<<<<<<<<<<<<<");

System.***out***.println("1. Nhập n sinh viên");

System.***out***.println("2. Hiển thị thông tin sinh viên");

System.***out***.println("3. Thông tin sinh viên có điểm TB max ");

System.***out***.println("4. Danh sách sinh viên đạt học bổng");

System.***out***.println("5. Danh sách sinh viên theo thứ tự điểm trung bình tăng dần");

System.***out***.println("6. Danh sách sinh viên theo thứ tự từ điển");

System.***out***.println("7. Danh sách sinh viên nữ có điểm TB >7.0 và điểm 2>6");

System.***out***.println("8. Thoát");

}

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

ArrayList<SinhVien> ds=**new** ArrayList<>();

**int** choose,dem;

Scanner sc= **new** Scanner(System.***in***);

**do** {

*menu*();

System.***out***.println("Choose: ");

choose =Integer.*parseInt*(sc.nextLine());

**switch**(choose)

{

**case** 1:

**int** n;

System.***out***.println("Nhập số lượng sinh viên: ");

n=Integer.*parseInt*(sc.nextLine());

**for**(**int** i=0; i<n;i++) {

System.***out***.println("\n"+(i+1));

SinhVien std= **new** SinhVien();

std.inputInfo();

ds.add(std);

}

**break**;

**case** 2:

System.***out***.println("=====DANH SÁCH SINH VIÊN=====");

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

{

System.***out***.print( (i+1)+".");

ds.get(i).showInfo();

}

**break**;

**case** 3:

**int** maxIndex=0;

**double** maxDTB;

maxDTB=ds.get(0).getDiemTB();

**for**(**int** i=1;i<ds.size();i++)

{

**if**(ds.get(i).getDiemTB()>maxDTB) {

maxDTB=ds.get(i).getDiemTB();

maxIndex=i;

}

}

System.***out***.println("--------Thông tin SV có ĐTB cao nhất--------");

ds.get(maxIndex).showInfo();

**break**;

**case** 4:

System.***out***.println("--------DSSV đạt học bổng--------");

dem=0;

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

**if**(ds.get(i).checkHocBong()==**true**)

{

dem+=1;

System.***out***.print(dem+".");

ds.get(i).showInfo();

}

}

**break**;

**case** 5:

Collections.*sort*(ds,**new** Comparator<SinhVien>() {

@Override

**public** **int** compare(SinhVien o1, SinhVien o2) {

**return** o1.getDiemTB()>=o2.getDiemTB()?1:-1;

}

});

System.***out***.println("--------DSSV theo thứ tự điểm trung bình tăng dần--------");

dem=0;

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

dem+=1;

System.***out***.print(dem+".");

ds.get(i).showInfo();

}

**break**;

**case** 6:

Collections.*sort*(ds,**new** Comparator<SinhVien>() {

@Override

**public** **int** compare(SinhVien o1, SinhVien o2) {

**int** cmp=o1.getHoTen().compareTo(o2.getHoTen());

**if**(cmp>=0)

{

**return** 1;

}

**return** -1;

}

});

System.***out***.println("--------DSSV theo thứ tự từ điển--------");

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

System.***out***.print((i+1)+".");

ds.get(i).showInfo();

}

**break**;

**case** 7:

System.***out***.println("-------- DSSV nữ có điểm TB lớn hơn 7.0 và điểm 2 lớn hơn 6.0 --------");

dem=0;

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

**if**(ds.get(i).checkGioiTinh()==**true** && ds.get(i).getDiemTB()>7.0 && ds.get(i).getDiem2()>6)

{

dem+=1;

System.***out***.print((dem+"."));

ds.get(i).showInfo();

}

}

**break**;

**case** 8:

System.***out***.println(" Goodbye!!!!!!!!!!");

**break**;

**default**:

System.***err***.println("Nhập lụi rồi nha");

}

}**while**(choose!=8);

}

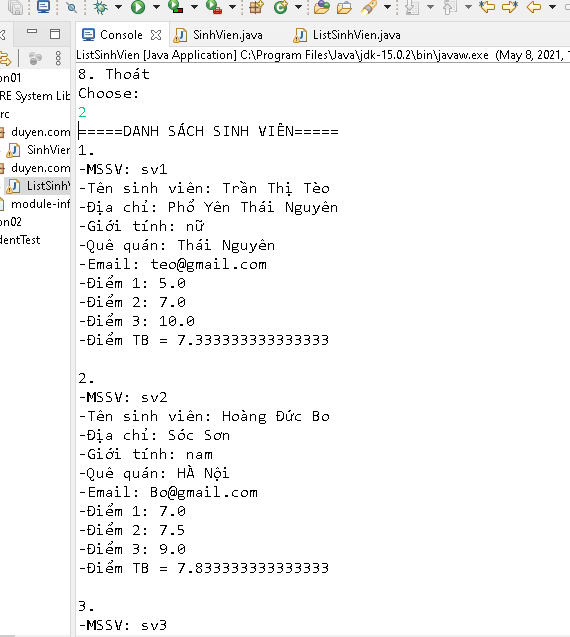
}

**c. Kết quả:**

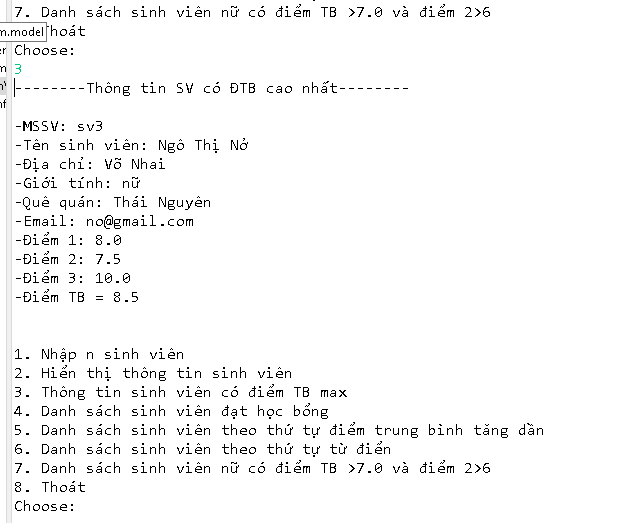
**-**Nhập danh sách sinh viên( có kiểm soát 1 số lỗi khi người dùng nhập như ví dụ):

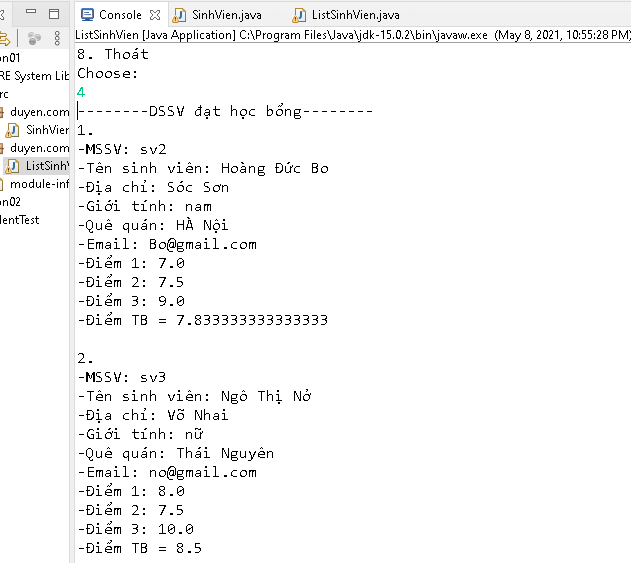
|  |  |
| --- | --- |
|  |  |

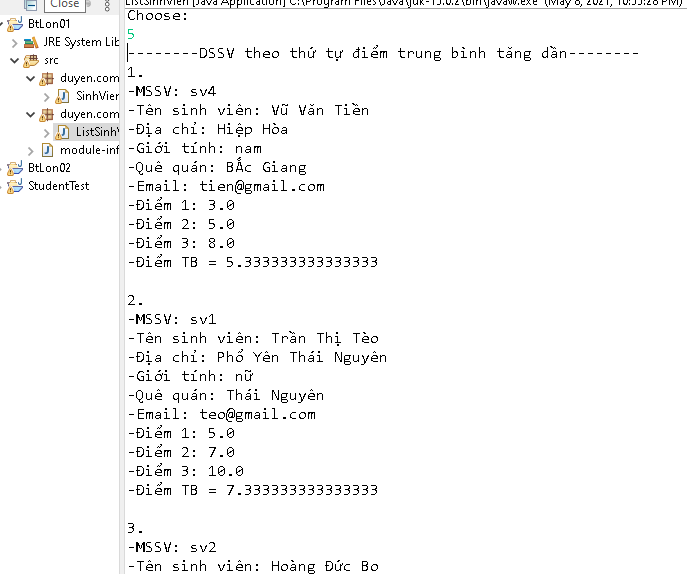
**-**Hiển Thị danh sách sinh viên:

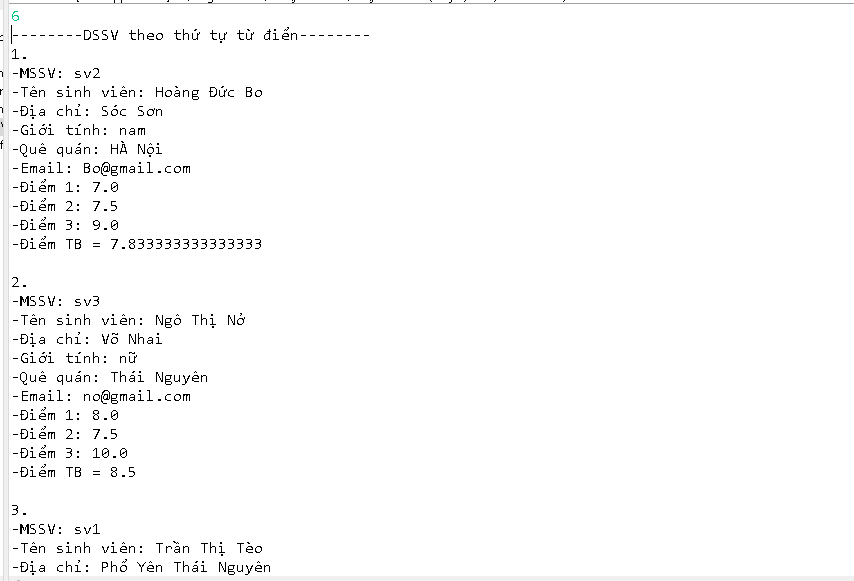


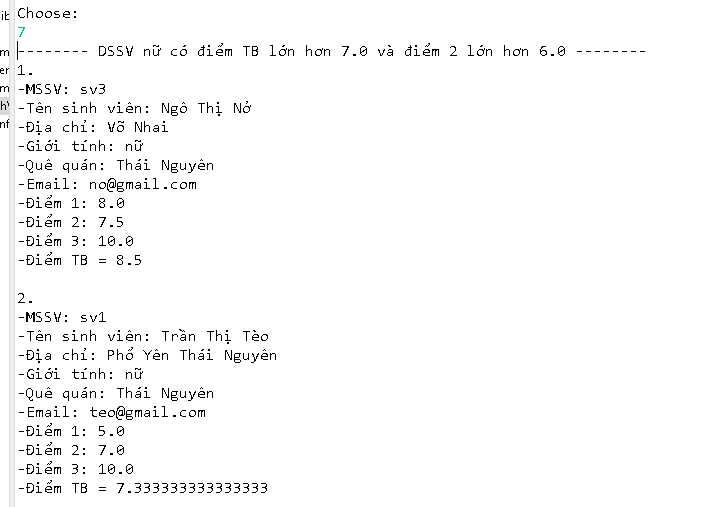
**-**Giải đáp các yêu cầu còn lại:











**Bài tập 2:** Nhập vào một danh sách bệnh nhân bao gồm: MaBN, tênBN, địa chỉ, Đienthoai, Giới tính, ngày sinh, bảo hiểm y tế.

Một danh sách bệnh án tương đương với danh sách bệnh nhân nêu trên bao gồm các thuộc tính: Mã BA, tenBA, khoa, phòng, ngày vao, ngày ra, viện phí.

Yêu cầu:

-Tính viện phí của danh sách bệnh nhân trên. Biết rằng

-viện phí= số ngày nằm viện\*200ngan – bảo hiểm y tế(80% đúng tuyến, 50% trái tuyến, nếu là bệnh nhân<5 tuổi thì bằng 100%)

-In ra màn hình danh sách bệnh nhân dưới 5 tuổi

-In ra màn hình danh sách những bệnh nhân >70 tuổi nằm khoa Ngoại

-Cho biết danh sách bệnh nhân có tiền viện phí cao nhất

-Cho biết từ ngày 20/2/2021 đến 13/4/2021 có bao nhiêu bệnh nhân nằm viện

-Cho biết từ ngày 20/2/2021 đến 13/4/2021 có bao nhiêu bệnh nhân được ra viện

-Tính tổng số tiền viện phí thu được trong quý 1 năm 2021(từ 1/1/2021-31/3/2021)

-In ra danh sách bệnh nhân nam 30 tuổi nằm viện trong quý 1

-In ra danh sách bệnh nhân có bảo hiểm y tế và có tiền viện phí>2 triệu

-In ra danh sách bệnh nhân không có bảo hiểm y tế có viện phí >10 triệu

- In ra danh sách bênh nhân dưới<30 tuổi trái tuyến bảo hiểm

a.Ý tưởng

-Tạo Class BenhNhan và BenhAn kế thừa từ bệnh nhân:

+ có các thuộc tính như đề cho, tạo các **getters, setters**, hàm **inpuInfo()** để nhập thông tin, **showInfo()** để xuất thông tin sinh viên.( có kết hợp nhập có điều kiện đối với 1 số thuộc tính như: giới tính(nam/nữ), ngày sinh+ngày vào+ngày ra(dd/MM/yyy), bhyt(trái tuyến, đúng tuyến, không tham gia)).

+ riêng BenhAn: Thêm phương thức SoNgayNamVien(),TuoiCuaBN() để phụ vụ cho việc tính tiền viện phí và nhiều yêu cầu khác của bài

-Class testBenhAn: Chứa hàm main giải quyết các yêu cầu bài toán

+Sử dụng ArrayList để lưu danh sách bệnh án

+Sử dụng 1 số thư viện sẵn có như: compareToIgnoreCase và compareTo để so sánh chuỗi hay các điều kiện bài cho

Note: Cần sử dụng khéo léo SimpleDateFormat và Calendar để xửa lý ngày tháng

b. Code:

**package** duyen.com.model;

**import** java.text.SimpleDateFormat;

**import** java.util.Calendar;

**import** java.util.Date;

**import** java.util.Scanner;

**public** **class** BenhNhan {

**protected** String maBN;

**protected** String tenBN;

**protected** String diaChi;

**protected** String dienThoai;

**protected** String gioiTinh;

**protected** Date ngaySinh;

**protected** String bhyt;

**public** BenhNhan() {

**super**();

}

**public** BenhNhan(String maBN, String tenBN, String diaChi, String dienThoai,

String gioiTinh, Date ngaySinh,String bhyt) {

**super**();

**this**.maBN = maBN;

**this**.tenBN = tenBN;

**this**.diaChi = diaChi;

**this**.dienThoai = dienThoai;

**this**.gioiTinh = gioiTinh;

**this**.ngaySinh = ngaySinh;

**this**.bhyt = bhyt;

}

**public** **void** inputInfo()

{

Scanner in=**new** Scanner(System.***in***);

System.***out***.print("-Mã bệnh nhân: ");

maBN=in.nextLine();

System.***out***.print("-Tên bệnh nhân: ");

tenBN=in.nextLine();

System.***out***.print("-Địa chỉ: ");

diaChi=in.nextLine();

System.***out***.print("-Điện thoại: ");

dienThoai=in.nextLine();

System.***out***.print("-Giới tính(nam/nữ): ");

**while**(**true**)

{

String sexInput=in.nextLine();

**boolean** check =setGioiTinh(sexInput);

**if**(check)

**break**;

}

System.***out***.print("-Ngày sinh(dd/MM/yyyy): ");

**while**(**true**)

{

String ngaySinhInput=in.nextLine();

**boolean** check =setNgaySinh(ngaySinhInput);

**if**(check)

**break**;

}

System.***out***.print("-Bảo hiểm y tế(đúng tuyến/trái tuyến/không tham gia): ");

**while**(**true**)

{

String bhytInput=in.nextLine();

**boolean** check =setBhyt(bhytInput);

**if**(check)

**break**;

}

}

**public** **void** showInfo()

{

System.***out***.print("\n-Mã bệnh nhân: "+ maBN);

System.***out***.print("\n-Tên bệnh nhân: "+ tenBN);

System.***out***.print("\n-Địa chỉ: "+ diaChi);

System.***out***.print("\n-Điện thoại: "+dienThoai);

System.***out***.print("\n-Giới tính: "+ gioiTinh);

SimpleDateFormat sdf=**new** SimpleDateFormat("dd/MM/yyyy");

System.***out***.print("\n-Ngày sinh: "+sdf.format(ngaySinh));

System.***out***.print("\n-Bảo hiểm y tế: "+ bhyt);

}

**public** String getMaBN() {

**return** maBN;

}

**public** **void** setMaBN(String maBN) {

**this**.maBN = maBN;

}

**public** String getTenBN() {

**return** tenBN;

}

**public** **void** setTenBN(String tenBN) {

**this**.tenBN = tenBN;

}

**public** String getDiaChi() {

**return** diaChi;

}

**public** **void** setDiaChi(String diaChi) {

**this**.diaChi = diaChi;

}

**public** String getDienThoai() {

**return** dienThoai;

}

**public** **void** setDienThoai(String dienThoai) {

**this**.dienThoai = dienThoai;

}

**public** String getGioiTinh() {

**return** gioiTinh;

}

**public** **boolean** setGioiTinh(String gioiTinh) {

**if**(gioiTinh.contentEquals("nữ")==**true**|| gioiTinh.contentEquals("nam") == **true**) {

**this**.gioiTinh = gioiTinh;

**return** **true**;

}

**else** {

System.***err***.println("=>Nhập lại giới tính('nam' hoặc 'nữ'): ");

**return** **false**;

}

}

**public** Date getNgaySinh() {

**return** ngaySinh;

}

**public** **boolean** setNgaySinh(String ngaySinh) {

SimpleDateFormat sdf=**new** SimpleDateFormat("dd/MM/yyyy");

**try** {

**this**.ngaySinh=sdf.parse(ngaySinh);

**return** **true**;

} **catch** (Exception e) {

System.***err***.println("&Note: Ngày, tháng hoặc năm không hợp lệ!"

+ " \n=> Mời nhập lại: ");

**return** **false**;

}

}

**public** String getBhyt() {

**return** bhyt;

}

**public** **boolean** setBhyt(String bhyt) {

**if**(bhyt.contentEquals("đúng tuyến")==**true**||bhyt.contentEquals("trái tuyến")==**true**|| bhyt.contentEquals("không tham gia") == **true**) {

**this**.bhyt = bhyt;

**return** **true**;

}

**else** {

System.***err***.println("=>Nhập lại thông tin bhyt('đúng tuyến' hoặc 'trái tuyến') hoặc 'không tham gia: ");

**return** **false**;

}

}

}

**package** duyen.com.model;

**import** java.text.ParseException;

**import** java.text.SimpleDateFormat;

**import** java.util.Calendar;

**import** java.util.Date;

**import** java.time.LocalDate;

**import** java.util.Scanner;

**public** **class** BenhAn **extends** BenhNhan {

**private** String maBA;

**private** String tenBA;

**private** String khoa;

**private** **int** phong;

**private** Date ngayVao;

**private** Date ngayRa;

**private** **float** vienPhi;

**public** BenhAn() {

**super**();

}

**public** BenhAn(String maBA, String tenBA, String khoa, **int** phong, Date ngayVao, Date ngayRa, **float** vienPhi) {

**super**();

**this**.maBA = maBA;

**this**.tenBA = tenBA;

**this**.khoa = khoa;

**this**.phong = phong;

**this**.ngayVao = ngayVao;

**this**.ngayRa = ngayRa;

**this**.vienPhi = vienPhi;

}

@Override

**public** **void** inputInfo() {

// **TODO** Auto-generated method stub

**super**.inputInfo();

Scanner in=**new** Scanner(System.***in***);

System.***out***.print("-Mã bệnh án: ");

maBA=in.nextLine();

System.***out***.print("-Tên bệnh án: ");

tenBA=in.nextLine();

System.***out***.print("-Khoa: ");

khoa=in.nextLine();

System.***out***.print("-Phòng: ");

phong=Integer.*parseInt*(in.nextLine());

System.***out***.print("-Ngày vào(dd/MM/yyyy): ");

**while**(**true**)

{

String ngayVaoInput=in.nextLine();

**boolean** check =setNgayVao(ngayVaoInput);

**if**(check)

**break**;

}

System.***out***.print("-Ngày ra(dd/MM/yyyy): ");

**while**(**true**)

{

String ngayRaInput=in.nextLine();

**boolean** check =setNgayRa(ngayRaInput);

**if**(check)

**break**;

}

// System.out.print("-Viện phí: ");

// vienPhi=Integer.parseInt(in.nextLine());

}

**public** String getMaBA() {

**return** maBA;

}

**public** **void** setMaBA(String maBA) {

**this**.maBA = maBA;

}

**public** String getTenBA() {

**return** tenBA;

}

**public** **void** setTenBA(String tenBA) {

**this**.tenBA = tenBA;

}

**public** String getKhoa() {

**return** khoa;

}

**public** **void** setKhoa(String khoa) {

**this**.khoa = khoa;

}

**public** **int** getPhong() {

**return** phong;

}

**public** **void** setPhong(**int** phong) {

**this**.phong = phong;

}

**public** Date getNgayVao() {

**return** ngayVao;

}

**public** **boolean** setNgayVao(String ngayVao) {

SimpleDateFormat sdf=**new** SimpleDateFormat("dd/MM/yyyy");

**try** {

**this**.ngayVao=sdf.parse(ngayVao);

**return** **true**;

} **catch** (Exception e) {

System.***err***.println("&Note: Ngày, tháng hoặc năm không hợp lệ!"

+ " \n=> Mời nhập lại: ");

**return** **false**;

}

}

**public** Date getNgayRa() {

**return** ngayRa;

}

**public** **boolean** setNgayRa(String ngayRa) {

SimpleDateFormat sdf=**new** SimpleDateFormat("dd/MM/yyyy");

**try** {

**this**.ngayRa=sdf.parse(ngayRa);

**return** **true**;

}

**catch** (Exception e) {

System.***err***.println("&Note: Ngày, tháng hoặc năm không hợp lệ!"

+ " \n=> Mời nhập lại: ");

**return** **false**;

}

}

**public** **float** getVienPhi() {

**float** vp=SoNgayNamVien(ngayVao, ngayRa)\*200;

**if**(TuoiCuaBN()<5) {

vienPhi=0;

}

**else** {

**if**(bhyt.contentEquals("đúng tuyến")==**true**)

vienPhi= (**float**) (vp-vp\*0.8);

**else** **if**(bhyt.contentEquals("trái tuyến")==**true**)

vienPhi= (**float**) (vp-vp\*0.5);

**else**

vienPhi=vp;

}

**return** vienPhi;

}

**public** **void** showInfo2() {

//super.showInfo();

System.***out***.print("\n-Mã bệnh án: "+ maBA);

System.***out***.print("\n-Tên bệnh án: "+ tenBA);

System.***out***.print("\n-Khoa: "+ khoa);

System.***out***.print("\n-Phòng: "+phong);

SimpleDateFormat sdf=**new** SimpleDateFormat("dd/MM/yyyy");

System.***out***.print("\n-Ngày vào: "+ sdf.format(ngayVao));

System.***out***.print("\n-Ngày ra: "+sdf.format(ngayRa));

System.***out***.print("\n-Viện phí: "+ vienPhi+" ngàn đồng");

}

**public** **long** SoNgayNamVien(Date d1, Date d2) {

// chuyển 2 thời điểm sang mili giây

**long** startValue= d1.getTime();

**long** endValue= d2.getTime();

//tính mili giây hiệu 2 thời điểm

**long** tmp=Math.*abs*(startValue-endValue);

//chia hiệu đó cho 1 ngày=> số ngày chênh lệch

**long** result = tmp/(24\*60\*60\*1000);

**return** result;

}

**public** **long** TuoiCuaBN() {

Calendar cal= Calendar.*getInstance*();

SimpleDateFormat sdf=**new** SimpleDateFormat("dd/MM/yyyy");

String dateNow= sdf.format(cal.getTime());

**long** tuoi = 0;

**try** {

Date timeNow= sdf.parse(dateNow);

**long** startValue= timeNow.getTime();

**long** endValue= ngaySinh.getTime();

**long** tmp=Math.*abs*(startValue-endValue);

**long** result = tmp/(24\*60\*60\*1000);

tuoi=result/365;

} **catch** (ParseException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** tuoi;

}

}

**package** duyen.com.test;

**import** java.text.ParseException;

**import** java.text.SimpleDateFormat;

**import** java.util.ArrayList;

**import** java.util.Date;

**import** java.util.Scanner;

**import** duyen.com.model.BenhAn;

**import** duyen.com.model.BenhNhan;

**public** **class** testBenhAn {

**public** **static** ArrayList<BenhAn> *ds*=**new** ArrayList<BenhAn>();

**static** **void** Menu()

{

System.***out***.println("\n----------------- MENU ----------------");

System.***out***.println("1. Thêm bệnh án");

System.***out***.println("2. Hiển thị thông tin bệnh án");

System.***out***.println("3. Danh sách bệnh nhân dưới 5 tuổi ");

System.***out***.println("4. Danh sách bệnh nhân trên 70 tuổi nằm 'khoa ngoại'");

System.***out***.println("5. Danh sách bệnh nhân có tiền viện phí cao nhất");

System.***out***.println("6. Tổng số bệnh nhân nằm viện từ ngày 20/2/2021 đến 13/4/2021");

System.***out***.println("7. Tổng số bệnh nhân ra viện từ ngày 20/2/2021 đến 13/4/2021");

System.***out***.println("8. tổng số tiền viện phí thu được trong quý 1 năm 2021(từ 1/1/2021-31/3/2021)");

System.***out***.println("9. Danh sách bệnh nhân nam 30 tuổi nằm viện trong quý 1 năm 2021");

System.***out***.println("10.Danh sách bệnh nhân có bảo hiểm y tế và có tiền viện phí>2 triệu");

System.***out***.println("11.Danh sách bệnh nhân không có bảo hiểm y tế có viện phí >10 triệu");

System.***out***.println("12.Danh sách bênh nhân dưới<30 tuổi trái tuyến bảo hiểm");

System.***out***.println("13.Thoát");

}

**public** **static** **void** menu()

{

SimpleDateFormat sdf= **new** SimpleDateFormat("dd/MM/yyyy");

Scanner sc= **new** Scanner(System.***in***);

**int** dem;

**int** chon;

**do** {

*Menu*();

System.***out***.print("\n==>Mời nhập lựa chọn? ");

chon =Integer.*parseInt*(sc.nextLine());

**switch**(chon)

{

**case** 1:

**int** n;

System.***out***.println("Nhập số lượng bệnh án: ");

n=Integer.*parseInt*(sc.nextLine());

**for**(**int** i=0; i<n;i++) {

System.***out***.println("\n"+(i+1));

BenhAn ba=**new** BenhAn();

ba.inputInfo();

*ds*.add(ba);

}

**break**;

**case** 2:

System.***out***.println("----- DANH SÁCH BỆNH NHÂN -----");

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

{

System.***out***.println((i+1));

*ds*.get(i).getVienPhi();

*ds*.get(i).showInfo();

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

}

System.***out***.println("----- DANH SÁCH BỆNH ÁN -----");

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

{

System.***out***.println((i+1));

*ds*.get(i).showInfo2();

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

}

**break**;

**case** 3:

System.***out***.println("-------- DANH SÁCH BỆNH NHÂN DƯỚI 5 TUỔI --------");

dem=0;

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

**if**(*ds*.get(i).TuoiCuaBN() <5 )

{

dem+=1;

System.***out***.print((dem+"."));

*ds*.get(i).showInfo();

}

}

**break**;

**case** 4:

System.***out***.println("-------- DANH SÁCH BỆNH NHÂN TRÊN 70 TUỔI NẰM KHOA NGOẠI --------");

dem=0;

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

**if**(*ds*.get(i).TuoiCuaBN()>70 && *ds*.get(i).getKhoa().compareToIgnoreCase("Ngoại")==0 )

{

dem+=1;

System.***out***.print((dem+"."));

*ds*.get(i).showInfo();

}

}

**break**;

**case** 5:

**float** max=*ds*.get(0).getVienPhi();

System.***out***.println("-------- DANH SÁCH BỆNH NHÂN CÓ TIỀN VIỆN PHÍ CAO NHẤT --------");

**for**(**int** i=1;i<*ds*.size();i++) {

**if**(*ds*.get(i).getVienPhi()>max)

{

max=*ds*.get(i).getVienPhi();

}

}

dem=0;

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

{

**if**(max==*ds*.get(i).getVienPhi())

{

dem+=1;

System.***out***.print(("\n"+dem+"."));

*ds*.get(i).showInfo();

}

}

**break**;

**case** 6:

System.***out***.println("-------- TỔNG SỐ BỆNH NHÂN NẰM VIỆN TỪ 20/02/2021 ĐẾN 13/04/2021 --------");

**int** tongNam=0;

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

{

**try** {

**if**(*ds*.get(i).getNgayVao().compareTo(sdf.parse("20/02/2021"))>=0 && *ds*.get(i).getNgayVao().compareTo(sdf.parse("13/04/2021"))<=0 )

{

tongNam+=1;

}

} **catch** (ParseException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

System.***out***.println(">>>>>>>> Kết quả: "+tongNam+" bệnh nhân");

**break**;

**case** 7:

System.***out***.println("-------- TỔNG SỐ BỆNH NHÂN RA VIỆN TỪ 20/02/2021 ĐẾN 13/04/2021 --------");

**int** tongRa=0;

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

{

**try** {

**if**(*ds*.get(i).getNgayRa().compareTo(sdf.parse("20/02/2021"))>=0 && *ds*.get(i).getNgayRa().compareTo(sdf.parse("13/04/2021"))<=0 )

{

tongRa+=1;

}

} **catch** (ParseException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

System.***out***.println(">>>>>> Kết quả: "+tongRa+" bệnh nhân");

**break**;

**case** 8:

System.***out***.println("-------- TỔNG TIỀN VIỆN PHÍ THU ĐƯỢC TRONG QUÝ 1 NĂM 2021 --------");

**float** tongTienVP=0;

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

{

**try** {

**if**(*ds*.get(i).getNgayRa().compareTo(sdf.parse("01/01/2021"))>=0 && *ds*.get(i).getNgayRa().compareTo(sdf.parse("31/03/2021"))<=0 )

{

tongTienVP+=*ds*.get(i).getVienPhi();

}

} **catch** (ParseException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

System.***out***.println("Tổng tiền: "+ tongTienVP+" ngàn đồng");

**break**;

**case** 9:

System.***out***.println("-------- DANH SÁCH BỆNH NHÂN NAM 30 TUỔI NẰM VIỆN TRONG QUÝ 1 NĂM 2021 --------");

dem=0;

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

{

**try** {

**if**(*ds*.get(i).getGioiTinh().contains("nam")==**true** && *ds*.get(i).TuoiCuaBN()==30

&& *ds*.get(i).getNgayVao().compareTo(sdf.parse("01/01/2021"))>=0 && *ds*.get(i).getNgayVao().compareTo(sdf.parse("31/03/2021"))<=0 )

{

dem+=1;

System.***out***.print(("\n"+dem+"."));

*ds*.get(i).showInfo();

}

} **catch** (ParseException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**break**;

**case** 10:

System.***out***.println("-------- DANH SÁCH BỆNH NHÂN CÓ BHYT VÀ TIỀN VIỆN PHÍ TRÊN 2 TRIỆU --------");

dem=0;

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

{

**if**(*ds*.get(i).getBhyt().contains("không tham gia")==**false** && *ds*.get(i).getVienPhi()>2000)

{

dem+=1;

System.***out***.print(("\n"+dem+"."));

*ds*.get(i).showInfo();

}

}

**break**;

**case** 11:

System.***out***.println("-------- DANH SÁCH BỆNH NHÂN KHÔNG CÓ BHYT VÀ TIỀN VIỆN PHÍ TRÊN 10 TRIỆU --------");

dem=0;

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

{

**if**(*ds*.get(i).getBhyt().contains("không tham gia")==**true** && *ds*.get(i).getVienPhi()>10000)

{

dem+=1;

System.***out***.print(("\n"+dem+"."));

*ds*.get(i).showInfo();

}

}

**break**;

**case** 12:

System.***out***.println("-------- DANH SÁCH BỆNH NHÂN DƯỚI 20 TUỔI TRÁI TUYẾN BẢO HIỂM --------");

dem=0;

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

{

**if**(*ds*.get(i).getBhyt().contains("trái tuyến")==**true** && *ds*.get(i).TuoiCuaBN()<20 )

{

dem+=1;

System.***out***.print(("\n"+dem+"."));

*ds*.get(i).showInfo();

}

}

**break**;

**case** 13:

System.***out***.println(">>>>>>>> SEE YOU <<<<<<<<");

**break**;

**default**:

System.***err***.println("Nhập lụi rồi nha");

}

}**while**(chon!=13);

}

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

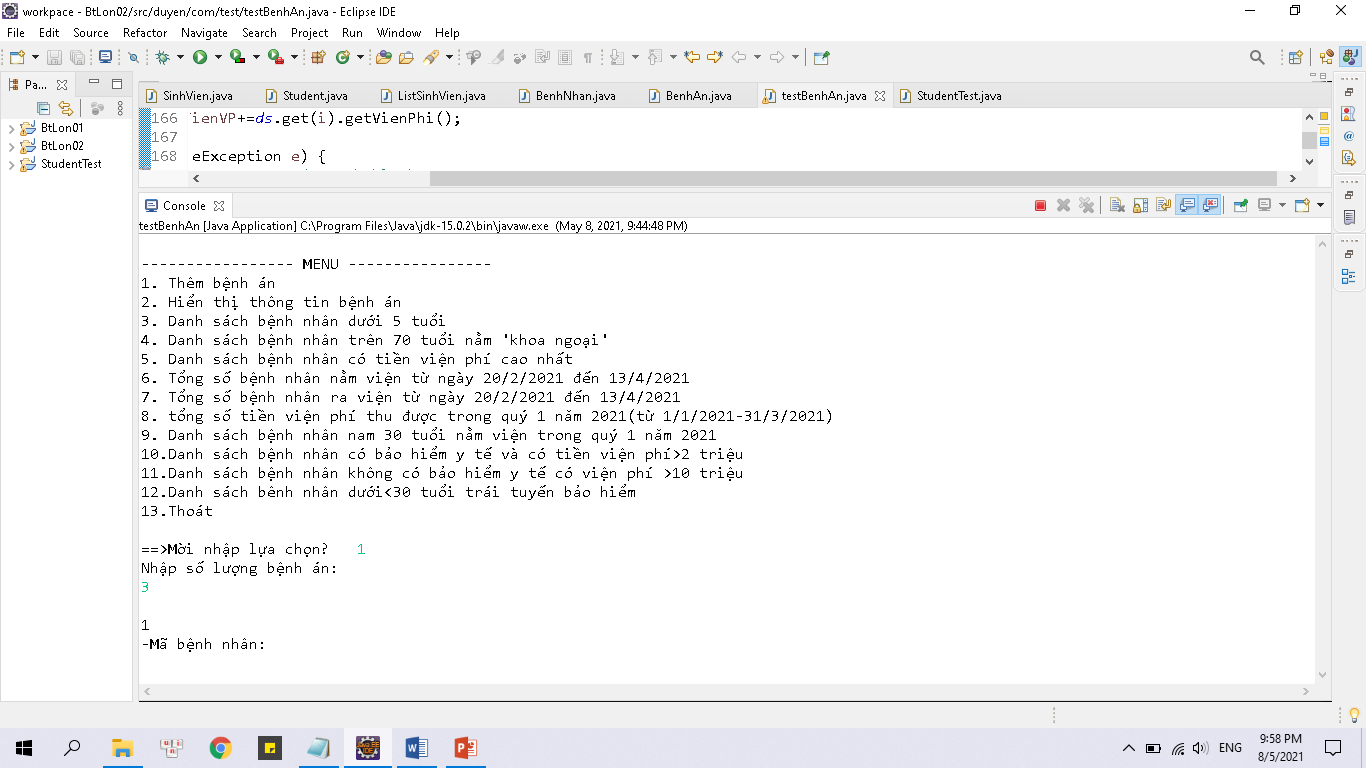
*menu*();

}

}

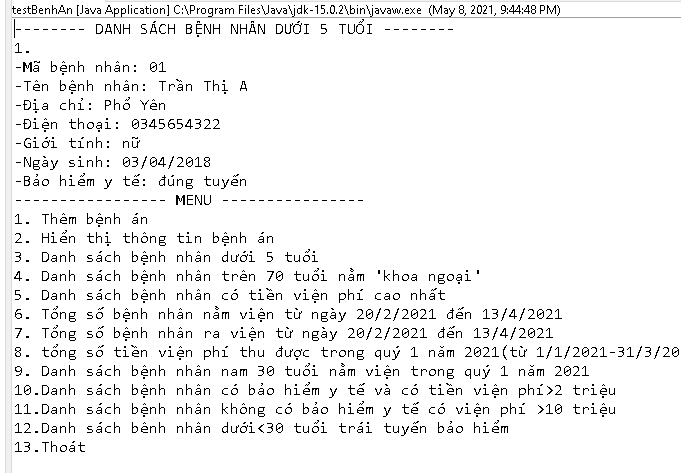
c. Kết quả:

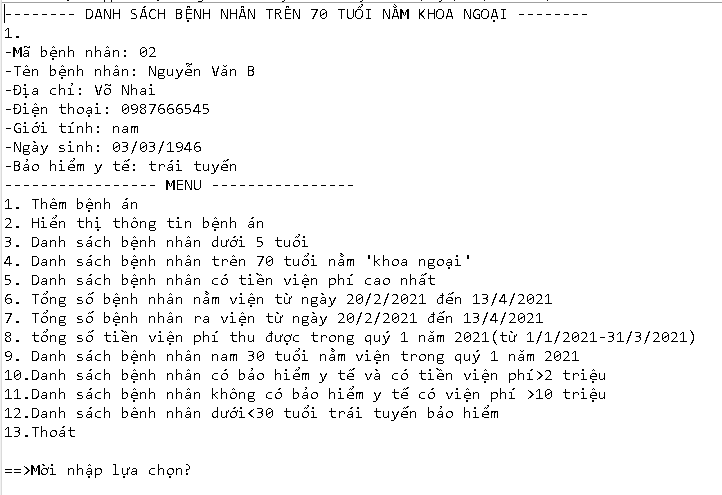
**-**Nhập n bệnh án:

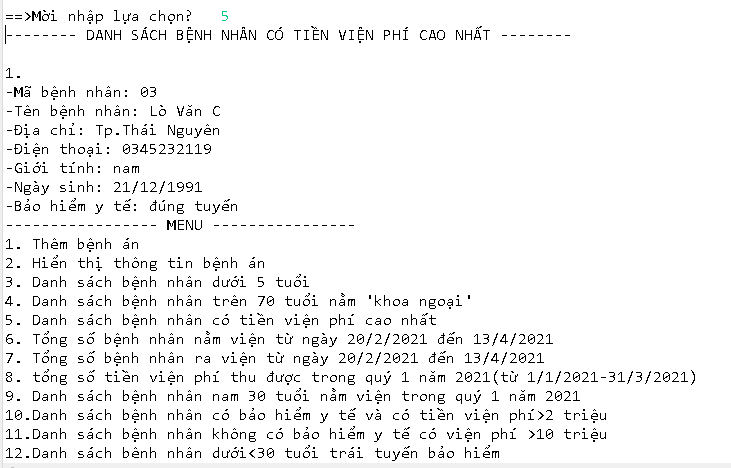
 Hiển thị được chia ra danh sách bệnh nhân và danh sách bệnh án rõ ràng:

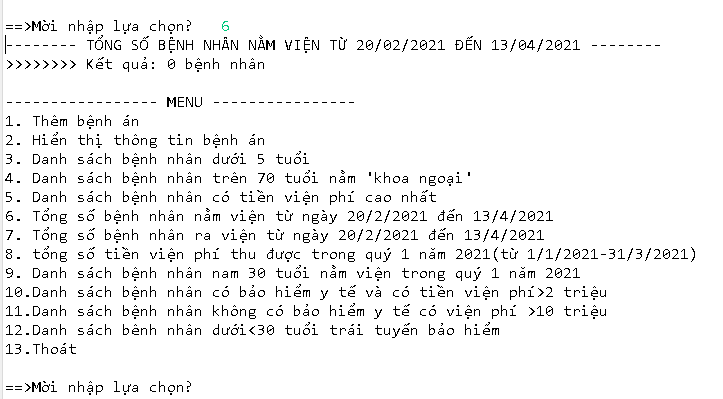
|  |  |
| --- | --- |
| --- |  |

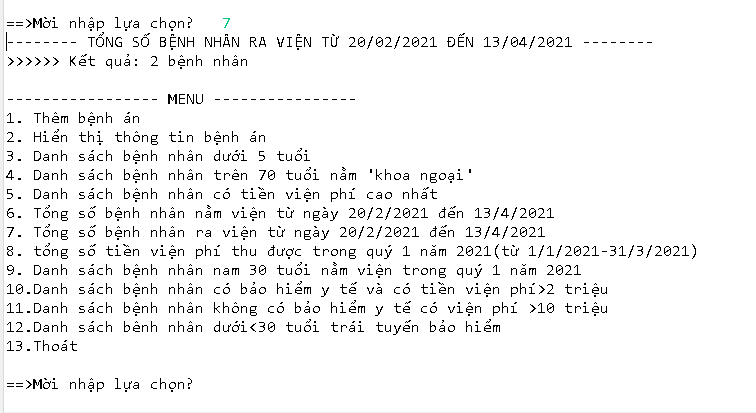
-Sau đây lần lượt giải đap các yêu cần đề bài:

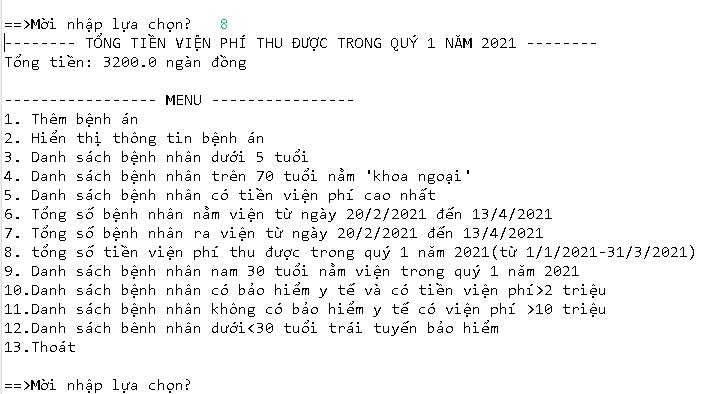


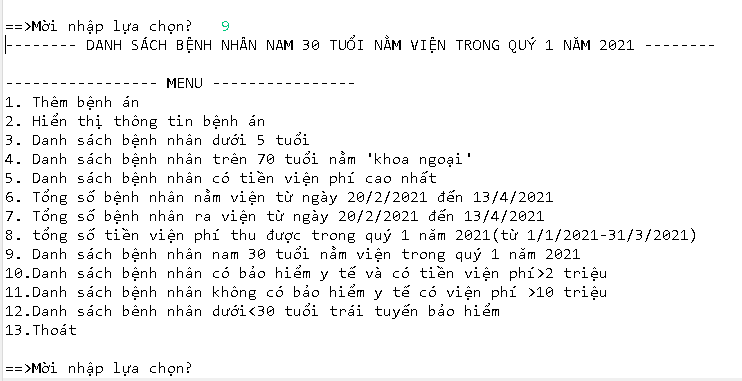


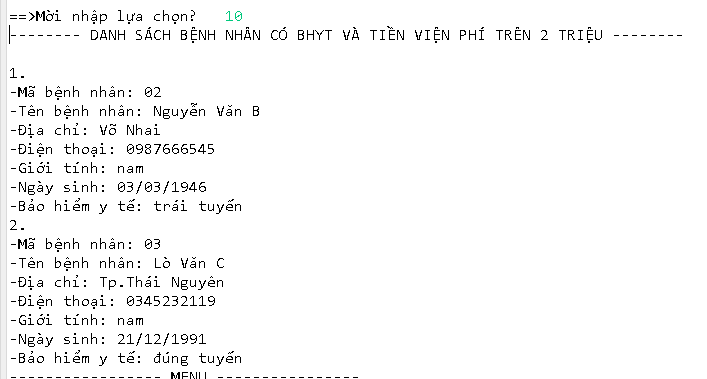


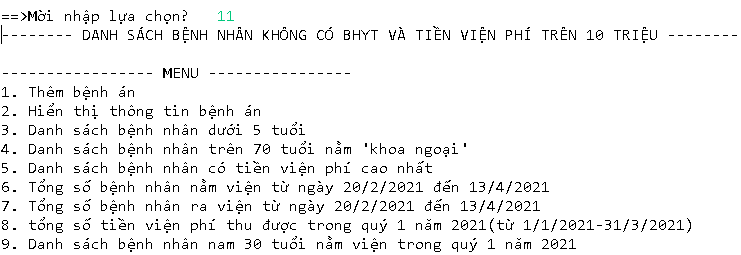


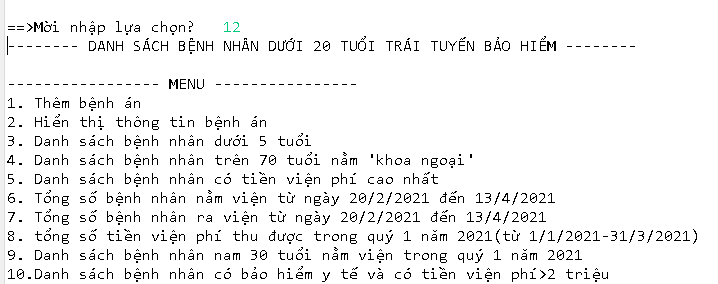












**----------------------------End ----------------------------**