Name of the Course : Complete Java SE8 Developer Bootcamp

Level : Difficult

Tool Stack : Java8 and Junit4

Problem Statement : Provide a code solution for a menu based application in doctors clinic for enrolment of patients.

Description : ***All-Cure Doctors’ Clinic*** creates daily patients list in a Map at the time of enlistment of a patient. The patient’s details like Patient Name, Doctor’s name (whom the patient wishes to visit), patient address will accepted in a comma separated format. A serial number for Patient (which starts with 1 everyday) will be auto generated. You need to develop a Java menu based application which allows 1. Enrol a new patient and generate a serial number, 2. Display all patients for the day, 3. Display doctor wise patient list (i.e. accept doctor name then display patients visited him/her).

1. Create class Patient with attributes(all are private):

String serialNo,

String patientName,

String address,

String doctorName,

Write constructors and getter and setter methods of the class. Override the **toString()** method in the format specified in the output.   
**String.format("%-5s %-20s %-20s %-20s ",....);**

1. Create class PatientRecord with following members:
2. private static Map<Integer,Patient> patientMap.
3. public static int generateSerialNumber().
4. public static List<Patient> getAllPatients().
5. public static void patientEntry(Patient patient);

1. Create class Main with methods
2. private static List<Patient> filterPatient (String doctorName): it will filter the patients of a particular doctor.
3. public static void main(String [] arg): It provides a menu to user to work on the application which is as follows:
4. **Patient Entry.**
5. **Display All Patient for the day.**
6. **Display Patient Doctor-wise.**
7. **Exit**

**Enter Choice(1-4).**

Code:

**public** **class** Patient {

**private** Integer serialNo;

**private** String patientName;

**private** String address;

**private** String doctorName;

**public** Patient() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** Patient(Integer serialNo, String patientName, String address, String doctorName) {

**super**();

**this**.serialNo = serialNo;

**this**.patientName = patientName;

**this**.address = address;

**this**.doctorName = doctorName;

}

**public** Integer getSerialNo() {

**return** serialNo;

}

**public** **void** setSerialNo(Integer serialNo) {

**this**.serialNo = serialNo;

}

**public** String getPatientName() {

**return** patientName;

}

**public** **void** setPatientName(String patientName) {

**this**.patientName = patientName;

}

**public** String getAddress() {

**return** address;

}

**public** **void** setAddress(String address) {

**this**.address = address;

}

**public** String getDoctorName() {

**return** doctorName;

}

**public** **void** setDoctorName(String doctorName) {

**this**.doctorName = doctorName;

}

@Override

**public** String toString() {

String output=String.*format*("%-5s %-20s %-20s %-20s",serialNo,patientName,address,doctorName);

**return** output;

}

}

**import** java.util.Map;

**import** java.util.Set;

**import** java.util.TreeSet;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** java.util.List;

**public** **class** PatientRecord {

**private** **static** Map<Integer,Patient> *patientMap*=**new** HashMap<>();

**public** **static** **void** patientEntry(Patient patient){

*patientMap*.put(patient.getSerialNo(),patient);

}

**public** **static** **int** generateSerialNumber(){

**int** value=1;

**if**(*patientMap*.size()!=0)

{

Set<Integer> set=*patientMap*.keySet();

TreeSet<Integer> tSet=**new** TreeSet<>(set);

value=tSet.last()+1;

}

**return** value;

}

**public** **static** List<Patient> getAllPatients(){

List<Patient> patientList=**new** ArrayList<Patient>();

Set<Integer> set=*patientMap*.keySet();

**for**(Integer serialNo:set)

patientList.add(*patientMap*.get(serialNo));

**return** patientList;

}

}

**import** java.util.ArrayList;

**import** java.util.List;

**import** java.util.Scanner;

**public** **class** Main {

**private** **static** List<Patient> filterPatient (String doctorName)

{

List<Patient> newList=**new** ArrayList<Patient>();

List<Patient> patientList=PatientRecord.*getAllPatients*();

**for**(Patient patient:patientList)

{

**if**(doctorName.equalsIgnoreCase(patient.getDoctorName()))

newList.add(patient);

}

**return** newList;

}

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

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

String output=String.*format*("%-5s %-20s %-20s %-20s","Serial No","Patient Name","Address","Doctor");

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

{

System.***out***.println("1.Patient Entry.");

System.***out***.println("2.Display All Patient for the day.");

System.***out***.println("3.Display Patient Doctor-wise.");

System.***out***.println("4.Exit");

System.***out***.println("Enter Choice(1-4).");

String option=scanner.nextLine();

**switch**(option)

{

**case** "1": System.***out***.println("Enter patient details in comma separate format:");

String details=scanner.nextLine();

String arr[]=details.split(",");

**int** serialNo=PatientRecord.*generateSerialNumber*();

Patient patient=**new** Patient(serialNo,arr[0],arr[1],arr[2]);

System.***out***.println(output);

System.***out***.println(patient);

PatientRecord.*patientEntry*(patient);

**break**;

**case** "2" : List<Patient> patientList=PatientRecord.*getAllPatients*();

System.***out***.println(output);

**for**(Patient patient2:patientList)

System.***out***.println(patient2);

**break**;

**case** "3" : System.***out***.println("Enter Doctor's Name:");

String doctorName=scanner.nextLine();

List<Patient> doctorPatientList=*filterPatient*(doctorName);

**if**(doctorPatientList.size()==0)

System.***out***.println("Wrong Doctor Name ");

**else**

{

System.***out***.println(output);

**for**(Patient patient3:doctorPatientList)

System.***out***.println(patient3);

}

**break**;

**case** "4" : System.*exit*(0);

}// end of switch

}// end of loop

}// end of main

}// end of class

Junit Testing

**import** **static** org.junit.Assert.\*;

**import** org.junit.Test;

**public** **class** PatientRecordTest {

@Test

**public** **void** testGenerateSerialNumber() {

Patient patient1=**new** Patient(10,"John","Park Road","Dr.Eric");

Patient patient2=**new** Patient(15,"Johny","Park Road","Dr.Eric");

PatientRecord.*patientEntry*(patient2);

PatientRecord.*patientEntry*(patient1);

*assertEquals*(16,PatientRecord.*generateSerialNumber*());

}

}

**import** **static** org.junit.Assert.\*;

**import** org.junit.Test;

**public** **class** MainTest {

@Test

**public** **void** testFilterPatient() {

Patient patient1=**new** Patient(10,"John","Park Road","Dr.Steve");

Patient patient2=**new** Patient(11,"Mike","Lake Row","Dr.Eric");

Patient patient3=**new** Patient(14,"Liz","Lake Road","Dr.Eric");

Patient patient4=**new** Patient(15,"Johny","Park Row","Dr.Steve");

PatientRecord.*patientEntry*(patient1);

PatientRecord.*patientEntry*(patient2);

PatientRecord.*patientEntry*(patient3);

PatientRecord.*patientEntry*(patient4);

*assertEquals*(**new** Integer(11),(Integer)Main.*filterPatient*("Dr.Eric").get(0).getSerialNo());

*assertEquals*(**new** Integer(10),(Integer)Main.*filterPatient*("Dr.Steve").get(0).getSerialNo());

}

}

Test Data1

1.Patient Entry.

2.Display All Patient for the day.

3.Display Patient Doctor-wise.

4.Exit

Enter Choice(1-4).

1

Enter patient details in comma separate format:

Jim,16Park Street,Dr.Gomes

Serial No Patient Name Address Doctor

1 Jim 16Park Street Dr.Gomes

1.Patient Entry.

2.Display All Patient for the day.

3.Display Patient Doctor-wise.

4.Exit

Enter Choice(1-4).

1

Enter patient details in comma separate format:

John,12Brook Lane,Dr.Currie

Serial No Patient Name Address Doctor

2 John 12Brook Lane Dr.Currie

1.Patient Entry.

2.Display All Patient for the day.

3.Display Patient Doctor-wise.

4.Exit

Enter Choice(1-4).

1

Enter patient details in comma separate format:

Steve,20York Road,Dr.Gomes

Serial No Patient Name Address Doctor

3 Steve 20York Road Dr.Gomes

1.Patient Entry.

2.Display All Patient for the day.

3.Display Patient Doctor-wise.

4.Exit

Enter Choice(1-4).

1

Enter patient details in comma separate format:

Rogers,11Fleet Row,Dr.Currie

Serial No Patient Name Address Doctor

4 Rogers 11Fleet Row Dr.Currie

1.Patient Entry.

2.Display All Patient for the day.

3.Display Patient Doctor-wise.

4.Exit

Enter Choice(1-4).

2

Serial No Patient Name Address Doctor

1 Jim 16Park Street Dr.Gomes

2 John 12Brook Lane Dr.Currie

3 Steve 20York Road Dr.Gomes

4 Rogers 11Fleet Row Dr.Currie

1.Patient Entry.

2.Display All Patient for the day.

3.Display Patient Doctor-wise.

4.Exit

Enter Choice(1-4).

3

Enter Doctor's Name:

Dr.John

Wrong Doctor Name

1.Patient Entry.

2.Display All Patient for the day.

3.Display Patient Doctor-wise.

4.Exit

Enter Choice(1-4).

3

Enter Doctor's Name:

Dr.Gomes

Serial No Patient Name Address Doctor

1 Jim 16Park Street Dr.Gomes

3 Steve 20York Road Dr.Gomes

1.Patient Entry.

2.Display All Patient for the day.

3.Display Patient Doctor-wise.

4.Exit

Enter Choice(1-4).

4

Learning outcome: Participant could able to learn how to create Menu based application using various collection classes with programming logic .