**JAVA MINI PROJECT**

Hospital management system

Name:G.RITHIKA

Roll no:231401084

Sub:oops with java

Code:

import java.util.ArrayList;

import java.util.Scanner;

class Patient {

private String name;

private int age;

private String gender;

private String illness;

public Patient(String name, int age, String gender, String illness) {

this.name = name;

this.age = age;

this.gender = gender;

this.illness = illness;

}

public String getName() {

return name;

}

public int getAge() {

return age;

}

public String getGender() {

return gender;

}

public String getIllness() {

return illness;

}

@Override

public String toString() {

return "Patient Name: " + name + ", Age: " + age + ", Gender: " + gender + ", Illness: " + illness;

}

}

class Doctor {

private String name;

private String specialty;

private int experienceYears;

public Doctor(String name, String specialty, int experienceYears) {

this.name = name;

this.specialty = specialty;

this.experienceYears = experienceYears;

}

public String getName() {

return name;

}

public String getSpecialty() {

return specialty;

}

public int getExperienceYears() {

return experienceYears;

}

@Override

public String toString() {

return "Doctor Name: " + name + ", Specialty: " + specialty + ", Experience: " + experienceYears + " years";

}

}

class Appointment {

private Patient patient;

private Doctor doctor;

private String appointmentTime;

public Appointment(Patient patient, Doctor doctor, String appointmentTime) {

this.patient = patient;

this.doctor = doctor;

this.appointmentTime = appointmentTime;

}

public Patient getPatient() {

return patient;

}

public Doctor getDoctor() {

return doctor;

}

public String getAppointmentTime() {

return appointmentTime;

}

@Override

public String toString() {

return "Appointment for Patient: " + patient.getName() + " with Doctor: " + doctor.getName() + " at " + appointmentTime;

}

}

public class HospitalManagementSystem {

private static ArrayList<Patient> patients = new ArrayList<>();

private static ArrayList<Doctor> doctors = new ArrayList<>();

private static ArrayList<Appointment> appointments = new ArrayList<>();

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int choice;

// Predefined doctors for simplicity

doctors.add(new Doctor("Dr. Smith", "Cardiologist", 10));

doctors.add(new Doctor("Dr. Johnson", "Neurologist", 8));

doctors.add(new Doctor("Dr. Lee", "Orthopedic", 12));

do {

System.out.println("\n------ Hospital Management System ------");

System.out.println("1. Add Patient");

System.out.println("2. View Patients");

System.out.println("3. Add Doctor");

System.out.println("4. View Doctors");

System.out.println("5. Schedule Appointment");

System.out.println("6. View Appointments");

System.out.println("7. Exit");

System.out.print("Enter your choice: ");

choice = scanner.nextInt();

scanner.nextLine(); // Consume newline

switch (choice) {

case 1:

addPatient(scanner);

break;

case 2:

viewPatients();

break;

case 3:

addDoctor(scanner);

break;

case 4:

viewDoctors();

break;

case 5:

scheduleAppointment(scanner);

break;

case 6:

viewAppointments();

break;

case 7:

System.out.println("Exiting the system...");

break;

default:

System.out.println("Invalid choice, please try again.");

}

} while (choice != 7);

}

private static void addPatient(Scanner scanner) {

System.out.print("Enter Patient's Name: ");

String name = scanner.nextLine();

System.out.print("Enter Patient's Age: ");

int age = scanner.nextInt();

scanner.nextLine(); // Consume newline

System.out.print("Enter Patient's Gender: ");

String gender = scanner.nextLine();

System.out.print("Enter Patient's Illness: ");

String illness = scanner.nextLine();

patients.add(new Patient(name, age, gender, illness));

System.out.println("Patient added successfully!");

}

private static void viewPatients() {

if (patients.isEmpty()) {

System.out.println("No patients available.");

} else {

for (Patient patient : patients) {

System.out.println(patient);

}

}

}

private static void addDoctor(Scanner scanner) {

System.out.print("Enter Doctor's Name: ");

String name = scanner.nextLine();

System.out.print("Enter Doctor's Specialty: ");

String specialty = scanner.nextLine();

System.out.print("Enter Doctor's Years of Experience: ");

int experience = scanner.nextInt();

scanner.nextLine(); // Consume newline

doctors.add(new Doctor(name, specialty, experience));

System.out.println("Doctor added successfully!");

}

private static void viewDoctors() {

if (doctors.isEmpty()) {

System.out.println("No doctors available.");

} else {

for (Doctor doctor : doctors) {

System.out.println(doctor);

}

}

}

private static void scheduleAppointment(Scanner scanner) {

System.out.println("Available Doctors:");

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

System.out.println((i + 1) + ". " + doctors.get(i));

}

System.out.print("Select a doctor by number: ");

int doctorIndex = scanner.nextInt() - 1;

scanner.nextLine(); // Consume newline

if (doctorIndex < 0 || doctorIndex >= doctors.size()) {

System.out.println("Invalid doctor selection.");

return;

}

System.out.println("Available Patients:");

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

System.out.println((i + 1) + ". " + patients.get(i));

}

System.out.print("Select a patient by number: ");

int patientIndex = scanner.nextInt() - 1;

scanner.nextLine(); // Consume newline

if (patientIndex < 0 || patientIndex >= patients.size()) {

System.out.println("Invalid patient selection.");

return;

}

System.out.print("Enter Appointment Time (e.g., 10:00 AM): ");

String appointmentTime = scanner.nextLine();

Appointment appointment = new Appointment(patients.get(patientIndex), doctors.get(doctorIndex), appointmentTime);

appointments.add(appointment);

System.out.println("Appointment scheduled successfully!");

}

private static void viewAppointments() {

if (appointments.isEmpty()) {

System.out.println("No appointments scheduled.");

} else {

for (Appointment appointment : appointments) {

System.out.println(appointment);

}

}

}

}