**HOTEL BOOKING SYSTEM**

**CODE:**

**// HotelBookingSystem.java**

**import java.util.\*;**

**class Guest {**

**private String name;**

**private String cnic;**

**private String contact;**

**public Guest(String name, String cnic, String contact) {**

**this.name = name;**

**this.cnic = cnic;**

**this.contact = contact;**

**}**

**public String getName() { return name; }**

**public String getCnic() { return cnic; }**

**public String getContact() { return contact; }**

**public String toString() {**

**return name + ", " + cnic + ", " + contact;**

**}**

**}**

**abstract class Room {**

**protected int roomNumber;**

**protected boolean isAvailable = true;**

**public Room(int roomNumber) {**

**this.roomNumber = roomNumber;**

**}**

**public boolean isAvailable() { return isAvailable; }**

**public void setAvailable(boolean available) { this.isAvailable = available; }**

**public int getRoomNumber() { return roomNumber; }**

**public abstract double calculateCharges(int nights);**

**public abstract String getType();**

**public abstract String getFeatures();**

**}**

**class EconomyRoom extends Room {**

**public EconomyRoom(int roomNumber) { super(roomNumber); }**

**@Override**

**public double calculateCharges(int nights) { return nights \* 3000; }**

**public String getType() { return "Economy"; }**

**public String getFeatures() { return "Single bed, Wi-Fi, TV"; }**

**}**

**class DeluxeRoom extends Room {**

**public DeluxeRoom(int roomNumber) { super(roomNumber); }**

**@Override**

**public double calculateCharges(int nights) { return nights \* 6000; }**

**public String getType() { return "Deluxe"; }**

**public String getFeatures() { return "Double bed, AC, Mini-fridge, Wi-Fi, TV"; }**

**}**

**class SuiteRoom extends Room {**

**public SuiteRoom(int roomNumber) { super(roomNumber); }**

**@Override**

**public double calculateCharges(int nights) { return nights \* 10000; }**

**public String getType() { return "Suite"; }**

**public String getFeatures() { return "King-size bed, Living area, Kitchen, Premium Wi-Fi, TV"; }**

**}**

**class Booking {**

**private String bookingID;**

**private Guest guest;**

**private Room room;**

**private int nights;**

**private double advancePaid;**

**public Booking(String bookingID, Guest guest, Room room, int nights, double advancePaid) {**

**this.bookingID = bookingID;**

**this.guest = guest;**

**this.room = room;**

**this.nights = nights;**

**this.advancePaid = advancePaid;**

**}**

**public String getBookingID() { return bookingID; }**

**public Guest getGuest() { return guest; }**

**public Room getRoom() { return room; }**

**public int getNights() { return nights; }**

**public double getAdvancePaid() { return advancePaid; }**

**public double getTotalCharges() { return room.calculateCharges(nights); }**

**public String toString() {**

**return bookingID + ", " + guest.toString() + ", Room " + room.getRoomNumber() + " (" + room.getType() + ")" + ", Nights: " + nights;**

**}**

**}**

**public class HotelBookingSystem {**

**private static ArrayList<Room> rooms = new ArrayList<>();**

**private static ArrayList<Booking> bookings = new ArrayList<>();**

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

**private static int bookingCounter = 1;**

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

**initializeRooms();**

**int choice;**

**do {**

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

**System.out.println("1. Book a Room");**

**System.out.println("2. View Available Rooms");**

**System.out.println("3. Check-out Guest");**

**System.out.println("4. View All Bookings");**

**System.out.println("5. Exit");**

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

**while (!input.hasNextInt()) {**

**System.out.println("Invalid input. Please enter a number between 1-5.");**

**input.next();**

**}**

**choice = input.nextInt();**

**input.nextLine();**

**switch (choice) {**

**case 1: bookRoom(); break;**

**case 2: showAvailableRooms(); break;**

**case 3: checkout(); break;**

**case 4: showBookings(); break;**

**case 5: System.out.println("Thank you for using the system!"); break;**

**default: System.out.println("Invalid choice.");**

**}**

**} while (choice != 5);**

**}**

**private static void initializeRooms() {**

**for (int i = 101; i <= 110; i++) rooms.add(new EconomyRoom(i));**

**for (int i = 201; i <= 210; i++) rooms.add(new DeluxeRoom(i));**

**for (int i = 301; i <= 310; i++) rooms.add(new SuiteRoom(i));**

**}**

**private static void displayCategories() {**

**System.out.println("\nRoom Categories:");**

**System.out.println("1. Economy - Single bed, Wi-Fi, TV");**

**System.out.println("2. Deluxe - Double bed, AC, Mini-fridge, Wi-Fi, TV");**

**System.out.println("3. Suite - King-size bed, Living area, Kitchen, Premium Wi-Fi, TV");**

**}**

**private static void showAvailableRooms(String type) {**

**System.out.println("\nAvailable " + type + " Rooms:");**

**for (Room room : rooms) {**

**if (room.isAvailable() && room.getType().equals(type)) {**

**System.out.println("Room " + room.getRoomNumber());**

**}**

**}**

**}**

**private static void showAvailableRooms() {**

**for (Room room : rooms) {**

**if (room.isAvailable()) {**

**System.out.println("Room " + room.getRoomNumber() + " (" + room.getType() + ")");**

**}**

**}**

**}**

**private static void bookRoom() {**

**System.out.print("Enter Guest Name: ");**

**String name = input.nextLine();**

**System.out.print("Enter CNIC: ");**

**String cnic = input.nextLine();**

**System.out.print("Enter Contact: ");**

**String contact = input.nextLine();**

**Guest guest = new Guest(name, cnic, contact);**

**displayCategories();**

**System.out.print("Select Room Category (1-3): ");**

**int cat = input.nextInt(); input.nextLine();**

**String type = "";**

**if (cat == 1) type = "Economy";**

**else if (cat == 2) type = "Deluxe";**

**else if (cat == 3) type = "Suite";**

**else {**

**System.out.println("Invalid category.");**

**return;**

**}**

**showAvailableRooms(type);**

**System.out.print("Enter Room Number to Book: ");**

**int roomNum = input.nextInt(); input.nextLine();**

**Room selectedRoom = null;**

**for (Room room : rooms) {**

**if (room.getRoomNumber() == roomNum && room.isAvailable() && room.getType().equals(type)) {**

**selectedRoom = room;**

**break;**

**}**

**}**

**if (selectedRoom != null) {**

**System.out.print("Enter Number of Nights: ");**

**int nights = input.nextInt(); input.nextLine();**

**double total = selectedRoom.calculateCharges(nights);**

**double advance = total / 2;**

**selectedRoom.setAvailable(false);**

**String bookingID = "BK" + (bookingCounter++);**

**Booking booking = new Booking(bookingID, guest, selectedRoom, nights, advance);**

**bookings.add(booking);**

**System.out.println("\nRoom Booked Successfully!");**

**System.out.println("Booking ID: " + bookingID);**

**System.out.println("Advance Paid: Rs. " + advance);**

**System.out.println("Total Bill: Rs. " + total);**

**System.out.println("Room Card Issued for Room No: " + selectedRoom.getRoomNumber());**

**} else {**

**System.out.println("Room not available or invalid room number.");**

**}**

**}**

**private static void checkout() {**

**System.out.print("Enter Booking ID: ");**

**String id = input.nextLine();**

**Booking found = null;**

**for (Booking b : bookings) {**

**if (b.getBookingID().equals(id)) {**

**found = b;**

**break;**

**}**

**}**

**if (found != null) {**

**Room room = found.getRoom();**

**double total = room.calculateCharges(found.getNights());**

**double due = total - found.getAdvancePaid();**

**room.setAvailable(true);**

**System.out.println("\nCheckout Complete for " + found.getGuest().getName());**

**System.out.println("Total Bill: Rs. " + total);**

**System.out.println("Advance Paid: Rs. " + found.getAdvancePaid());**

**System.out.println("Remaining Due: Rs. " + due);**

**System.out.println("Room Cleared. You may now checkout.\n");**

**} else {**

**System.out.println("Booking ID not found.");**

**}**

**}**

**private static void showBookings() {**

**System.out.println("\nAll Bookings:");**

**for (Booking b : bookings) {**

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

**}**

**}**

**}**