CLASS ROOM

class Room {

int roomNumber;

String category;

boolean isAvailable;

double price;

Room(int roomNumber, String category, double price) {

this.roomNumber = roomNumber;

this.category = category;

this.price = price;

this.isAvailable = true;

}

void bookRoom() {

isAvailable = false;

}

void releaseRoom() {

isAvailable = true;

}

void displayRoom() {

System.out.println("Room " + roomNumber + " (" + category + ") - " +

(isAvailable ? "Available" : "Booked") + " - Price: " + price);

}

}

CLASS CUSTOMER  
  
class Customer {

String name;

String phone;

Customer(String name, String phone) {

this.name = name;

this.phone = phone;

}

void displayCustomer() {

System.out.println("Customer: " + name + " | Phone: " + phone);

}

}  
  
CLASS RESERVATION  
  
class Reservation {

Room room;

Customer customer;

Reservation(Room room, Customer customer) {

this.room = room;

this.customer = customer;

room.bookRoom();

}

void displayReservation() {

System.out.println("Reservation Details:");

customer.displayCustomer();

System.out.println("Room Booked: " + room.roomNumber + " | " + room.category);

}

}

CLASS PAYMENT  
  
  
class Payment {

double amount;

Payment(double amount) {

this.amount = amount;

}

void processPayment() {

System.out.println("Payment of " + amount + " processed successfully.");

}

}  
  
  
CLASS HOTEL  
  
  
class Hotel {

Room[] rooms;

Reservation[] reservations;

int totalReservations;

Hotel() {

rooms = new Room[]{

new Room(101, "Single", 5000),

new Room(102, "Double", 8000),

new Room(103, "Suite", 12000)

};

reservations = new Reservation[10];

totalReservations = 0;

}

void displayAvailableRooms() {

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

for (Room room : rooms) {

if (room.isAvailable) {

room.displayRoom();

}

}

}

void makeReservation() {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter Customer Name: ");

String name = scanner.nextLine();

System.out.print("Enter Phone Number: ");

String phone = scanner.nextLine();

displayAvailableRooms();

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

int roomNumber = scanner.nextInt();

for (Room room : rooms) {

if (room.roomNumber == roomNumber && room.isAvailable) {

Customer customer = new Customer(name, phone);

reservations[totalReservations] = new Reservation(room, customer);

totalReservations++;

Payment payment = new Payment(room.price);

payment.processPayment();

System.out.println("Room booked successfully!");

return;

}

}

System.out.println("Invalid selection or room not available.");

}

void viewReservations() {

System.out.println("All Reservations:");

for (int i = 0; i < totalReservations; i++) {

reservations[i].displayReservation();

}

}

}  
  
  
MAIN CLASS  
  
class Main {

public static void main(String[] args) {

Hotel hotel = new Hotel();

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.println("\nHotel Reservation System:");

System.out.println("1. View Available Rooms");

System.out.println("2. Make a Reservation");

System.out.println("3. View Reservations");

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

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

int choice = scanner.nextInt();

switch (choice) {

case 1:

hotel.displayAvailableRooms();

break;

case 2:

hotel.makeReservation();

break;

case 3:

hotel.viewReservations();

break;

case 4:

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

return;

default:

System.out.println("Invalid choice. Try again.");

}

}

}

}  
  
  
OUTPUT  
  
  
