Write a Java program to create a class with methods to search for flights and hotels, and to book and cancel reservations.

Sample Solution:

Java Code:

```java
// TravelApp.java
// Import the ArrayList class
import java.util.ArrayList;
// Import the Random class
import java.util.Random;

// Define the TravelApp class
public class TravelApp {
  // Declare an ArrayList to store flights
  private ArrayList flights;
  // Declare an ArrayList to store hotels
  private ArrayList hotels;

  // Constructor to initialize the ArrayLists
  public TravelApp() {
    // Initialize the flights ArrayList
    this.flights = new ArrayList();
    // Initialize the hotels ArrayList
    this.hotels = new ArrayList();
  }

  // Method to search for flights
  public void searchFlights(String origin, String destination, String date
    // Print the search details for flights
    System.out.println("Searching for flights from " + origin + " to " + de
  }

  // Method to search for hotels
  public void searchHotels(String location, String checkIn, String checkOu
    // Print the search details for hotels
    System.out.println("Searching for hotels in " + location + " from " +
  }

  // Method to book a flight
  public void bookFlight(int flightNumber, String passengerName, String ori
    // Create a new Flight object with the provided details
    Flight flight = new Flight(flightNumber, passengerName, origin, destina
    // Generate a confirmation number
    int confirmationNumber = generateConfirmationNumber();
    // Set the confirmation number for the flight
    flight.setConfirmationNumber(confirmationNumber);
    // Add the flight to the flights ArrayList
    this.flights.add(flight);
    // Print the confirmation number for the booked flight
    System.out.println("Flight booked! Confirmation number: " + confirmatio
  }

  // Method to book a hotel
  public void bookHotel(int hotelId, String guestName, String location, Str
    // Create a new Hotel object with the provided details
    Hotel hotel = new Hotel(hotelId, guestName, location, checkIn, checkOut
    // Generate a confirmation number
    int confirmationNumber = generateConfirmationNumber();
    // Set the confirmation number for the hotel
    hotel.setConfirmationNumber(confirmationNumber);
    // Add the hotel to the hotels ArrayList
    this.hotels.add(hotel);
    // Print the confirmation number for the booked hotel
    System.out.println("Hotel booked! Confirmation number: " + confirmation
  }

  // Method to cancel a reservation
  public void cancelReservation(int confirmationNumber) {
    // Loop through the flights ArrayList to find the reservation
    for (Flight flight : this.flights) {
      // If the confirmation number matches, remove the flight reservation
      if (flight.getConfirmationNumber() == confirmationNumber) {
        this.flights.remove(flight);
        // Print the cancellation message for the flight
        System.out.println("Flight reservation with confirmation number " +
        return;
      }
    }
    // Loop through the hotels ArrayList to find the reservation
    for (Hotel hotel : this.hotels) {
      // If the confirmation number matches, remove the hotel reservation
      if (hotel.getConfirmationNumber() == confirmationNumber) {
        this.hotels.remove(hotel);
        // Print the cancellation message for the hotel
        System.out.println("Hotel reservation with confirmation number " +
        return;
      }
    }
    // Print a message if no reservation is found with the provided confir
    System.out.println("No reservation found with confirmation number " + c
  }

  // Method to generate a random 6-digit confirmation number
  private int generateConfirmationNumber() {
    // Create a Random object
    Random rand = new Random();
    // Generate and return a random 6-digit number
    return rand.nextInt(900000) + 100000;
  }
}
```

The above Java class is used for searching and booking flights and hotels, as well as cancelling reservations. It contains methods to search for flights and hotels based on specific criteria, book flights and hotels by creating new Flight and Hotel objects, cancel reservations by confirmation number, and generate a random confirmation number using the Random class in Java. It also contains private instance variables to store ArrayLists of Flight and Hotel objects that represent the current reservations.

```
// Flight.java
// Define the Flight class
public class Flight {
    // Declare an integer variable to store the flight number
    private int flightNumber;
    // Declare a string variable to store the passenger name
    private String passengerName;
    // Declare a string variable to store the origin of the flight
    private String origin;
    // Declare a string variable to store the destination of the flight
    private String destination;
    // Declare a string variable to store the date of the flight
    private String date;
    // Declare an integer variable to store the number of passengers
    private int numPassengers;
    // Declare a double variable to store the price of the flight
    private double price;
    // Declare an integer variable to store the confirmation number
    private int confirmationNumber;

    // Constructor to initialize the Flight object with provided details
    public Flight(int flightNumber, String passengerName, String origin, St
        // Initialize the flight number
        this.flightNumber = flightNumber;
        // Initialize the passenger name
        this.passengerName = passengerName;
        // Initialize the origin of the flight
        this.origin = origin;
        // Initialize the destination of the flight
        this.destination = destination;
        // Initialize the date of the flight
        this.date = date;
        // Initialize the number of passengers
        this.numPassengers = numPassengers;
        // Initialize the price of the flight
        this.price = price;
    }

    // Method to get the flight number
    public int getFlightNumber() {
        return flightNumber;
    }

    // Method to get the passenger name
    public String getPassengerName() {
        return passengerName;
    }

    // Method to get the origin of the flight
    public String getOrigin() {
        return origin;
    }

    // Method to get the destination of the flight
    public String getDestination() {
        return destination;
    }

    // Method to get the date of the flight
    public String getDate() {
        return date;
    }

    // Method to get the number of passengers
    public int getNumPassengers() {
        return numPassengers;
    }

    // Method to get the price of the flight
    public double getPrice() {
        return price;
    }

    // Method to get the confirmation number
    public int getConfirmationNumber() {
        return confirmationNumber;
    }

    // Method to set the confirmation number
    public void setConfirmationNumber(int confirmationNumber) {
        this.confirmationNumber = confirmationNumber;
    }
}
```

The above “Flight” class represents a flight. It has flight number, passenger name, origin, destination, date, number of passengers, price, and confirmation number. It has a constructor to create a Flight object and getters and setters to access and modify the object's properties. The confirmation number is set after a flight is booked to identify a reservation.

```
// Hotel.java
// Define the Hotel class
public class Hotel {
  // Declare an integer variable to store the hotel ID
  private int hotelId;
  // Declare a string variable to store the guest name
  private String name;
  // Declare a string variable to store the location of the hotel
  private String location;
  // Declare a string variable to store the check-in date
  private String checkIn;
  // Declare a string variable to store the check-out date
  private String checkOut;
  // Declare an integer variable to store the number of guests
  private int numGuests;
  // Declare a double variable to store the price of the stay
  private double price;
  // Declare an integer variable to store the confirmation number
  private int confirmationNumber;

  // Constructor to initialize the Hotel object with provided details
  public Hotel(int hotelId, String name, String location, String checkIn, S
    // Initialize the hotel ID
    this.hotelId = hotelId;
    // Initialize the guest name
    this.name = name;
    // Initialize the location of the hotel
    this.location = location;
    // Initialize the check-in date
    this.checkIn = checkIn;
    // Initialize the check-out date
    this.checkOut = checkOut;
    // Initialize the number of guests
    this.numGuests = numGuests;
    // Initialize the price of the stay
    this.price = price;
  }

  // Method to get the hotel ID
  public int getHotelId() {
    return hotelId;
  }
```

```java
  // Method to get the guest name
  public String getName() {
    return name;
  }

  // Method to get the location of the hotel
  public String getLocation() {
    return location;
  }

  // Method to get the check-in date
  public String getCheckIn() {
    return checkIn;
  }

  // Method to get the check-out date
  public String getCheckOut() {
    return checkOut;
  }

  // Method to get the price of the stay
  public double getPrice() {
    return price;
  }

  // Method to get the number of guests
  public int getNumGuests() {
    return numGuests;
  }

  // Method to get the confirmation number
  public int getConfirmationNumber() {
    return confirmationNumber;
  }

  // Method to set the confirmation number
  public void setConfirmationNumber(int confirmationNumber) {
    this.confirmationNumber = confirmationNumber;
  }
}
```

The above “Hotel” class represents a hotel, with an ID, a name, a location, a check-in date, a check-out date, a number of guests, a price, and a confirmation number. It contains a constructor that initializes these properties, as well as getter and setter methods for each property. The confirmation number is randomly generated and set through the setter method.

```java
// Main.java
// Define the Main class
public class Main {
  // Main method, the entry point of the application
  public static void main(String[] args) {
    // Create a new TravelApp object
    TravelApp app = new TravelApp();
    // Search for flights from New York to London on 2022-09-01 for 1 passe
    app.searchFlights("New York", "London", "2022-09-01", 1);
    // Search for hotels in London from 2022-08-01 to 2022-09-05 for 2 gues
    app.searchHotels("London", "2022-08-01", "2022-09-05", 2);
    // Book a flight with flight number 12345670 for Martin Nadine from Ne
    app.bookFlight(12345670, "Martin Nadine", "New York", "London", "2022-0
    // Book a flight with flight number 67843513 for Jennifer Ulrike from N
    app.bookFlight(67843513, "Jennifer Ulrike", "New York", "London", "2022
    // Book a hotel with hotel ID 98765432 for Martin Nadine in London from
    app.bookHotel(98765432, "Martin Nadine", "London", "2022-09-01", "2022-
    // Cancel the reservation with confirmation number 12345670
    app.cancelReservation(12345670);
  }
}
```

In the main() method of the above class, an instance of the “TravelApp” class is created. Several methods of the “TravelApp” class are called to perform various tasks related to travel. These tasks include searching for flights and hotels, booking flights and hotels, and cancelling reservations.

In particular, the “searchFlights()” method is called with the arguments "New York", "London", "2022-09-01", and 1, to search for flights from New York to London on September 1, 2022 for one passenger. Similarly, the “searchHotels()” method is called with the arguments "London", "2022-08-01", "2022-09-05", and 2, to search for hotels in London from August 1, 2022 to September 5, 2022 for two guests.

Then, two flights and one hotel are booked using the “bookFlight()” and “bookHotel()” methods respectively, with different arguments. Finally, a reservation for one of the flights is cancelled using the “cancelReservation()” method with the argument 12345670.

Sample Output:

```
Searching for flights from New York to London on 2022-09-01 for 1 passengers.
Searching for hotels in London from 2022-08-01 to 2022-09-05 for 2 guests.
Flight booked! Confirmation number: 528140
Flight booked! Confirmation number: 664315
Hotel booked! Confirmation number: 392396
No reservation found with confirmation number 12345670.
```

Flowchart: