Project Documentation

➤ Introduction

This project implements a Java console application designed to manage information related to an Easter holiday tourist destination. The user can add, delete, and view data related to attractions, accommodations, routes, and other points of interest. The application's data is stored locally using Serialization, ensuring persistence between executions.

➤ Application Objectives

The application implements the following basic functionalities:

1. Display all registered attractions, accommodations, routes, clients, etc.
2. Add a new entry in a specific category (e.g., new POI or new accommodation).
3. Search for entries based on name.
4. Delete all records.
5. View reviews for points of interest (where available).
6. Exit the application while saving data to files.

➤ Object-Oriented Design

The application is based on object-oriented programming principles. It includes classes such as:

- **Client**
- **Reservation**, which has the following subclasses:
    - **Destination**
    - **POI (Point of Interest)**
    - **Route**
- **Cost**
- **Custom Date**
- **Food**
- **Accommodation**

Inheritance is utilized in the Reservation class, which includes the subclasses Destination, POI, and Route.

➤ **Folder Structure**

- **Data**: Contains the data stored using Java serialization.
- **tourismApp**: Contains the main class and all other classes used by the program.

➤ **Main**

Purpose:
The main method is the starting point of the "Tourism App" application. The user interacts through a simple menu and can add, view, search, delete, and save tourist-related data (clients, reservations, accommodations, food).

Execution Steps:

1. **Call loadData():**
   Loads saved data from .ser files (clients, reservations, accommodations).
2. **Execute Menu in a do-while loop:**
   The user sees a menu and enters a number (1-6).
   The selected option is passed to the Action() method for execution.
   The loop continues until the user selects 6 (Save and Exit).

User Menu Options:

1. **View Lists:**
   Displays available records: Clients, Accommodations, Reservations, Food Types.
2. **Add New Entry:**
   The user can add:
   - A new Client (linked with food, accommodation, and reservation)
   - A new Accommodation
   - A new Reservation (Route, POI, Destination)
3. **Search for Client:**
   Searches for a client by name.
4. **Delete All Records:**
   Clears all client, reservation, and accommodation lists.
5. **View Reservation Ratings:**
   Displays reservation titles and their ratings (0 to 5).
6. **Save and Exit:**
   Saves lists to .ser files and terminates the program.

Saving & Loading Data:
Data is stored in the 'data' folder as serialized objects (using ObjectOutputStream).
At startup, the application attempts to read these files to reload previous entries.

Notes:

- The method ensures data consistency by prompting the user to enter valid options for food, accommodation, and reservation.
- The application relies on classes (Client, Reservation, Accommodation, Food) and uses serialization (Serializable) to store them.

➤ Example Executions

- Suppose we choose to add another client to the list:

```
Data loaded successfully!
-------------------------------------------------
Choose what you want to do:
1. View all: Clients / Accommodations / Reservations / Food
2. Add new Record: Client / Accommodation / Reservation
3. Search Client by Name
4. Delete all Records: Client,Accommodation,Reservation
5. View ratings
6. Save and exit
-------------------------------------------------
2
What record do you want to add? 1. Client 2. Accommodation 3. Reservation (1/2/3):
1
```

- The application will now request the client's details:

```
Give Client's name:
person3
Give Client's email:
person3@gmail.com
Available Food Options:
1. Burger - $5
2. Pasta - $8
3. Pizza - $10
4. steak - $15
5. salmon - $20
6. soup - $12
Enter your preferred food from the list:
Pizza
Available Accommodation Options:
1. hotel1 - $100 - Max Customers: 0
Enter your preferred accommodation from the list:
hotel1
Available Reservation Options:
1. codepath - $15
Enter your preferred reservation from the list:
codepath
Give Reservation Date (day):
1
Give Reservation Date (month):
1
Give Reservation Date (year):
2027
Client added successfully!
-------------------------------------------------
```

- Then we can observe the client has been added to the list:

```
Choose what you want to do:
1. View all: Clients / Accommodations / Reservations / Food
2. Add new Record: Client / Accommodation / Reservation
3. Search Client by Name
4. Delete all Records: Client,Accommodation,Reservation
5. View ratings
6. Save and exit
-------------------------------------------------
1
What do you want to view? 1. Clients 2. Accommodations 3. Reservations 4. Food (1/2/3/4):
1
[Client{name='person1', email='person1@gmail.com', food1='pizza', accomodation='hotel1', ReservationName='codepath', dat
e='Date: 01/01/2025'}, Client{name='person2', email='person2@gmail.com', food1='steak', accomodation='hotel1', Reservati
onName='codepath', date='Date: 01/01/2027'}, Client{name='person3', email='person3@gmail.com', food1='Pizza', accomodati
on='hotel1', ReservationName='codepath', date='Date: 01/01/2027'}]
```

The same applies to accommodations and reservations.

- Now we select function 3 (Find client):

```
Enter the name of the client to search:
person2
Client found: Client{name='person2', email='person2@gmail.com', food1='steak', accomodation='hotel1', ReservationName='c
odepath', date='Date: 01/01/2027'}
-------------------------------------------------
```

- If an unregistered name is entered, it returns that the client was not found...

```
3
Enter the name of the client to search:
person0
No client found with the name: person0
-------------------------------------------------
```

- Suppose we select option 5:

```
5
Ratings of Reservations:
-------------------------------------------------
Title: codepath             | Rating: 5/5
Title: museum               | Rating: 4/5
-------------------------------------------------
```

- The names of the reservations and their ratings are now displayed:

➤ Saving Data

- If we select option 6, the data for Clients, Accommodations, and Reservations will be saved in the data folder, and the program will exit.

```
6
Saving and exiting...
Data saved successfully!
-------------------------------------------------
```