

CS101- Algorithms and Programming I

Lab 07

Lab Objectives: ArrayLists.

IMPORTANT:

- For all labs in CS 101, your solutions must conform to the CS101 style guidelines (rules!).
 - You should not use break statements in your solution.
 - Before implementing your programs, you should analyse the problem and plan and write the algorithm you will use to solve the problem.
1. For this lab you will create an application that is used by a restaurant to store reservations. The information stored about each reservation is the date/time, the name of the customer, the phone number of the customer.

Your application will use 3 `ArrayLists`, one to store the Date/Times of the reservation, one to store the names of the customers and one to store the phone numbers of the customers making the reservations.

- `reservations`: an `ArrayList` that stores `LocalDateTime` objects each one representing a reservation.
- `customerNames`: an `ArrayList` that stores the names the customers who have made the reservations
- `customerPhones`: an `ArrayList` that stores the phone numbers of the customers making the reservations.

The `ArrayLists` are parallel lists, meaning that the reservation date/time that is at index 0 of the reservation list, will have been made by the customer whose name and phone number is at the same index in the `customerNames` and `customerPhones` list.

You will store reservation dates as `LocalDateTime` objects. In your methods to compare reservations, you may need to create `LocalDateTime` objects. To construct a `LocalDateTime` object, the constructor is as follows:

```
LocalDateTime(int year, Month month, int date, int hour, int minute )
```

You may create final constant global variables (static class variable) for year (assume 2024) and month ("APRIL"). To get the `Month` object you may use `Month.valueOf()`. Example:

`Month.valueOf("APRIL")` creates a `Month` object for April.

2. Create an application, `Lab07Q1.java`. Your application should implement the following helper methods.
 - `boolean isBetween()` : takes as parameters three `LocalDateTime` objects, check, start, end. The method returns true if the check object is between the start and end `LocalDateTime` and false if not. Hint: see the documentation for the `compareTo()` method in the `LocalDateTime` class.

- `ArrayList<Integer> findReservations() :` takes an int date and a list of reservations as parameters, and returns an `ArrayList` containing the indexes of the reservations in the list with **the given date** (ex: between 00:00 and 23:59 on the given date). To simplify the question, you may assume all reservations are in the same month in 2024. You should use the `isBetween()` method to find reservations on a given date.
- `boolean cancelReservation() :` takes a name and phone number, and `ArrayLists` of reservations, customer names and customer phone numbers as parameters. Finds the matching customer (with the same name and phone number) and removes the customer's name, phone number and reservation date from the `ArrayLists`.
- `printReservations() :` takes an `ArrayList` containing Integer indexes, and `ArrayLists` containing reservations, customer names and customer phone numbers as parameters. Displays the reservation information for the customers with the given indexes. If null is passed as the first parameter, displays the information for all customers.

3. Your application should do the following:

- Download `ReservationGenerator` and using the static methods provided in `ReservationGenerator`, create 3 `ArrayLists` that store the reservation dates, customer names, and customer phone numbers for 10 reservations. Note: `ReservationGenerator` will generate reservations for the given month, and between the given dates for 2024. Reservations will be between 18:00 – 22:00.
- Display the menu shown in the sample run, and implement the functionality as follows:
 1. Search reservations by date: Inputs an int date and finds all reservations on the given date. Display the reservations using the `printReservations()` method.
 2. Cancel reservation: inputs a name and phone number and cancels the reservation with the given name and phone number. Display a message if the reservation is/is not found.
 3. Displays all reservations.
 4. Exit.

Sample Run:

1)Search reservations by date

2)Cancel Reservation

3)Display all reservations

4)Exit

Enter choice: 3

Serdar	(531)	542	33	32	APRIL, 16	8:30
Aslan	(532)	544	33	32	APRIL, 17	7:30
Eda	(521)	762	33	32	APRIL, 16	8:00
Ergem	(598)	394	33	32	APRIL, 16	7:00
Okan	(531)	977	33	32	APRIL, 16	5:30
Ceylan	(503)	542	22	32	APRIL, 16	7:30
Ela	(504)	872	33	32	APRIL, 16	5:00
Lara	(543)	922	33	32	APRIL, 16	5:00
Selin	(530)	776	33	32	APRIL, 17	7:00
Can	(541)	892	33	32	APRIL, 16	6:30

- 1)Search reservations by date
- 2)Cancel Reservation
- 3)Display all reservations
- 4)Exit

Enter choice: 1

Enter date to search: 16

Reservations on April 16

Serdar	(531)	542	33	32	APRIL, 16	8:30
Eda	(521)	762	33	32	APRIL, 16	8:00
Ergem	(598)	394	33	32	APRIL, 16	7:00
Okan	(531)	977	33	32	APRIL, 16	5:30
Ceylan	(503)	542	22	32	APRIL, 16	7:30
Ela	(504)	872	33	32	APRIL, 16	5:00
Lara	(543)	922	33	32	APRIL, 16	5:00
Can	(541)	892	33	32	APRIL, 16	6:30

- 1)Search reservations by date
- 2)Cancel Reservation
- 3)Display all reservations
- 4)Exit

Enter choice: 2

Enter name of customer: Ela

Enter phone number of customer: (504) 872 33 32

Reservation cancelled....

- 1)Search reservations by date
- 2)Cancel Reservation
- 3)Display all reservations
- 4)Exit

Enter choice: 2

Enter name of customer: Enes

Enter phone number of customer: (531) 542 33 32

Reservation not found...

- 1)Search reservations by date
- 2)Cancel Reservation
- 3)Display all reservations
- 4)Exit

Enter choice: 1

Enter date to search: 10

No reservations found...

- 1)Search reservations by date
- 2)Cancel Reservation
- 3)Display all reservations
- 4)Exit

Enter choice: 4