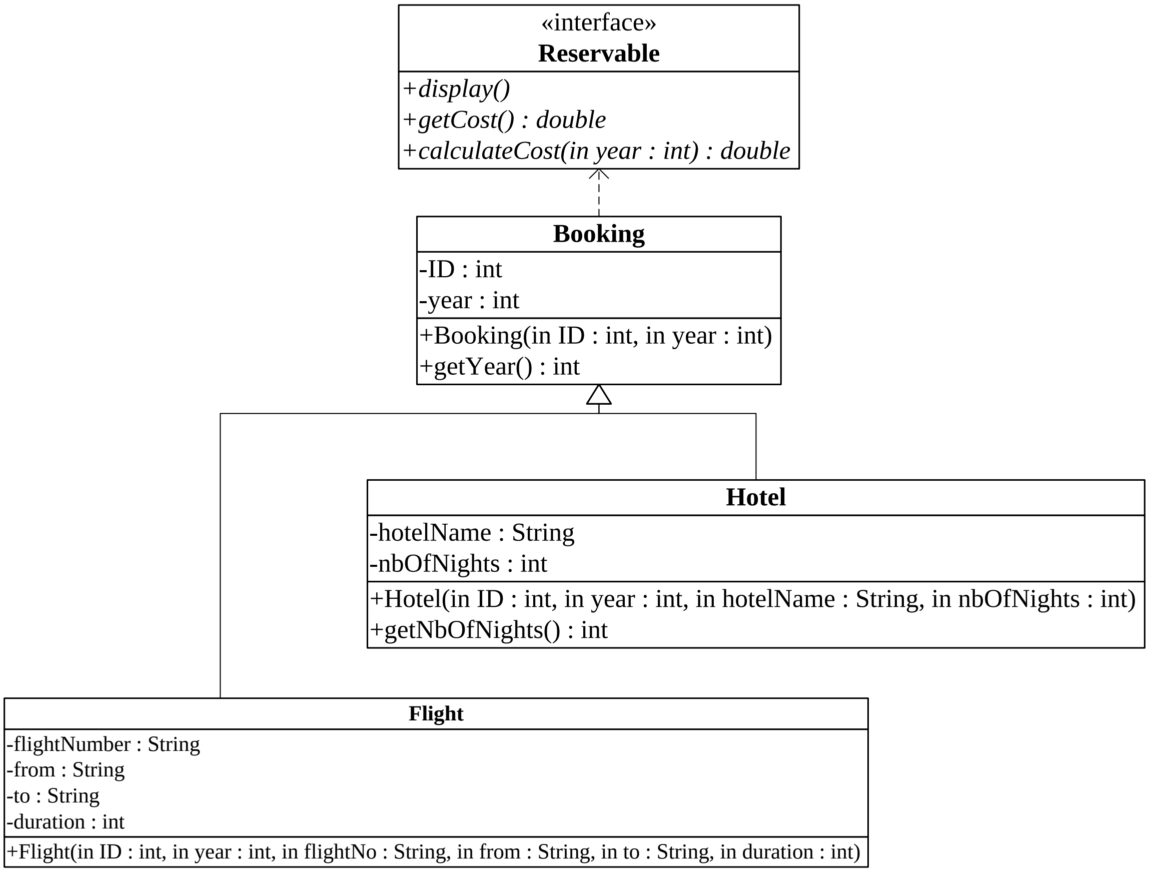
King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC113 – Computer Programming II – Midterm 2 Exam – Spring 2016

Exercise1:


«interface»
Reservable
+display()
+getCost() : double
+calculateCost(in year : int) : double
Booking
-ID : int
-year : int
+Booking(in ID : int, in year : int)
+getYear() : int
Hotel
-hotelName : String
-nbOfNights : int
+Hotel(in ID : int, in year : int, in hotelName : String, in nbOfNights : int)
+getNbOfNights() : int
Flight
-flightNumber : String
-from : String
-to : String
-duration : int
+Flight(in ID : int, in year : int, in flightNo : String, in from : String, in to : String, in duration : int)

Reservable Interface***:***

- Methods:
  - ***display()*:** this method displays **<u>all</u>** attributes of the Reservable object.
  - ***getCost()*:** this method returns the cost of the Reservable object calculated by the method calculateCost and using the attribute year of the Reservable object.
  - ***calculateCost(year: int)*:** this method calculates and returns the cost of the Reservable object. It is calculated as follows:
    - For ***Flight Booking***: cost = year / 10 + Flight Duration * 10 .
    - For ***Hotel Booking:*** if the Hotel Booking is done in or before 2010, the cost is 2000 SAR. For any year after 2010 the cost is 10 % greater than the cost of the previous year.

      Cost for current year = 2000 SAR if current year is less or equal to 2010.

      Otherwise: cost for current year = 1.1 * cost for previous year.

Booking class***:***

- Attributes:
  - ***ID***: the ID of the Booking.
  - ***year:*** the year of the Booking.
- Methods:
  - ***Booking (ID:*** *int****,*** ***year:*** *int****)***: constructor
  - ***getYear()*****:** this method returns the year of the Booking.

Hotel class

- Attributes:
  - ***hotelName:*** the name of the Hotel.
  - ***nbOfNights:*** the number of nights spent in the Hotel.
- Methods:
  - ***Hotel (ID:*** *int****,*** ***year:*** *int****,*** ***hotelName:*** *String****,*** ***nbOfNights:*** *int****)***: constructor.
  - ***getNbOfNights()***: this method returns the number of nights spent in the Hotel.

Flight class

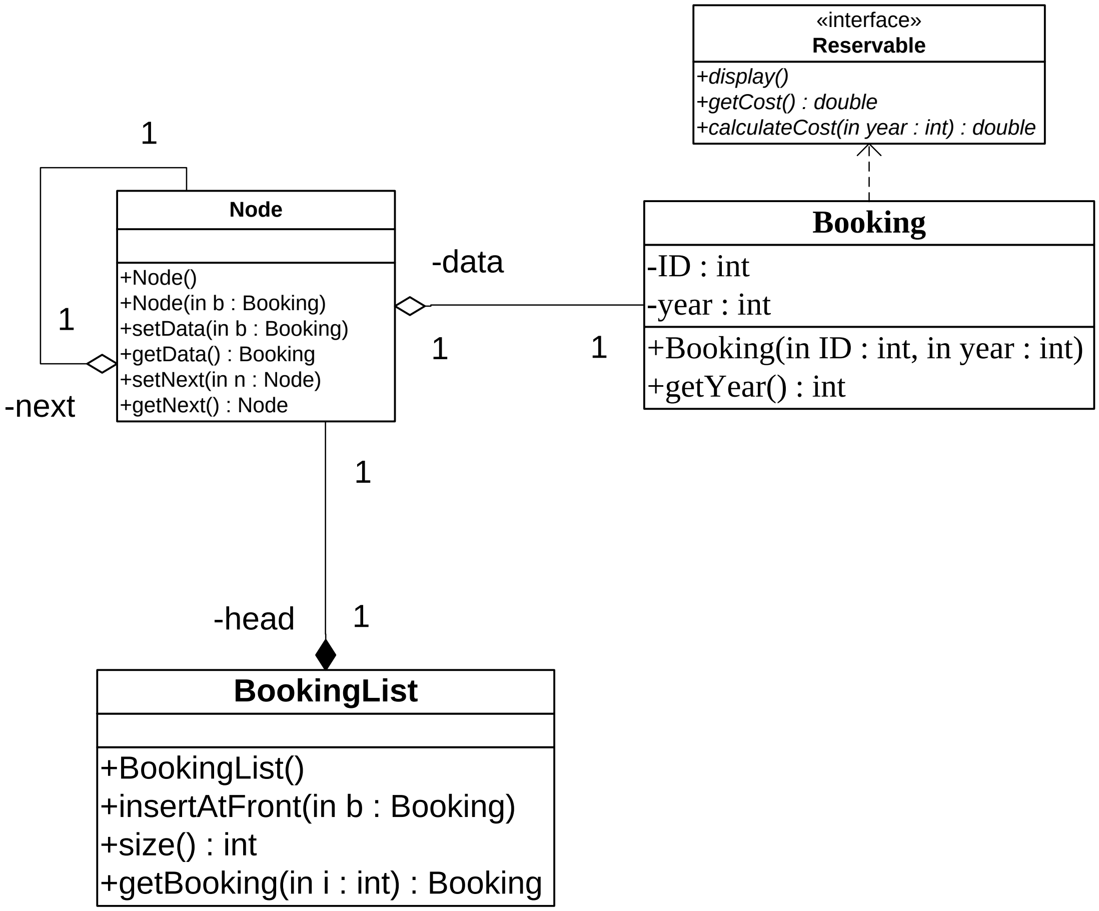
- Attributes:
  - ***flightNumber:*** the Flight number.
  - ***from***: the name of the departure Airport.
  - ***to:*** the name of the arrival Airport.
  - ***duration:*** the Flight's duration (in minutes).
- Methods:
  - ***Flight (ID:*** *int****,*** ***year:*** *int****,*** ***flightNo:*** *String****,*** ***from:*** *String****,*** ***to:*** *String****,*** ***duration:*** *int****)***: constructor.

QUESTION: Translate into Java code:

- the Interface ***Reservable***,
- the class ***Booking***
- and the class ***Hotel***.
- For the method ***calculateCost*** , propose 2 solutions (an **iterative solution** and a **recursive solution**).

Exercise 2:

Let’s consider the same class ***Booking*** described in exercise 1.



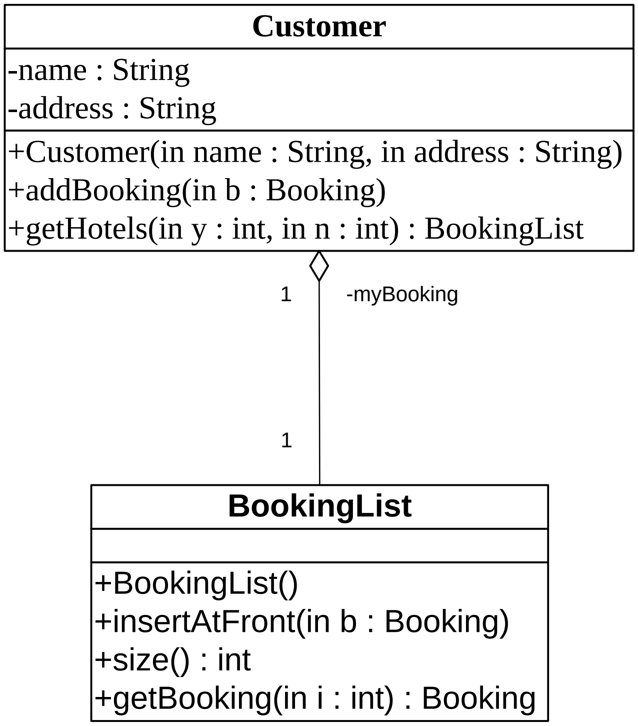
BookingList class***:***

- Methods:
  - ***BookingList ()***: constructor.
  - ***insertAtFront (b: Booking)***: this method insert the Booking ***b*** at the beginning of the list.
  - ***size* ():** this method returns the number of elements of the list.
  - ***getBooking(i: int*)**: this method returns the Booking object stored in the node at position ***i***. The position of the first node is 1. If the parameter ***i*** is less than 1 or greater than the number of elements of the list, this method throws an ***Exception*** with the message “Position out of bounds”.

QUESTION: Translate into Java code the class ***BookingList.***

Exercise 3:

Let’s consider the same class ***BookingList*** described in exercise 2.


Customer
-name : String
-address : String
+Customer(in name : String, in address : String)
+addBooking(in b : Booking)
+getHotels(in y : int, in n : int) : BookingList
1
-myBooking
1
BookingList
+BookingList()
+insertAtFront(in b : Booking)
+size() : int
+getBooking(in i : int) : Booking

Customer class***:***

- Attributes:
    - ***name***: the customer name.
    - ***address :*** the address of the customer.
- Methods:
    - ***Customer (name:*** *String****,*** ***address:*** *String****)***: constructor.
    - ***addBooking (b: Booking)***: this method adds the Booking ***b*** to the customer.
    - ***getHotels(y:*** *int****,*** ***n:*** *int****):*** this method returns a BookingList object containing all Hotel Bookings in the year ***y*** and having the number of nights greater than ***n***.

QUESTION: Translate into Java code the class ***Customer.***