INSTITUTO POLITÉCNICO DE BRAGANÇA
Escola Superior de Tecnologia e Gestão

Informatics Engineering / Management Informatics
April, 5th, 2018

Delivery deadline: 04/23/2018

MANAGEMENT OF HOTEL ROOMS

A particular hotel urgently needs a system that allows to make a proper rooms occupation management. To do this, the system should keep the record of all occupations (accommodations) of its rooms, past and present. The relevant information for each accommodation are: an integer code that identifies the accommodation, the discount rate applied, the total cost, the check-in and checkout dates. Both the checkout date as the total cost are saved in the checkout moment, being the total cost determined based on the daily room price, discount rate applied and the days of occupation.

Over time, the same room may be occupied by multiple guests and a single guest may stay in several of them. The rooms are characterized by the room number, number of beds and daily price. Already for each guest, it will only be necessary to know the name and the integer code of the guest.

As features, the application should allow the registration of new rooms and new guests, accommodate a guest in a particular room (register new accommodation), checkout a given room, convert to a string the unoccupied rooms, convert to a string every stay of a given guest, and determine the total amount that has been invoiced to a given guest.

The code for identification of different entities (rooms, clients and occupation) should be automatically assigned by the system, by using the code 1 for the first element, 2 for the second, and so on – in the case of occupations, once they are distributed throughout several collections, the id values must be obtained using a global counter of instances of this class.

Work to develop

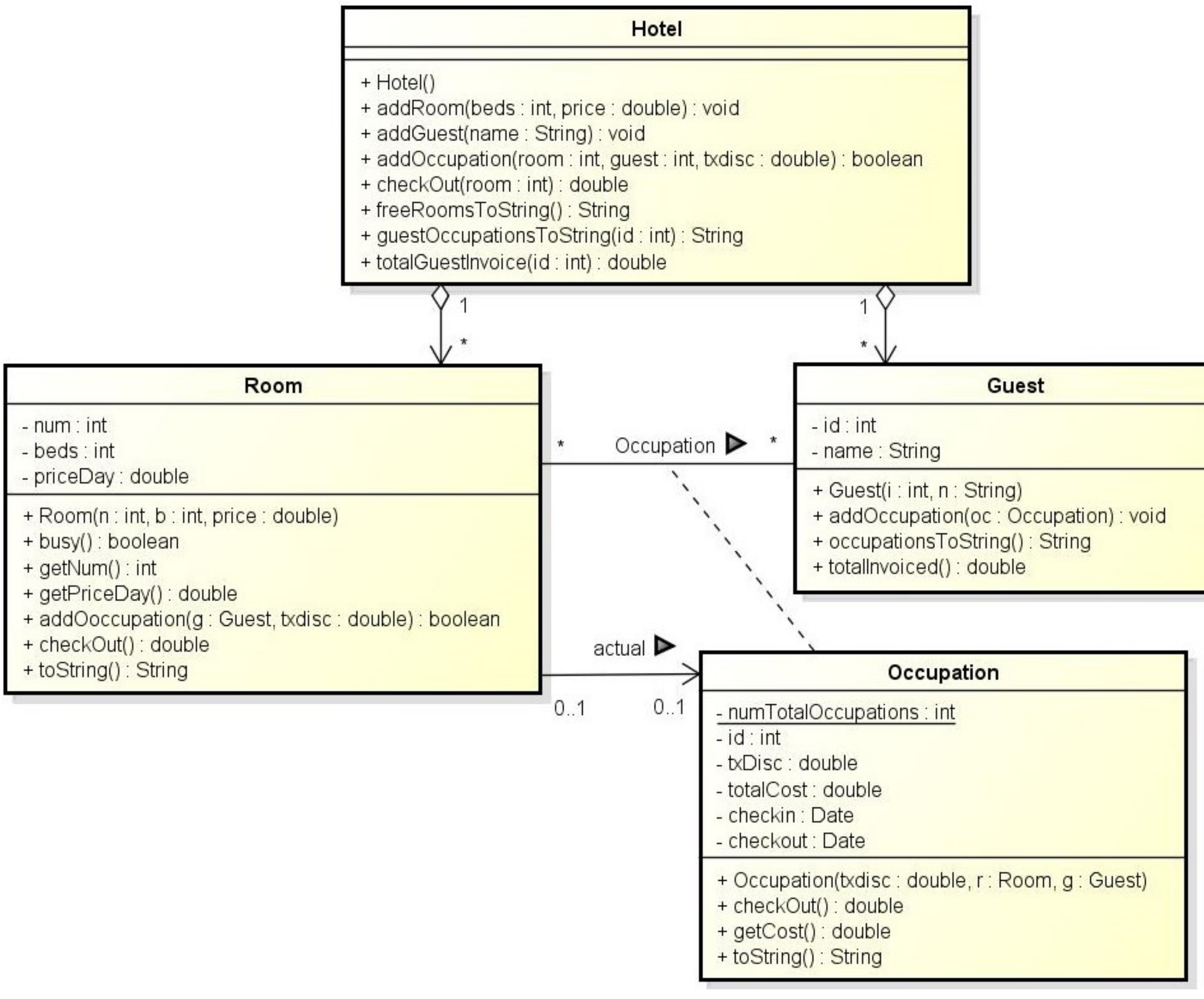
Implement in Java a solution for the problem above. The solution should be a faithful translation of the UML class diagram given at the end of this document.

Considerations to be consider in the implementation

- The application must be developed in NetBeans IDE, JDK 1.7 or later.
- All developed classes should reside in a package named "tp1 ".
- The code should include the appropriate javadoc comments for further production of the application documentation.
- The javadoc block preceding the Hotel class must include @author tags with the identification of the group elements (name, number and degree program).
- Fully respect the attributes and methods presented in the class diagram, in particular, use the same identifiers (note the capitalization) and does not implement any other methods or attributes that are not in the diagram.
- On the implementation of collections, you should use the collections of the JCF (Java Collections Framework).
- Suppose the application is running in real time. Therefore, for the checkin and checkout you must consider the current date, obtained from the operating system - this is what will make sense when the application is in its normal operation. To do this, instantiate the class Date, since its default constructor creates an object with the current date and time.
- You should build a class with a main method that uses all the features of the application developed.

Global Rules (mandatory)

1. The assignment must be carried out by groups of 2 or 3 students.
2. This and the next work are optional for students who have already achieved approval in the practical component of the curricular unit in one of the last two academic years.
3. Only shall be accepted for assessment work whose implementation does not present any compilation error and with a minimum of features perfectly operational.
4. It's expressly forbidden to copy all or part of code from sources other than the documentation made available by the teachers of ADS.
5. The work must be delivered, within the delivery deadline, to the e-learning portal (at http://virtual.ipb.pt/, choose <Trabalho Pratico 1> in the <Trabalhos> tab, within the AED area), and under no circumstances may be sent by email.
6. It should be submitted the compressed folder of the project. To do this, from the File menu of NetBeans, select Export Project->To ZIP.
7. The work may only be submitted with a maximum delay of five days. The classification is subtracted by one, for each day of delay in the delivery of the work.
8. Resubmissions will not be allowed (when submitting ensure that it is the final version).
9. The students may need to present in person the work, on a date to scheduler by the teachers, demonstrating being able to implement the code, understand it and explain it.


Hotel
+ Hotel()
+ addRoom(beds : int, price : double) : void
+ addGuest(name : String) : void
+ addOccupation(room : int, guest : int, txdisc : double) : boolean
+ checkOut(room : int) : double
+ freeRoomsToString() : String
+ guestOccupationsToString(id : int) : String
+ totalGuestInvoice(id : int) : double
1
*
1
*
Room
- num : int
- beds : int
- priceDay : double
+ Room(n : int, b : int, price : double)
+ busy() : boolean
+ getNum() : int
+ getPriceDay() : double
+ addOoccupation(g : Guest, txdisc : double) : boolean
+ checkOut() : double
+ toString() : String
*
Occupation
*
Guest
- id : int
- name : String
+ Guest(i : int, n : String)
+ addOccupation(oc : Occupation) : void
+ occupationsToString() : String
+ totalInvoiced() : double
actual
0..1
0..1
Occupation
- numTotalOccupations : int
- id : int
- txDisc : double
- totalCost : double
- checkin : Date
- checkout : Date
+ Occupation(txdisc : double, r : Room, g : Guest)
+ checkOut() : double
+ getCost() : double
+ toString() : String