

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, show the unoccupied rooms, show 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 C++ 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 MS Visual C++ 2010 or later.
- Fully respecting 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 – the underline is used in the diagram to represent static members. Only in classes where appropriate, should also include the operator that allows the objects to be collectibles.
- Each class must be defined using two files, one with the class declaration (*.h) and another with its implementation (*.cpp). Both files should have exactly the same name as the class.
- On the implementation of collections, you should use the template Collection and/or HybridCollection (available on ipb.virtual) and it's totally forbidden to change whatever it is that template.
- For the datatype date you must use the TDate, provided for this purpose on ipb virtual platform (.h and .cpp files) and it's totally forbidden to change whatever it is that class.
- Consider that the application is running in real time. So, for checkin and checkout dates, you should consider the current date, given by the operating system – that's what will make sense when the application is in normal running. Use, for the purpose, the functionalities of TDate class.
- You should build a main function that uses all the features of the application developed.

Global considerations

- The assignment must be carried out by groups of 2 students.
- Only shall be accepted for assessment work whose implementation does not present any compilation error or of linking and with a minimum of features perfectly operational.
- It's expressly forbidden to copy all or part of code from sources other than the documentation made available by the teachers of OOP.
- The work must be delivered to the e-learning portal (<http://virtual.ipb.pt/>, option "Trabalhos"), within the delivery deadline. Should be submitted the compressed folder of the solution in C++, once deleted, if any, the ".sdf", ".VC" and ".VC.db" file extensions and all the subfolders with name "Debug", "Release" and "ipch".
- The name of the solution folder in the Visual C++ to submit should be like "NameLastName1n1c1_NameLastName2n2c2", containing the first and last name, and course of two elements of the group. (Example: "MariaRita2045ig_RuiSilva2183ei.zip".)
- One point will be subtracted from your grade for each day of delay in the delivery of the work.
- Students must present in person the work, on a date to scheduler by the teachers, Demonstrating be able to implement the code, understand it and explain it.

