Lab Statement

13th November, 2018

In Lab7, you had created few classes that simulate a various types of rooms. Further in Lab8 classes, you added the **HotelSuite** and **Hotel** classes which helped in Hotel Management. In this Lab, we built upon the same.

You are required to create a class **HotelBill**.

It has the following fields:

- `String filename` :

  Holds the filename of the file which has to be parsed. The file is present in the current directory.
- `Hotel hotel` :

  Stores the hotel which is created by the reading the contents of the file, filename.

It also has the following main functionalities :

- `int readContent()` :

  This function reads the content of the file, given by filename and creates a hotel. It also returns the number of *HotelSuites* in the thus created *Hotel*.

  The contents of the file are in the following format :

```
HotelSuite:
Room:<roomtype>,<length>,<width>,<height>
Room:<roomtype>,<length>,<width>,<height>
....(multiple rooms can come here)
HotelSuite:
Room:<roomtype>,<length>,<width>,<height>
Room:<roomtype>,<length>,<width>,<height>
....(multiple rooms can come here)
....(multiple hotel suites can come here)
```

where the roomtype can be either lr, wr, br corresponding to *LivingRoom*, *WashRoom*, and *BedRoom* respectively, and the length, width and height are integers.

A sample file is given below:

```
HotelSuite:
Room:lr,6,7,8
Room:wr,6,4,8
Room:br,6,6,8
HotelSuite:
Room:lr,10,7,8
Room:wr,10,4,8
Room:br,10,6,8
Room:br,10,6,8
Room:br,8,6,8
```

As before, a HotelSuite has has a one *LivingRoom*, one WashRoom and multiple (zero or more) BedRooms (bound by the `MAX_NUMBER_OF_BED_ROOMS` variable).

A pseudocode for a possible implementation of the above objective is as follows :

```
Open the file.
While the file is not empty :
    Read a line
    If the line starts with “HotelSuite” :
        Create and add the HotelSuite to
        hotelSuites Array appropriately.

        Initialize the necessary temporary
        variables for creating the next
        HotelSuite.
        Else if it starts with “Room” :
            Extract room features separated by
            ‘,’ to get the dimensions of the
            room

            If it is a ‘lr’ :
                Create a LivingRoom and store
                in temporarily
            Else if it is a ‘wr’ :
                Create a WashRoom and store
                in temporarily
            Else if it is ‘br’ :
                Create a BedRoom and store
                in a temporary array of the
                Same

end while

Create the Hotel from the created HotelSuites and
store the same in the hotel field

Close the file correctly.

Return the number of HotelSuites created
```

Also take care of any exceptions. If any exceptions occur, Print the error's toString() and return *-1* .

- `void writeResult(String content)` :

  Opens the file, filename and appends the string, content to it. Note: Write a newline character before

- `int getCost()` :

  Returns the cost of the Hotel hotel if not null. Returns -1 otherwise.

- `static void main(String[] args)` :

  The main function reads the filename from the user. Following which it creates a object of *HostelBill* passing the filename read from the user as the constructor parameter. It then prints on the screen using the println() method and append to the file, the following string :

  ```
  Total:<numberOfHotelSuites>,<costOftheHotel>
  ```

Additional Notes:

1. The document is complete and correct. If any changes are required, you would find it uploaded on Gitlab immediately.
2. **You are strongly advised to use Eclipse**.
3. Take care of the string format. Stick to the format specified, with the spaces and orientation of lowercase and uppercase characters.
4. All calculations should be of integer type. Do not use any other data type for the same.

5. Make sure the existing code works. Do not break the existing functionality while making the changes.
6. TAs will not be helping you to identify any of the exceptions.