

JS Advanced - Exam: 17.03.2019

Exam problems for the [“JavaScript Advanced” course @ SoftUni](#).

Submit your solutions in the SoftUni judge system at <https://judge.softuni.bg/Contests/1586/>.

Problem 3. Hotel

```
class Hotel {  
    // TODO: Implement this class...  
}
```

Write a **JavaScript** class **Hotel** which has the following **functionality**:

Constructor

You should implement at least these properties:

- **name** - string
- **capacity** - number
- **bookings** - array
- **currentBookingNumber** - number, initialized as '1'

At **initialization** of the **Hotel** class, the **constructor** receives only **2 parameters** (**name**, **capacity**).

The **bookings** property needs to be **empty** by **default**.

- The **name** property refers to the **hotel's name**.
- The **capacity** property refers for the **count** of the **hotel's rooms**.
 - The **single** rooms are **50 %** of the given **capacity**
 - The **double** rooms are **30 %** of the given **capacity**
 - The **maisonette** rooms are **20 %** of the given **capacity**
 - **NOTE**: All room counts should be integers.

Accessors

roomsPricing - Returns an **object**, containing the following properties:

- **single: 50** (price per night)
- **double: 90** (price per night)
- **maisonette: 135** (price per night)

servicesPricing - Returns an **object**, containing the following properties:

- **food: 10** (per order)
- **drink: 15** (per order)
- **housekeeping: 25** (per order)

Functions

rentARoom({clientName}, {roomType}, {nights})

Receive **3** parameters: **clientName** (string), **roomType** (string), **nights** (number)

If the hotel has **available** rooms of the requested **roomType**, you should create an **object** for the current client's **booking** which contains properties for (**client name**, **room type**, **nights**, and **room number**) and **store** it into the **bookings** property.

- You should also **increase** the **currentBookingNumber** by **1**. The first given **currentBookingNumber** is **1**.
- You should also **decrease** the **capacity** of the requested **roomType** rooms by **1**.

Then you should return a **string**, in the following format:

"Enjoy your time here Mr./Mrs. {clientName}. Your booking is {currentBookingNumber}."

Validations

- If there are **NO** available rooms of the requested **roomType**, the **function** should return a **string**, consisting of the message: "No {roomType} rooms available!"
And for **each** of the **other available** room types, you should **append** a **string** to the **message** in the following format:

"Available {roomType} rooms: {roomTypeRoomsRemaining}."

Example:

"No single rooms available! Available double rooms: 2. Available maisonette rooms: 1."

roomService({currentBookingNumber}, {serviceType})

Receive **two** parameters: **currentBookingNumber** (number) and **serviceType** (string)

You should **create** an **additional property** for the **booking** with the given **currentBookingNumber**, called **services** (array), and add the given **serviceType** to it (if it is already created, **just** add the new **serviceType** to it).

NOTE: The **services** property contains an array of **strings (serviceTypes)**. You may add the same **serviceType** **multiple times**.

Then you should return a **string**, in the following format:

"Mr./Mrs. {clientName}, Your order for {serviceType} service has been successful."

Validations

- If the given **currentBookingNumber** is **invalid (non-existent)** return the following string:
"The booking {currentBookingNumber} is invalid."
- If the **requested service** is **invalid**, return the following string:
"We do not offer {serviceType} service."

checkOut({currentBookingNumber})

Receive **one** parameter - **currentBookingNumber** (number)

You should **calculate** the due amount, as **totalMoney**, for the **booking** with the given **currentBookingNumber**, depending on the **nights** spent and the **price** of the **roomType**.

- You should also **increase** the **capacity** of the given **roomType** rooms by **1**.

Then you should return a **string**, in the following format:

"We hope you enjoyed your time here, Mr./Mrs. {clientName}. The total amount of money you have to pay is {totalMoney} BGN."

If the **booking** contains **services**, you should calculate the total price for all the **services**, as **totalServiceMoney**.

Then you should return a **string**, in the following format:

"We hope you enjoyed your time here, Mr./Mrs. {clientName}. The total amount of money you have to pay is {totalMoney + totalServiceMoney} BGN. You have used additional room services, costing {totalServiceMoney} BGN."

Validations

- If the given **currentBookingNumber** is **invalid** (non-existent) return the following string:
"The booking {currentBookingNumber} is invalid."

report()

This function represents the hotel's "database". Prints **information** about **each booking**.

The information should be presented in the following format:

"{hotelName.toUpperCase()} DATABASE:

----- (separate the header and bookings information - 20 dashes)

bookingNumber - {bookingNumber}

clientName - {clientName}

roomType - {roomType}

nights - {nights}

services: {services} (Optional. Only for those bookings who have used room services. The **services** should be joined with a **comma** and a **space**.)

----- (booking separator - 10 dashes)

...

Validations

- If there are no bookings, **return** the following **string**:
"{hotelName.toUpperCase()} DATABASE:
----- (separate the header and bookings information - 20 dashes)
There are currently no bookings."

Submission

Submit only the **Hotel** class as **JavaScript** code.

Constraints

The given capacity will be greater or equal to 10.

Examples

Input 1

```
let hotel = new Hotel('HotUni', 10);

console.log(hotel.rentARoom('Peter', 'single', 4));
console.log(hotel.rentARoom('Robert', 'double', 4));
console.log(hotel.rentARoom('Geroge', 'maisonette', 6));
```

Output 1

Enjoy your time here Mr./Mrs. Peter. Your booking is 1.
Enjoy your time here Mr./Mrs. Robert. Your booking is 2.
Enjoy your time here Mr./Mrs. Geroge. Your booking is 3.

Input 2

```
let hotel = new Hotel('HotUni', 10);

hotel.rentARoom('Peter', 'single', 4);
hotel.rentARoom('Robert', 'double', 4);
hotel.rentARoom('Geroge', 'maisonette', 6);

console.log(hotel.roomService(3, 'housekeeping'));
console.log(hotel.roomService(3, 'drink'));
console.log(hotel.roomService(2, 'room'));
```

Output 2

Mr./Mrs. Geroge, Your order for housekeeping service has been successful.
Mr./Mrs. Geroge, Your order for drink service has been successful.
We do not offer room service.

Input 3

```
let hotel = new Hotel('HotUni', 10);
```

```
hotel.rentARoom('Peter', 'single', 4);
hotel.rentARoom('Robert', 'double', 4);
hotel.rentARoom('Geroge', 'maisonette', 6);

hotel.roomService(3, 'housekeeping');
hotel.roomService(3, 'drink');
hotel.roomService(2, 'room');

console.log(hotel.report());
```

Output 3

HOTUNI DATABASE:

bookingNumber - 1

clientName - Peter

roomType - single

nights - 4

bookingNumber - 2

clientName - Robert

roomType - double

nights - 4

bookingNumber - 3

clientName - Geroge

roomType - maisonette

nights - 6

services: housekeeping, drink