

JS OOP Homework

A real estate agency operates with several types of estates: **apartments**, **offices**, **houses** and **garages**. It holds a database of **sale offers** and **rent offers**. Each estate has **unique name**, **type** (apartment, office, house or garage), **area** (in square meters) and **location** and can have **furniture** or not. **Apartments** and **offices** have additionally number of **rooms** and may have **elevator** in the building or not. **Houses** have additionally number of **floors**. **Garages** have additionally **width** and **height** (in meters). **Sale offers** hold an **estate** for sale and **sale price**. **Rent offers** hold an **estate** for rental and **rental price**.

Input Source Code

You are given a **JavaScript project (source code)** holding an **engine** for working with estates and offers that executes the following **commands** (see the sample input and output below):

- **create Apartment name area location isFurnished numberOfRooms hasElevator** – adds an apartment in the database by given unique name and other parameters.
- **create Office name area location isFurnished numberOfRooms hasElevator** – adds an office by given unique name and other parameters.
- **create House name area location isFurnished numberOfFloors** – adds a house.
- **create Garage name area location isFurnished width height** – adds a garage.
- **create RentOffer estateName rentPrice** – adds a rent offer for existing estate and given price.
- **create SaleOffer estateName salePrice** – adds a sale offer for existing estate and given price.
- **status** – prints all **estates** and **offers** from the database (in the order of their creation) in the format like in the sample output below. The engine knows how to find and print all estates and offers. You need to implement just the printing of each individual estate and offer.
- **find-sales-by-location location** – finds all sale offers for the specified location (case-sensitively), ordered by name and prints them in the format like in the sample output below. Prints "No offers" in case of no matches.

If a command or its parameters are invalid, it should fail to execute and its output should be **"Invalid command."**

Implement the Missing Classes

Your **first task** is to **implement all missing classes** to model the real estates agency, estates and offers using the **best practices** for JavaScript object-oriented programming (OOP). **Avoid duplicated code** though inheritance. Encapsulate correctly all fields with getters and setters with data validation. Create the following classes:

- **Estate**(name: string, area: number, location: string, isFurnished: boolean) – abstract class, holding an estate
- **BuildingEstate**(name: string, area: number, location: string, isFurnished: boolean, rooms: number, hasElevator: boolean) – abstract class, holding an estate in a building (used for apartments and offices)

- **Apartment**(name: string, area: number, location: string, isFurnished: boolean, rooms: number, hasElevator: boolean)
- **Office**(name: string, area: number, location: string, isFurnished: boolean, rooms: number, hasElevator: boolean)
- **House**(name: string, area: number, location: string, isFurnished: boolean, floors: number)
- **Garage**(name: string, area: number, location: string, isFurnished: boolean, width: number, height: number)
- **Offer**(estate: Estate, price: number) – abstract class holding an offer (used for sales offers and rent offers)
- **RentOffer**(estate: Estate, price: number)
- **SaleOffer**(estate: Estate, price: number)

See the class diagram for better understanding the class hierarchy:

Validation Rules

- Estate **name** and **location** should be non-empty strings.
- Estate **area** should be an integer in range [1...10000].
- All Boolean properties should be accept boolean values (**true** or **false**).
- Office / apartment **rooms** should be integer in range [0...100].
- House **floors** should be integer in range [1...10].
- Garage **widths** and **heights** should be integers in range [1...500].
- Offer **price** should be a positive integer.

Implement the Existing Commands

Implement your classes so that the engine (**EstatesEngine**) **executes correctly all above described commands**.

Implement Additional Commands

Modify the engine (**EstatesEngine**) to implement support for the following commands:

- **find-rents-by-location location** – finds and prints all rent offers for the specified location (case-sensitively), ordered by name, in the format like in the sample output below. Location cannot be empty.
- **find-rents-by-price minPrice maxPrice** – prints all rent offers within the specified price range (inclusively), ordered by price and by name (as second criteria), in the format like in the sample below. The parameters **minPrice** and **maxPrice** should be integers.

Sample Input

```
create Apartment aptLozenec24 150 Sofia true 4 true
create Apartment aptBotev28 54 Sofia true 2 false
```

```
create Apartment invalidApartment invalid_number true 5 true
create Apartment invalidApartment2 120 true 5
create Apartment invalidApartment3 invalid_number true 5 invalid
create Apartment invalidApartment2 120.5 true 5
create Hotel myHotel 150 true 20 true
```

status

```
create Office officeVitosha44 70 Sofia true 1 false
create Office officePlovdiv 44 Plovdiv false 1 true
create House houseBankya 206 Bankya true 3
create House houseSofia 120 Sofia true 1
create Garage garageLozenec 18 Sofia false 3 6
create RentOffer aptLozenec24 750
create SaleOffer aptLozenec24 195000
create RentOffer aptLozenec24 720
create SaleOffer officeVitosha44 96000
create RentOffer officeVitosha44 720
create RentOffer officePlovdiv 450
create SaleOffer houseBankya 320000
create RentOffer houseBankya 950
create RentOffer garageLozenec 100
create RentOffer garageLozenec 120
create SaleOffer garageLozenec 12000
create SaleOffer garageLozenec 11000
create RentOffer garageLozenec 720
```

```
create RentOffer invalid
create SaleOffer negative_price -5000
```

status

```
find-sales-by-location Sofia
find-rents-by-location Sofia
find-rents-by-price 700 1000
find-rents-by-price 0 99
```

```
invalid command
find-sales-by-location
```

find-rents-by-price 0

Note: the engine is designed to skip all empty input lines.

Sample Output

Apartment created.

Apartment created.

Invalid command.

Invalid command.

Invalid command.

Invalid command.

Invalid command.

Estates:

Apartment: Name = aptLozenec24, Area = 150, Location = Sofia, Furnitured = Yes, Rooms: 4, Elevator: Yes

Apartment: Name = aptBotev28, Area = 54, Location = Sofia, Furnitured = Yes, Rooms: 2, Elevator: No

No offers

Office created.

Office created.

House created.

House created.

Garage created.

RentOffer created.

SaleOffer created.

RentOffer created.

SaleOffer created.

RentOffer created.

RentOffer created.

SaleOffer created.

RentOffer created.

RentOffer created.

SaleOffer created.

SaleOffer created.

RentOffer created.

Invalid command.

Invalid command.

Estates:

Apartment: Name = aptLozenec24, Area = 150, Location = Sofia, Furnitured = Yes, Rooms: 4, Elevator: Yes

Apartment: Name = aptBotev28, Area = 54, Location = Sofia, Furnitured = Yes, Rooms: 2, Elevator: No

Office: Name = officeVitosha44, Area = 70, Location = Sofia, Furnitured = Yes, Rooms: 1, Elevator: No

Office: Name = officePlovdiv, Area = 44, Location = Plovdiv, Furnitured = No, Rooms: 1,

Elevator: Yes

House: Name = houseBankya, Area = 206, Location = Bankya, Furnitured = Yes, Floors: 3

House: Name = houseSofia, Area = 120, Location = Sofia, Furnitured = Yes, Floors: 1

Garage: Name = garageLozenec, Area = 18, Location = Sofia, Furnitured = No, Width: 3,
Height: 6

Offers:

Rent: Estate = aptLozenec24, Location = Sofia, Price = 750

Sale: Estate = aptLozenec24, Location = Sofia, Price = 195000

Rent: Estate = aptLozenec24, Location = Sofia, Price = 720

Sale: Estate = officeVitosha44, Location = Sofia, Price = 96000

Rent: Estate = officeVitosha44, Location = Sofia, Price = 720

Rent: Estate = officePlovdiv, Location = Plovdiv, Price = 450

Sale: Estate = houseBankya, Location = Bankya, Price = 320000

Rent: Estate = houseBankya, Location = Bankya, Price = 950

Rent: Estate = garageLozenec, Location = Sofia, Price = 100

Rent: Estate = garageLozenec, Location = Sofia, Price = 120

Sale: Estate = garageLozenec, Location = Sofia, Price = 12000

Sale: Estate = garageLozenec, Location = Sofia, Price = 11000

Rent: Estate = garageLozenec, Location = Sofia, Price = 720

Query Results:

[Estate: aptLozenec24, Location: Sofia, Price: 195000]

[Estate: garageLozenec, Location: Sofia, Price: 12000]

[Estate: garageLozenec, Location: Sofia, Price: 11000]

[Estate: officeVitosha44, Location: Sofia, Price: 96000]

Query Results:

[Estate: aptLozenec24, Location: Sofia, Price: 750]

[Estate: aptLozenec24, Location: Sofia, Price: 720]

[Estate: garageLozenec, Location: Sofia, Price: 100]

[Estate: garageLozenec, Location: Sofia, Price: 120]

[Estate: garageLozenec, Location: Sofia, Price: 720]

[Estate: officeVitosha44, Location: Sofia, Price: 720]

Query Results:

[Estate: aptLozenec24, Location: Sofia, Price: 720]

[Estate: garageLozenec, Location: Sofia, Price: 720]

[Estate: officeVitosha44, Location: Sofia, Price: 720]

[Estate: aptLozenec24, Location: Sofia, Price: 750]

[Estate: houseBankya, Location: Bankya, Price: 950]

No Results

Invalid command.

Invalid command.

Invalid command.