D0017D Examination Assignment: LTU Real Estate

June 2, 2022

1 Assignment description

Your task is to implement a bidding system for real estate. The program is supposed to handle information about different real estate objects, as well as the bidding process (see instructions).

Menu At the start, the program should print a menu of options that can be selected by the user. It should be possible to exit the program by pressing `q`

```
----------------------------------
# LTU REAL ESTATE
----------------------------------

1. Register new object
2. Register bid
3. End bidding process
4. Print bidding history for object
5. Print all unsold objects
6. Print all sold objects (by price)
q. End program
> Enter your option:
```

Registering an object An object is registered by providing its address, type, as well as its asking price. In the system, it is identified by using a unique randomized ID (in range 1000-9999). Make sure that multiple objects do not end up with the same ID number.

```
> Enter your option: 1
> Enter property's address: Ursviksgatan 2
> Enter property's type (Apartment, House or Commercial): Appartment
> Enter starting price: 1500000

Object ID: 5494
```

Bidding registration Once an object is registered, it should be possible to bid on the property. An object is referenced by the ID number which was generated in previous step. A bid should be higher than the current highest bid, but it can be lower than the asking price. Bidder's name should be registered by the system. Note that the system should provide an error message if there is no object with provided ID.

```
> Enter your option: 2
> Enter object's ID number: 5494
> Enter bidder's name: Maxim
> Enter bid: 1400000
Bid registered!
```

```
> Enter your option: 2
> Enter object's ID number: 5494
> Enter bidder's name: Erik
> Enter bid: 1300000
There is a higher bid present (1400000 by Maxim)! Could not register the bid.
```

Ending of bidding process Once the seller is satisfied with the highest bid, it should be possible to end the bidding process. In order to be able to end the bidding process, there should be at least one (1) bid present for the object. Note: it should be possible to abort the ending of the bidding process by typing ”No”.

```
> Enter your option: 3
> Enter object's ID number: 5494
> Accept bid by Maria (2000000)?: Yes
Bid accepted, object is sold!
```

Printing of bidding history It should be possible to print bidding history of an object. If the object is sold, the accepted bid should be highlighted in some way:

```
> Enter your option: 4
> Enter object ID: 5494

Bidder      Price       Accepted
Maxim       1400000
Sandeep     1600000
Maxim       1700000
Maria       2000000     Yes
```

Printing all unsold objects It should be possible to print all unsold objects:

```
> Enter your option: 5
ID      Address         Type        Asking price    Highest Price
1224    Bogatan 17      Appartment  800000          -
1002    Sveagatan 20    House       7000000         10000000
```

Printing all sold objects (sorted by highest price) Finally, there should be an option to print all sold objects. The output should be sorted by the selling price (lowest to highest).

```
> Enter your option: 6
ID      Address         Type        Asking price    Sold for
5495    Ursviksgatan 2  Appartment  1400000         2000000
4981    Odengaran 10    House       2000000         3100000
9001    Boviksbadet 1   Commercial  8000000         7100000
```

2 Instructions

The program should have the necessary error handling (ways of validating the input, such as checking that the price is a number, etc.). The program should under no circumstances crash when receiving incorrect input. It is allowed to store additional information, other than provided in the assignment description. You should create appropriate data structures to store the information. The programs that compile with the compilation error will not be graded (it should be possible to run the program).

3 Tips

- Do one functionality/method/option at a time. It is possible to pass the assignment by not finalizing all the parts! Leave the hardest parts till the end. For example, First get the program working for correct/valid inputs. Then update the program to handle error conditions, such as checking if a number is entered for price.

- Write down requirements for each function as comment on top of the function, this will help you not to forget a requirement

4 Assumptions

If you notice that some information is missing from the assignment description, you are allowed to make own assumptions as long as it does not change the condition and basis of the assignment.

5 Requirements

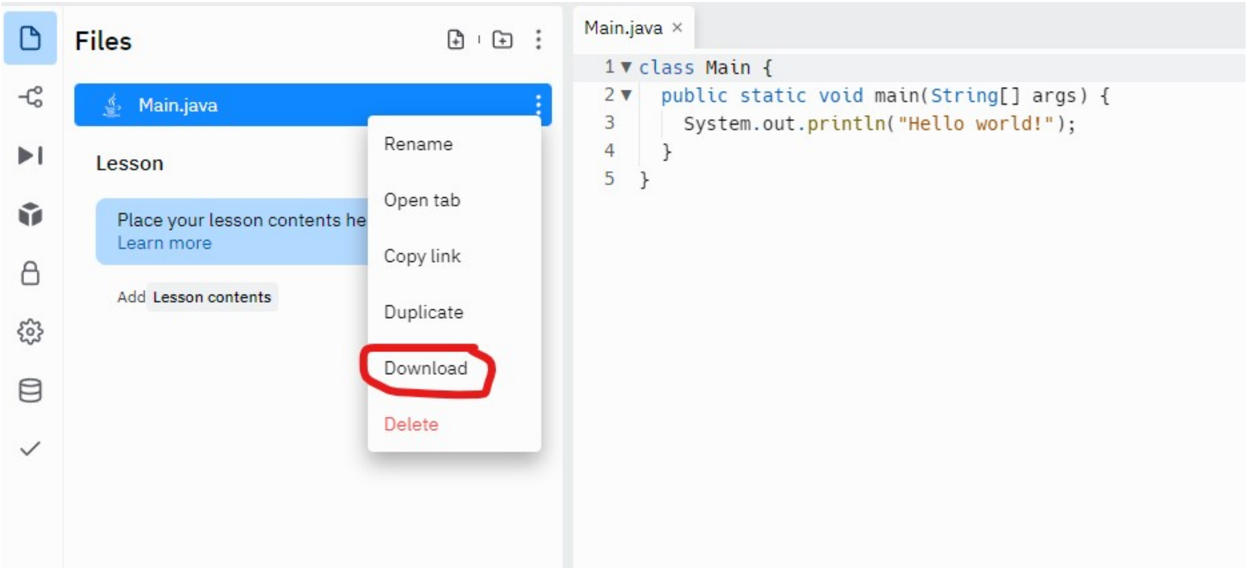
- ***Please do not forget to include your name, LTU username on the top of the file as a author***
- You are free to use either replit or any IDE of your choice.
- Do not use packages (there should not be any code, beginning with `package`).
- Follow the course's coding conventions and formatting conventions (method comments, usage of constants, indentation etc.).
- Use methods in your solution.
- It is not allowed to use any global variables (only global Scanner objects are allowed).
- It is not allowed to use built in functionality (such as ArrayLists etc.), in other words, same rules as your assignments. In other words, only import statement you will have is `import java.util.Scanner;`

The assignment is to be solved independently! It is allowed to get inspiration from the Internet, but you are to solve the task by your own! Always provide a source for your information that you get from external sources. **NOTE! It is not allowed to copy entire code sections!** You are, however, allowed to copy the code, that you have written yourself.

6 Submission

You must submit a single java file to canvas.

- If you are using replit, under the file tree on the left, click on the three dots and download the file. It will be downloaded as "'Main.java"'. Upload this single file to Canvas. You should also click on submit on replit. No need to submit the link on canvas.
- If you are using your own IDE, such as eclipse, make sure your filename is called "'Main.java"' and class name is called "'Main"'. Upload this single file Main.java to canvas.
- You can upload multiple times to Canvas.



```
class Main {
  public static void main(String[] args) {
    System.out.println("Hello world!");
  }
}
```

Figure 1: How to download file from replit