GIT Department of Computer Engineering
CSE 222/505 - Spring 2022
Homework #3 Report

Berkan AKIN
171044073

1. SYSTEM REQUIREMENTS

First of all, it has to be software. The software should simulate a street town. The software should be able to add and remove buildings on the street. If the area is full on this street, the building should not be built. Buildings to be built are Market House, Office and Playground. Except for the Playground, all others must take height. Each building has its own characteristics.

Features of the building have the number of rooms, color, and owner.

Office must-have features have job-type and owner.

The properties that the market should have, have owner and opening/closing times properties.

The software can process these buildings in 2 modes.

1) Edit Mode,

Buildings can be added and properties of buildings can be entered in Edit Mode. These buildings can be removed.

2) ViewMode.

- o display the total remaining length of lands on the street.
- he display the list of buildings on the street.
- o display the number and ratio of lenth of playgrounds in the street.
- he calculates the total length of street occupied by the markets, houses or offices. or offices. Display the skyline silhouette of the street .

.

Application should work for 4 data types ArrayList, LinkedList LDLinkedList and Array.
Algorithm analysis should be done for application 4 data types.

2. PROBLEM SOLUTION APPROACH

Regarding my system's requirements and problems, i created a container class to keep and modift the data easily. Then I was able to set up a hierarchy and find a solution, by correctly determining the class relationships and the ease provided by my container class.

PROBLEM SOLUTION APPROACH My Problem solution steps are;

1. – Specify the problem requirements
2. – Analyze the problem
3. – Design an algorithm and Program
4. – Implement the algorithm
5. – Test and verify the program
6. – Maintain and update the program

TEST CASES

Create every building (House,Market, Office Building)

```
House house1 = new House(10,2,5,"blue","Berkan"); // lenght,height,roomNumber, color, owner
Market market1 = new Market(15,4,10,30,18,30,"Ahmet");
Office office1 = new Office(10,4,"Software","Hatice");

House house2 = new House(10,2,5,"blue","Bilal"); // lenght,height,roomNumber, color, owner
Market market2 = new Market(15,4,10,30,18,30,"Mahmut");
Office office2 = new Office(10,4,"Software","Ferhat");
Playground pg = new Playground(10,"Hamdi");
```

Created four data type street class; I use four data type in Street class;

```
StreetArray streetArray = new StreetArray(100);
StreetArrayList streetArrayList = new StreetArrayList(100);
StreetLinkedList streetLinkedList = new StreetLinkedList(100);
StreetLDLinkedList streetLDLinkedList = new StreetLDLinkedList(100);
```

City Building Test

I tested four data type. City constructer take all data type street class. All results same for all data type.

You should open only one of the comment lines for City and test it.

```
City city = new City(streetArray);
//City city = new City(streetArrayList);
//City city = new City(streetLinkedList);
//City city = new City(streetLDLinkedList);
```

```
//City city = new City(streetArray);
City city = new City(streetArrayList);
//City city = new City(streetLinkedList);
//City city = new City(streetLDLinkedList);
```

```
//City city = new City(streetArray);
//City city = new City(streetArrayList);
City city = new City(streetLinkedList);
//City city = new City(streetLDLinkedList);
```

```
//City city = new City(streetArray);
//City city = new City(streetArrayList);
//City city = new City(streetLinkedList);
City city = new City(streetLDLinkedList);
```

Building add on street test

```
    city.getStreet().addBuildingSide1(house1, 0);
    city.getStreet().addBuildingSide1(market1, 20);
    city.getStreet().addBuildingSide1(office1, 50);
    city.getStreet().addBuildingSide1(pg, 80);

    city.getStreet().addBuildingSide2(house2, 0);
    city.getStreet().addBuildingSide2(market2, 20);
    city.getStreet().addBuildingSide2(office2, 50);
```

Building delete Test

```
city.getStreet().printBuildings();
System.out.println("------------------------------------------------------------");
System.out.println("Delete Test");
city.getStreet().deleteBuildings("Berkan");
city.getStreet().deleteBuildings("Mahmut");
```

Edit mode add test

```
System.out.println("------------------------------------------------------------");
System.out.println("Edit Mode Test");
EditMode emode = new EditMode(city); // EditMode Test
emode.addBuildingHouse("Alp", "blue", 5, 15, 5, 1, 70);
emode.addBuildingMarket("Ali", 10, 30, 18, 30, 5, 2, 3, 45);
emode.addBuildingOffice("Burak", "Software", 4, 3, 2, 90);
emode.addBuildingPlayground("Gazi",10, 20, 30);
```

Edit mode remove test

```
System.out.println("Edit Delete Test");
emode.removeBuilding("Ali");
emode.removeBuilding("Burak");
emode.removeBuilding("Hasan");  // there arent in street
```

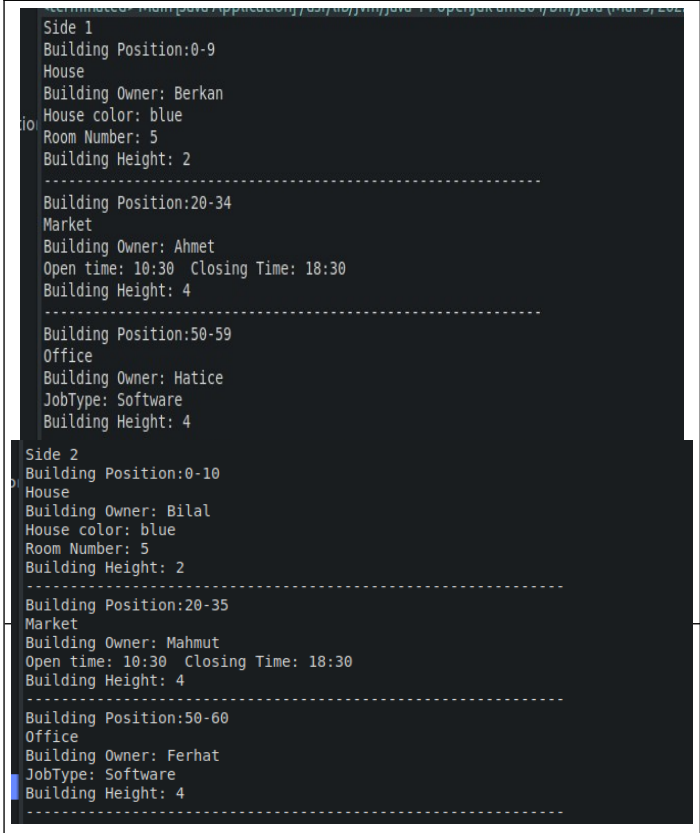
View mode test

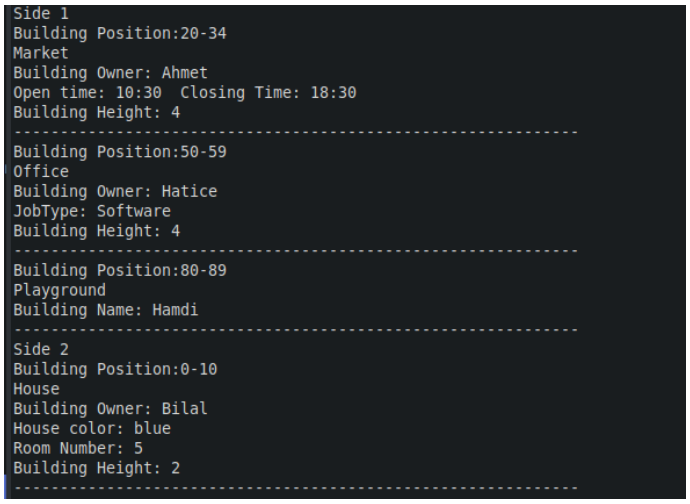
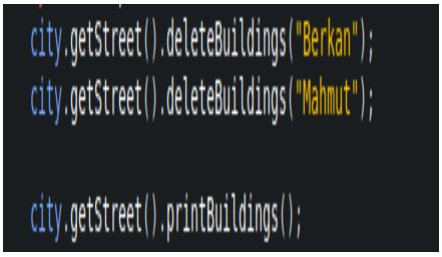
- display the total remaining length of lands on the street.
- display the list of buildings on the street.
- display the number and ratio of lenth of playgrounds in the street.
- calculate the total length of street occupied by the markets, houses or offices.
- display the skyline silhouette of the street (Please see the figure below).

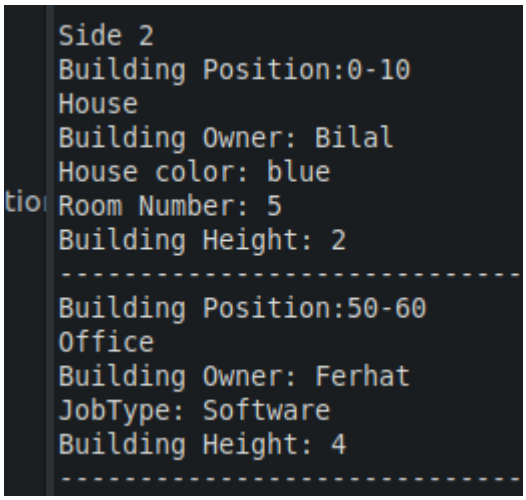
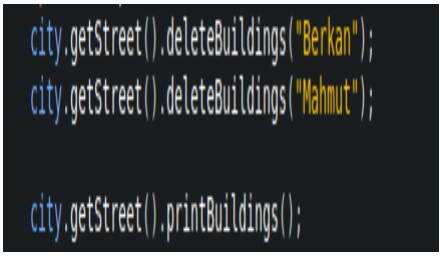
```
vmode.printListofStreet();
vmode.printRatioOfPlayground();
vmode.printRemaingLenght();
vmode.printStreetAccupied();
vmode.printSilhouette();
```

3. RUNNING AND RESULTS

Since the results are the same for the 4 data types of the test cases, I put them only once.

Output	Test	Results
	Add Buildng Side 1	Pass
	Add Buildng Side 2	Pass

	Delete Building Test 	Pass

	Delete Building Test 	Pass

<table>
<tr>
<td>

```
Edit Mode Test
The field is Full
Side 1
Building Position:20-34
Market
Building Owner: Ahmet
Open time: 10:30  Closing Time: 18:30
Building Height: 4
-------------------------------------
Building Position:50-59
Office
Building Owner: Hatice
JobType: Software
Building Height: 4
-------------------------------------
Building Position:80-89
Playground
Building Name: Hamdi
-------------------------------------
```

</td>
<td>Edit mode add test

```
System.out.println("-------------------------------------------------------");
System.out.println("Edit Mode Test");
EditMode emode = new EditMode(city); // EditMode Test
emode.addBuildingHouse("Alp", "blue", 5, 15, 5, 1, 70);
emode.addBuildingMarket("Ali", 10, 30, 18, 30, 5, 2, 3, 45);
emode.addBuildingOffice("Burak", "Software", 4, 3, 2, 90);
emode.addBuildingPlayground("Gazi",10, 20, 30);
```

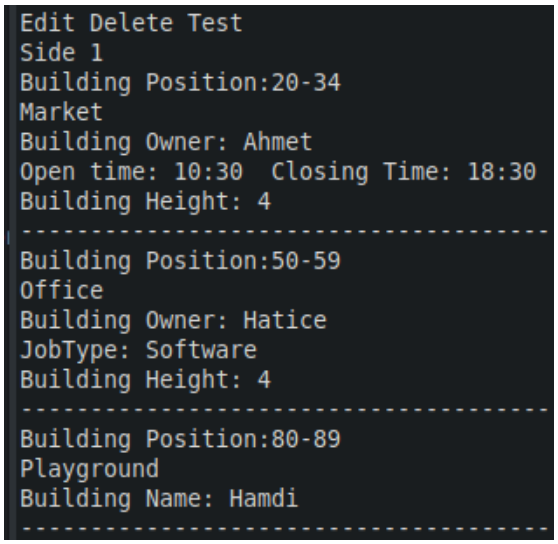
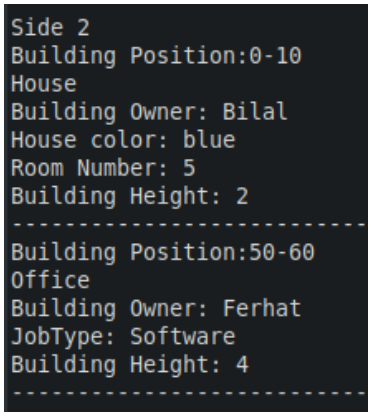
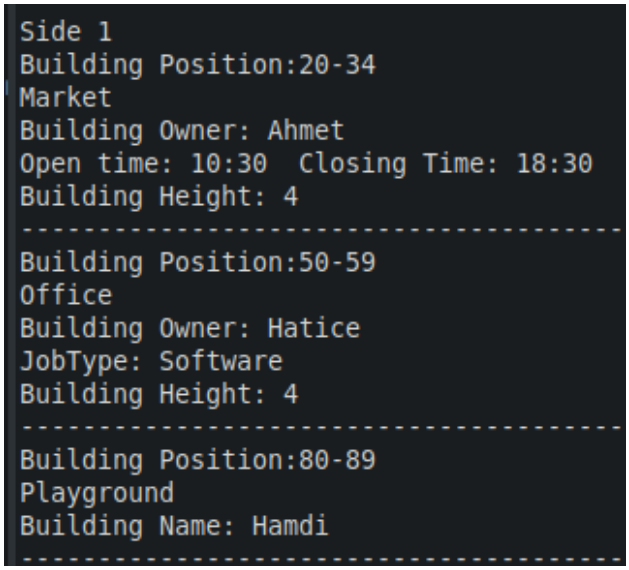
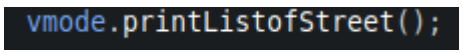
</td>
<td>Pass</td>
</tr>
<tr>
<td>

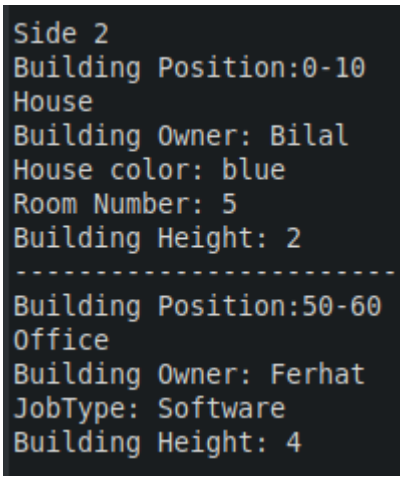
```
Side 2
Building Position:0-10
House
Building Owner: Bilal
House color: blue
Room Number: 5
Building Height: 2
-------------------------
Building Position:50-60
Office
Building Owner: Ferhat
JobType: Software
Building Height: 4
-------------------------
Building Position:90-94
Office
Building Owner: Burak
JobType: Software
Building Height: 3
-------------------------
```

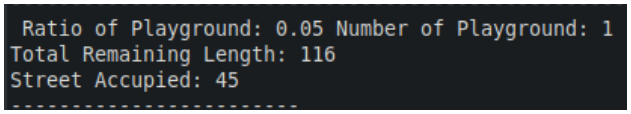
</td>
<td>Edit mode add test

```
System.out.println("-------------------------------------------------------");
System.out.println("Edit Mode Test");
EditMode emode = new EditMode(city); // EditMode Test
emode.addBuildingHouse("Alp", "blue", 5, 15, 5, 1, 70);
emode.addBuildingMarket("Ali", 10, 30, 18, 30, 5, 2, 3, 45);
emode.addBuildingOffice("Burak", "Software", 4, 3, 2, 90);
emode.addBuildingPlayground("Gazi",10, 20, 30);
```

Ali did not appear in the system because his field was full. There is another building</td>
<td>Pass</td>
</tr>
</table>

	Delete Building Test  Hasan There arent in street	Pass
	Delete Building Test 	pass
	View mode print all buildings 	Pass

	View mode print all buildings `vmode.printListofStreet();`	Pass

	View Mode Test Number and Ratio Playground, Remain Lenght(emty area), Street Acupied Area
	`vmode.printRatioOfPlayground();`
	`vmode.printRemaingLenght();`
	`vmode.printStreetAccupied();`
		Pass

*Class Diagram is in zip File.