

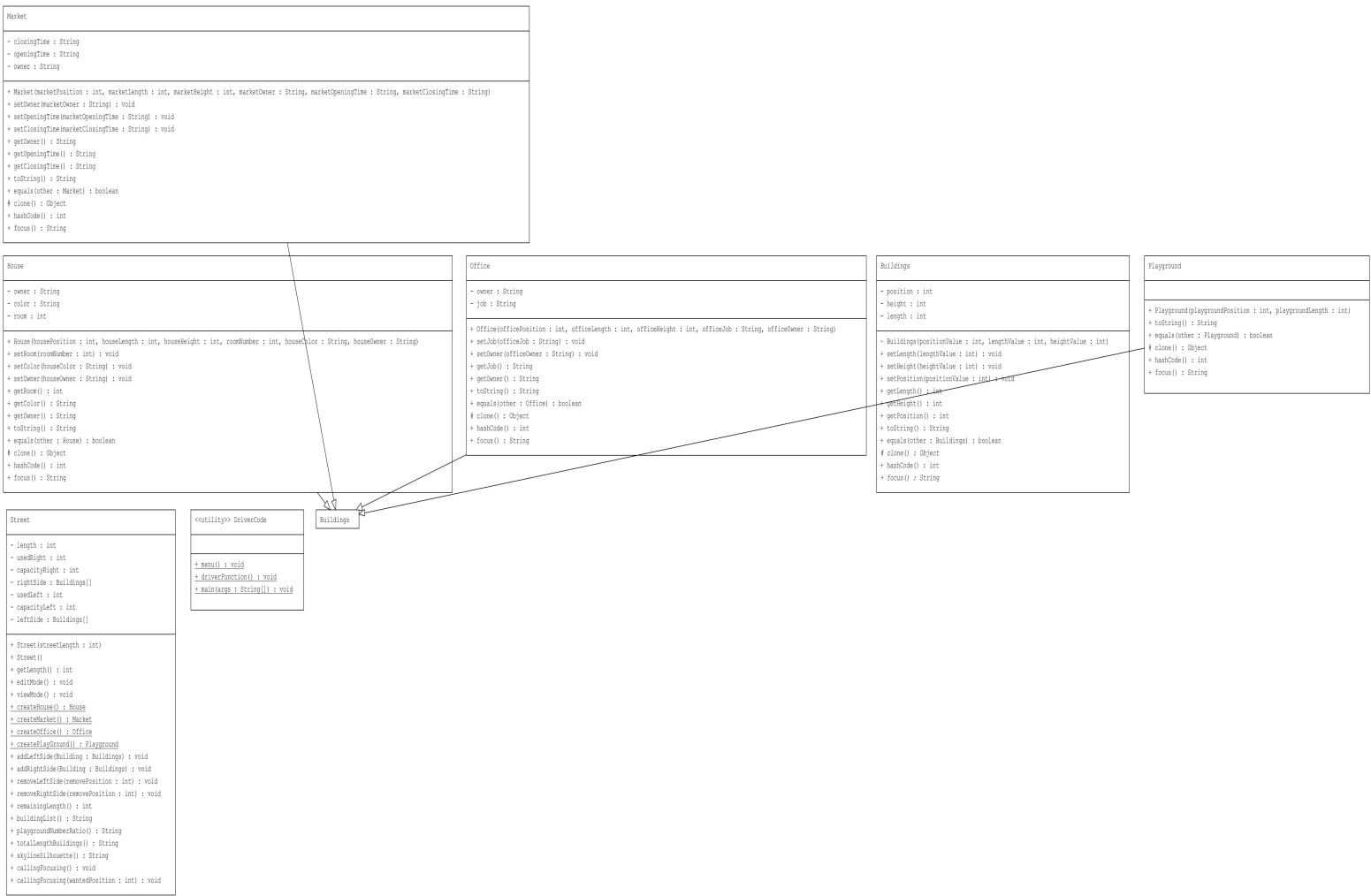
GTU Department of Computer Engineering
CSE 222/505 - Spring 2022
Homework 1 Report

Atacan Başaran
200104004008

1. SYSTEM REQUIREMENTS

This program is a city planning software for designing a small street. We need some designated buildings. Every building has to have something unique to look different. Then we need a street where we will place these created buildings. This street should have a separate structure for the left and right and the capacity to hold the data in these structures. It should be able to add and remove buildings from the street and look at the data of the buildings within it. This information includes the characteristics of the buildings, the appearance of the street, the places on the street, which building occupies how much space, etc. Also a menu is needed where the user can add and remove buildings and see information by guiding him.

2. CLASS DIAGRAM



3. PROBLEM SOLUTION APPROACH

First of all, I had to keep my approach at a basic level because I couldn't use anything different from an array. At first, I decided which classes I needed to create the program by drawing with paper and pencil. I extended the specified buildings from the buildings class. Of course, I filled these buildings with the necessary data and the necessary setter, getter/constructor etc. methods. I designed the street where I will place these buildings as a separate class. I divided the implementations for the left and right of the street (ex. addingLeftSide addingRightSide). I implemented the methods required by the edit mode and the view mode. I also created separate methods for these modes that will contact the user. My DriverCode class contains my driver method and this menu that I am talking about.

3. TEST CASES

Creating a street

```
Street testStreet = new Street(100);
```

Filling left of the street with 4 different building type (Buildings that cannot be added are shown as error output.)

```
testStreet.addLeftSide(new House(0, 5, 3, 4, "mavi", "Atacan"));
testStreet.addLeftSide(new Market(7, 10, 5, "Mehmet", "08.00", "19.00"));
testStreet.addLeftSide(new Office(20, 4, 20, "Textile", "Nurettin"));
testStreet.addLeftSide(new Playground(40, 30));
testStreet.addLeftSide(new House(15, 5, 5, 3, "pembe", "Ayşe"));
```

Filling right of the street with 4 different building type (Buildings that cannot be added are shown as error output.)

```
testStreet.addRightSide(new House(0, 5, 3, 4, "mor", "Elif"));
testStreet.addRightSide(new Market(7, 10, 5, "Hikmet", "06.00", "21.00"));
testStreet.addRightSide(new Office(20, 4, 20, "Freelancer", "Bedirhan"));
testStreet.addRightSide(new Playground(40, 30));
testStreet.addRightSide(new House(15, 5, 5, 3, "pembe", "Ayşe"));
```

Show data in view mode

```
System.out.printf("\nTotal remainig length of street: %d", testStreet.remainingLength());

System.out.println(testStreet.buildingList());

System.out.println(testStreet.playgroundNumberRatio());

System.out.println(testStreet.totalLengthBuildings());

System.out.println(testStreet.skylineSilhouette());

testStreet.callingFocusing(3);
testStreet.callingFocusing(150);
```

Removing buildings on the left of the street by position(non-removable buildings are shown as error output).

```
testStreet.removeLeftSide(15);
testStreet.removeLeftSide(90);
```

Removing buildings on the right of the street by position(non-removable buildings are shown as error output).

```
testStreet.removeRightSide(50);
testStreet.removeRightSide(80);
```

Show data in view mode after removing buildings

```
System.out.printf("\nTotal remainig length of street: %d", testStreet.remainingLength());

System.out.println(testStreet.buildingList());

System.out.println(testStreet.playgroundNumberRatio());

System.out.println(testStreet.totalLengthBuildings());

System.out.println(testStreet.skylineSilhouette());

testStreet.callingFocusing(20);|
testStreet.callingFocusing(-5);
```

Testing some overrided object class method

```
House testing1 = new House(0, 5, 5, 3, "gri", "Emre");
House testing2 = new House(0, 5, 5, 3, "gri", "Emre");
House testing3 = new House(5, 10, 10, 3, "kahverengi", "Salih");
System.out.println("\nTesting equals method");
if (testing1.equals(testing2))
    System.out.println("Testing1 and testing2 house are equal each other");
if (!(testing1.equals(testing3)))
    System.out.println("Testing1 and testing3 house are not equal each other");

System.out.println("\nTesting hashCode method");
System.out.print("\nTesting1 hashCode: " + testing1.hashCode());
System.out.print("\nTesting2 hashCode: " + testing2.hashCode());
System.out.print("\nTesting3 hashCode: " + testing3.hashCode());
```

5. RUNNING AND RESULTS

Menu when program run.

Creating street and menu shown

```
Enter the length of street: 100  
  
-----MENU-----  
(1) Editing mode  
(2) View mode  
(0) EXIT  
Enter the INPUT: █
```

Editing mode

```
(1)Adding new building to street  
(2)Removing building in street  
Enter the INPUT: █
```

Adding building

```
Choose the side you want to do the operation:  
(1)Left Side  
(2)Right Side  
Enter the INPUT: █
```

After choosing side adding building type

```
-----BUILDING TYPE-----  
(1)House  
(2)Market  
(3)Office  
(4)Playground  
Enter the INPUT:1  
  
Enter starting position of house: 0  
  
Enter length of house: 10  
  
Enter height of house: 5  
  
Enter room number of house: 3  
  
Enter color of house: mor  
  
Enter owner of house: Atacan
```

Removing building

```
(1)Adding new building to street
(2)Removing building in street
Enter the INPUT:2

Choose the side you want to do the operation:
(1)Left Side
(2)Right Side
Enter the INPUT:1

Enter the position of the building you want to remove:0
```

View Mode

```
-----MENU-----
(1) Editing mode
(2) View mode
(0) EXIT
Enter the INPUT:2

(1)Total remainig length of street
(2)List of buildings(number and details)
(3)Number and Ratio Length of Playgrounds
(4)Total length of different buildings
(5)Skyline silhouette
(6)Focusing a building with starting position
Enter the INPUT: █
```

Total Remaining length of street

```
Enter the INPUT: 1

Total remainig length of street: 175
```

List of buildings

```
There are 1 House,1 Market,0 Office,1 Playground in street
-----Building details-----
Building starting position: 0
Building length: 5
Building height: 5
House room number: 3
House color: mor
House owner: Atacan
-----
Building starting position: 40
Building length: 10
Building height: 5
-----
Building starting position: 0
Building length: 10
Building height: 5
Market owner: Mehmet
Market working times: 17.00 - 19.00
-----
```

Number and Ratio length of playgrounds

```
Enter the INPUT: 3

Occupied length by Playgrounds: 10
Ratio of length of Playgrounds in street: %5.0
```

Total length of different buildings

```
Enter the INPUT: 4

Occupied length by House: 5
Occupied length by Market: 10
Occupied length by Office: 0
```

Focusing a building

```
Enter the INPUT: 6

Enter the position of the building you want to focus on: 10

Building details on right side:
Building starting position: 0
Building length: 10
Building height: 5
Market owner: Mehmet
Market working times: 17.00 - 19.00

Focusing results on right side: 19.00
```