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

Ahmet Kadir Aksu
200104004114

	ArrayList	Linked List	LDLinkedList	Array
add()	O(n)	O(n^2)	O(n^2)	O(n)
delete(Building)	O(n)	O(n)	O(n)	O(n)
delete(int, String)	O(n^2)	O(n^2)	O(n^2)	O(n)
isSuitable()	O(n)	O(n^2)	O(n^2)	O(n)
listBuildings()	O(n)	O(n^2)	O(n^2)	O(n)
numberOfPlaygrounds()	O(n)	O(n^2)	O(n^2)	O(n)
PlaygroundRatio()	O(n)	O(n^2)	O(n^2)	O(n)
totalRemainings	O(n)	O(n^2)	O(n^2)	O(n)
totalLengths	O(n)	O(n^2)	O(n^2)	O(n)
displaySilhouette()	O(n^3)	O(n^4)	O(n^4)	O(n^3)
maxHeight()	O(n)	O(n^2)	O(n^2)	O(n)
toString()	O(n)	O(n^2)	O(n^2)	O(n)
equals()	O(1)	O(1)	O(1)	O(1)

LDLinkedList:

```
Execution time in nanoseconds / 100000: 1252
```

ArrayList:

```
Execution time in nanoseconds / 1000000: 79
```

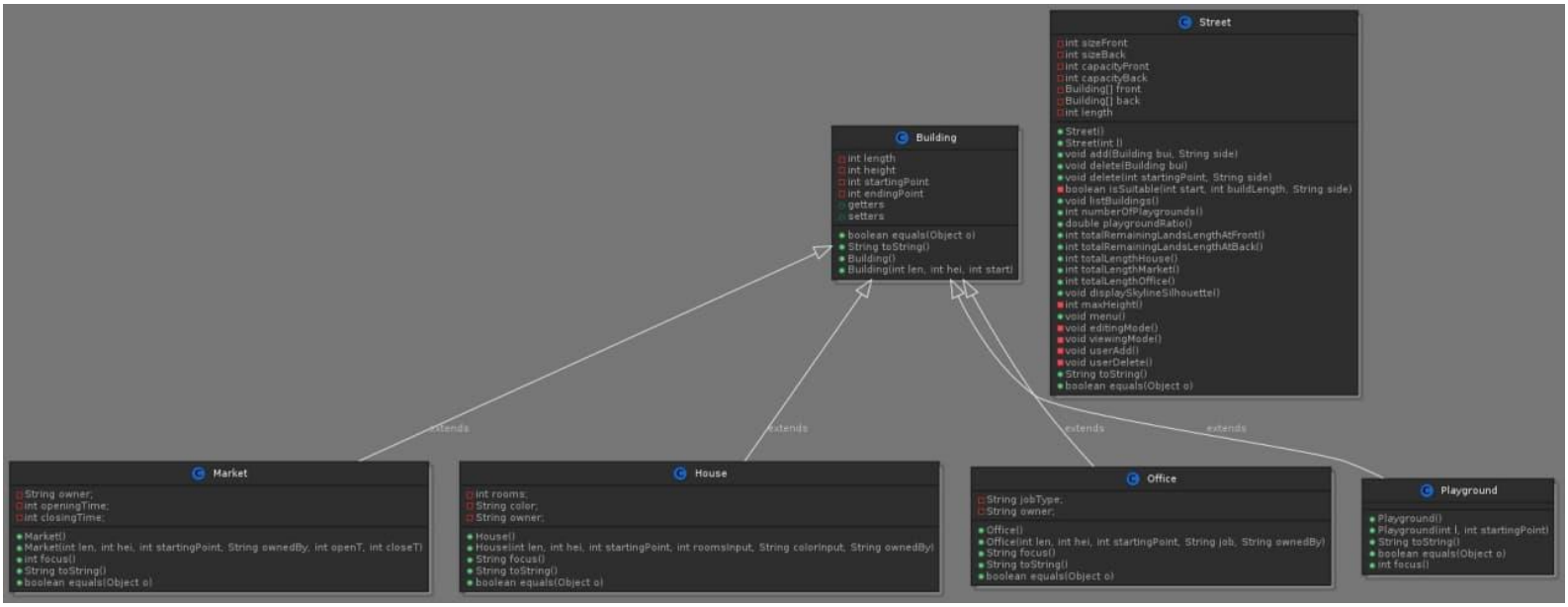
LinkedList:

```
Execution time in nanoseconds / 1000000: 74
```

1. **SYSTEM REQUIREMENTS**

The System is a city planning software, while the city has only one street.
So we need a street first, then buildings are needed (Markets, houses, playgrounds, offices).
Street length can be set at creation, but it is not possible to change it after.
There are add and delete option to edit the street by user.
There are options to view details of the buildings on the street.
There are 3 different kinds of the program with different data structures:
1-Using arrayList
2-Using LinkedList from the java Collection
3-Using LDLinkedList
Time complexities of all them shown above.

2. **USE CASE AND CLASS DIAGRAMS**



Building Class is a super class of Market, House, Office and Playground classes.

Street class needs building class for its implementations.

3. PROBLEM SOLUTION APPROACH

Here the problem is we need a city planning software. In this city there are some buildings which are markets, houses, offices. And there are also playgrounds in this street. So, it is obvious that we need to create classes for these buildings (From here on, playgrounds will be considered as a kind of building). Since these buildings have some common features as length, height or starting point, we can create a super class named building. We need a street class to keep these buildings.

User should be able to edit or view the details of the street. We add methods (for example: add, delete, or list buildings) to provide this to user. User can access these methods. User also can see the silhouette image of the street.

4. TEST CASES

Creates a new street.

```
Street myStreet = new Street(60);
```

Creates some buildings

```
House house1 = new House(8, 10, 6, 10, "Blue", "Hans");
Market market1 = new Market(12, 7, 45, "Kahn", 10, 19);
House house2 = new House(10, 15, 20, 4, "Black", "Osman");
House house3 = new House(8, 10, 6, 10, "Blue", "Hans");
```

Check equals method

```
System.out.println("Checking if house1 and house2 are equal: ");
System.out.println(house1.equals(house2));
```

```
System.out.println("Checking if house1 and house3 are equal");
System.out.println(house1.equals(house3));
```

```
System.out.println("Checking if house1 and house1 are equal");
System.out.println(house1.equals(house1));
```

Adds some buildings to the street.

```
System.out.println("Adding a house at the front side of the street");
myStreet.add(house1, "Front");
System.out.println("Adding a market at the back side of the street");
myStreet.add(market1, "Back");
System.out.println("Adding a playground at the front side of the street.");
myStreet.add(new Playground(7, 14), "Front");
System.out.println("Adding an office at the front side of the street.");
myStreet.add(new Office(10, 15, 25, "Software", "Aka Software"), "Front");
System.out.println("Adding a house at the back side of the street");
myStreet.add(new House(8, 10, 30, 5, "Black", "John"), "Back");
```

Lists buildings on the street

```
myStreet.listBuildings();
```

Displays silhouette of the street

```
myStreet.displaySkylineSilhouette();
```

Deletes the building which has a starting location at 14 and at front side of street

```
myStreet.delete(14, "Front");
```

Prints the total length of any building

```
System.out.println("Total length of the markets: "
                    + myStreet.totalLengthMarket());
System.out.println("Total length of the houses: "
                    + myStreet.totalLengthHouse());
System.out.println("Total length of the offices: "
                    + myStreet.totalLengthOffice());
```

Prints the empty lands on the street (front side and back side)

```
System.out.println("Total remaining length of lands");
System.out.println("At Front: "+ myStreet.totalRemainingLandsLengthAtFront());
System.out.println("At Back: "+ myStreet.totalRemainingLandsLengthAtBack());
```

Displays the number and ratio of length of playgrounds in the street.

```
System.out.println("There are " + myStreet.numberOfPlaygrounds()
                    + " playgrounds on the street");
System.out.println("Ratio is " + myStreet.playgroundRatio());
```

Exceptions

```
try{
    System.out.println("Trying to create a market with wrong opening hour");
    var Market = new Market(15, 15, 5, "Johnson", 25, 26);
} catch (Exception e){
    System.out.println(e);
}

    try{
        System.out.println("\nTrying to create a building with negative length");
        var Market = new Market(-5, 15, 5, "Johnson", 9, 12);
    } catch (Exception e){
        System.out.println(e);
    }

   try{
        System.out.println("\nTrying to delete a building with wrong input");
        var myStreet4 = new Street(50);
        myStreet4.add(new Office(20, 15, 10, "Coffee shop", "Osman"), "Front");
        myStreet4.delete(60, "Back");
   } catch(Exception e){
        System.out.println(e);
   }

try{
    System.out.println("\nTrying to add a building which is longer than the street"
    var myStreet3 = new Street(10);
    myStreet3.add(new House(15, 20, 5, 3, "Blue", "Hasan"), "Front");
} catch(Exception e){
    System.out.println(e);
}
```

```
try{
    System.out.println("\nTrying to create a street with negative length");
    var myStreet2 = new Street(-1);
}catch (Exception e){
    System.out.println(e);
}
```

Menu part

```
//Menu part
try{System.out.print("\n\nEnter the length of the street: ");
Scanner sc = new Scanner(System.in);
int len = sc.nextInt();                                                    );
var yourStreet = new Street(len);
yourStreet.menu();                                                         ;
sc.close();
} catch(Exception e){
    System.out.println(e);                                                 );
}
```

5- RUNNING AND RESULTS

Results of test cases

```
Checking if house1 and house2 are equal:
false
Checking if house1 and house1 are equal
true
Checking if house1 and house3 are equal
true
```

```
Adding a house at the front side of the street
Adding a market at the back side of the street
Adding a playground at the front side of the street.
Adding an office at the front side of the street.
Adding a house at the back side of the street

Buildings in the street:

Front :
1-House
2-Playground
3-Office

Back :
4-Market
5-House
```

```
Total remaining length of lands
At Front: 42
At Back: 40
```

```
There are 0 playgrounds on the street
Ratio is 0.0
Total length of the markets: 12
Total length of the houses: 16
Total length of the offices: 10
```

```
Deleting the playground building

Buildings in the street:

Front :
1-House
2-Office

Back :
3-Market
4-House
```

Menu results

```
Enter the length of the street: 80
----------------------------------------
Welcome to the City Planning Software
----------------------------------------


Please choose the mode you want to use
1-Editing mode
2-Viewing mode

0- Exit

Your choice: 1


1-add
2-delete

0-Back

Your choice: 1


Choose the building type you want to add
1-House
2-Office
3-Playground
4-Market
```

```
3-Playground
4-Market

0-Back

Your choice: 2
Enter the required informations about the office.
length: 13

height: 15

starting point: 10

job: barber shop

owner: fuat

Front(1) or back(2): 1


Choose the building type you want to add
1-House
2-Office
3-Playground
4-Market

0-Back
```

```
Your choice: 3
Enter the required informations about the playground.
length: 10

starting point: 15

Front(1) or back(2): 2


Choose the building type you want to add
1-House
2-Office
3-Playground
4-Market

0-Back

Your choice: 0


1-add
2-delete

0-Back

Your choice: 0
```

```
Please choose the mode you want to use
1-Editing mode
2-Viewing mode

0- Exit

Your choice: 2


1 - Display the remaining length of lands on the street
2 - Display the list of building on the street
3 - Display the number and ratio of lenth of playgrounds in the street
4 - Total length of street occupied by any building
5 - Display the silhouette of the street

0 - Back

Your choice: 5
```

```
Your choice: 5
          |-----------|
          |           |
          |           |
          |           |
          |           |
          |           |
          |           |
          |           |
          |           |
          |           |
          |           |
          |           |
          |           |
          |           |-|
          |             |
--------------------------------------------------------------------------------


1 - Display the remaining length of lands on the street
2 - Display the list of building on the street
3 - Display the number and ratio of lenth of playgrounds in the street
4 - Total length of street occupied by any building
5 - Display the silhouette of the street

0 - Back
```

```
Buildings in the street:

Front :
1-Office

Back :
2-Playground
----------------------------
Pick a building to delete
Your choice: 1


1-add
2-delete

0-Back

Your choice: 2

Buildings in the street:

Front :

Back :
1-Playground
----------------------------
```