

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

**Tuba Toprak**  
**161044116**

## 1. SYSTEM REQUIREMENTS

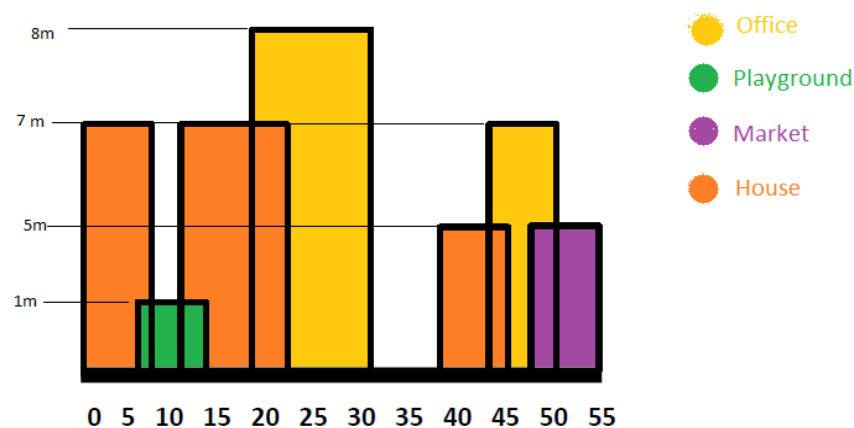
### User requirements and Functional Requirements

Homework 1 has been modified. In the assignment delivered as 3 separate files, the main assignment was implemented in 3 different ways as ArrayList, LinkedList, LDLinkedList.

ArrayList and LinkedList are taken from the Java Collections Framework.

LDLinkedList implemented the List interface and extend the AbstractList in Java Collections Framework. Abstract class methods are overridden. List interface methods are implemented. Necessary measurements were made. Complexity time was calculated and converted into a table.

When you start to run each file, the street is designed as follows and its silhouette image is printed.



Then, the add method in different sizes was measured and printed on the screen. It gives the same output every 3 scripts run. The lazy delete strategy was used in the Linked List implementation.

### Software requirements

This system is designed in IntelliJ idea. Version is java 15.0.2. The software runs on any operating system and IDE.

## 2. USE CASE AND CLASS DIAGRAMS

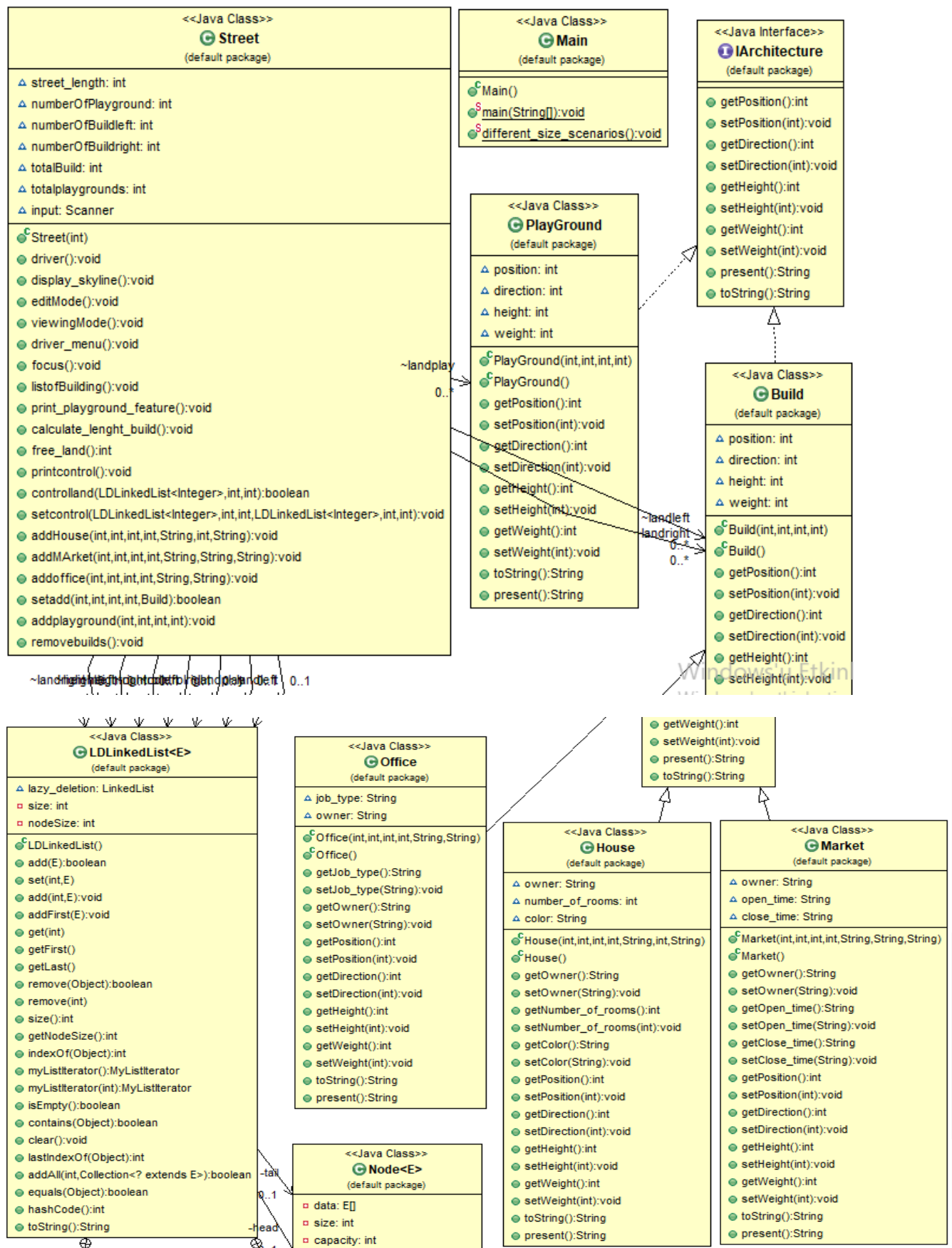
## ArrayList Class Diagram

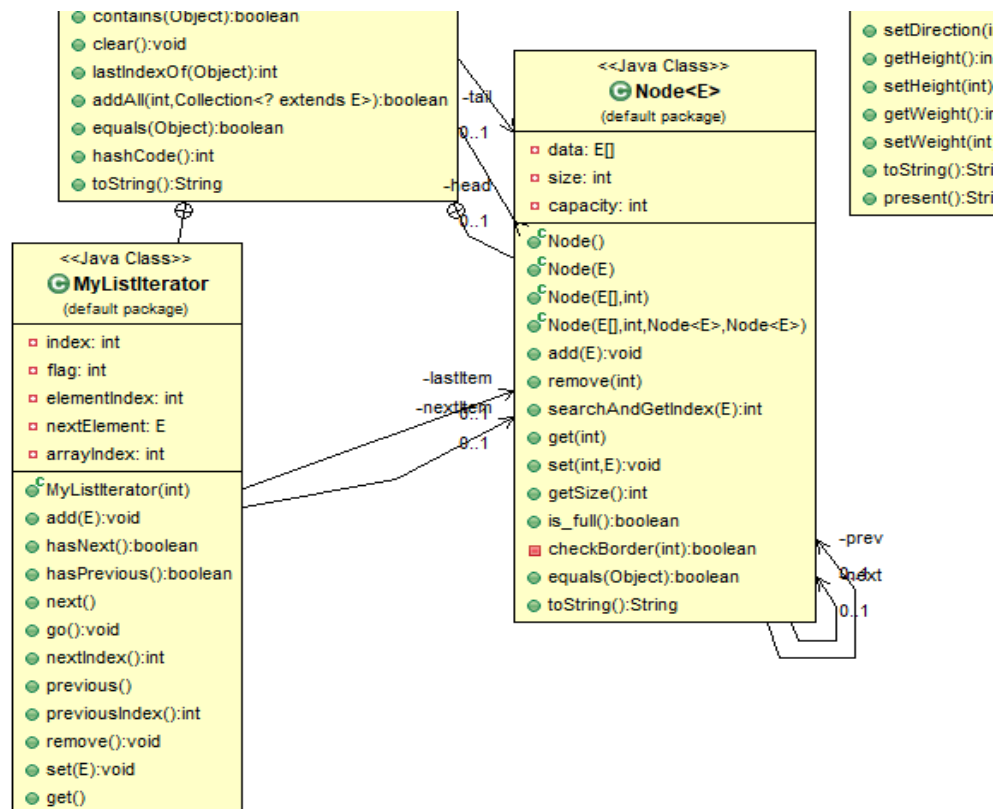


## LinkedList Class Diagram



## LDLinkedList Class Diagram



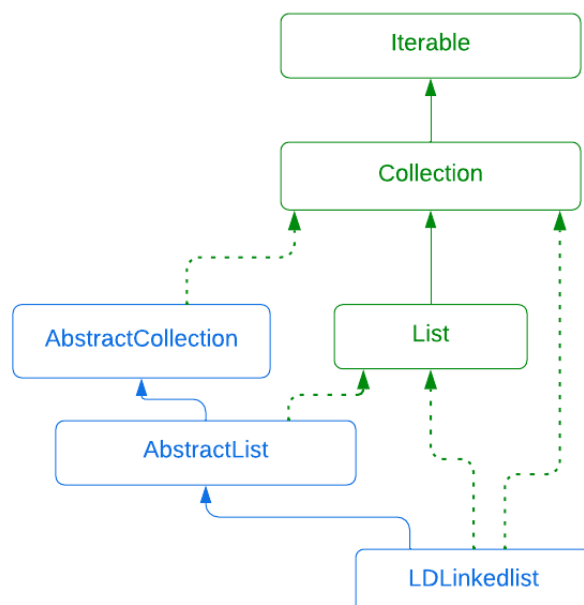


### 3. OTHER DIAGRAMS

### 4. PROBLEM SOLUTION APPROACH

Problem: How to implement LDLinkedList?

The LDLinkedList requested in the assignment was searched. First of all, list interface and abstractclass methods were examined to extend and implement. And it was looked at that the Abstractclass uses the LIST interface and how it is implemented.



Node class and iterator class are created as inner class in LDLinkedList.


applied methods in LDLinkedList:

1. add(E e): Appends the specified element to the end of this list (optional operation).
2. add(int index, E element): Inserts the specified element at the specified position in this list (optional operation).
3. addAll(int index, Collection c): Inserts all of the elements in the specified collection into this list at the specified position (optional operation).
4. clear(): Removes all of the elements from this list (optional operation).
5. equals(Object o): Compares the specified object with this list for equality.
6. get(int index): Returns the element at the specified position in this list.
7. hashCode(): Returns the hash code value for this list.
8. indexOf(Object o): Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element.
9. iterator(): Returns an iterator over the elements in this list in proper sequence.
10. lastIndexOf(Object o): Returns the index of the last occurrence of the specified element in this list, or -1 if this list does not contain the element.
11. listIterator(): Returns a list iterator over the elements in this list (in proper sequence).
12. listIterator(int index): Returns a list iterator over the elements in this list (in proper sequence), starting at the specified position in the list.
13. remove(int index): Removes the element at the specified position in this list (optional operation).
14. removeRange(int fromIndex, int toIndex): Removes from this list all of the elements whose index is between fromIndex, inclusive, and toIndex, exclusive.
15. set(int index, E element): Replaces the element at the specified position in this list with the specified element (optional operation).
- 16.

A linkedlist has been created for Lazy Deletion and deleted objects have been added to

it. `LinkedList lazy_deletion = new LinkedList<>();`

Experimental runtime was measured for the code generated in 3 different ways. For this, nano time measurement unit was used. When the code is run, the measured values are displayed after the image on the silhouette is printed on the screen. Measured values are in the output part. The values are compared in the table below.



Size->	100	1000	100000
Array	19500 times	25000	2627900
Arraylist	37199	166501	9917601
LinkedList	44500	364499	18215300
LDLinkedList	68600	272300	23809100

As seen in the measured values, the running time increases towards the direction of the red arrow.

Time Complexity	ArrayList	Linkedlist	LDLinkedlist	Array
Insert at last index	O(1)	O(1)	O(n)	O(n)
Insert at given index	O(n)	O(n)	O(n)	O(1)
Search by value	O(n)	O(n)	O(n)	O(1)
Get by index	O(1)	O(n)	O(n)	O(1)
Remove by value	O(n)	O(n)	O(n)	O(n)
Remove by index	O(n)	O(n)	O(n)	O(n)

## 5. TEST CASES

The same output is taken for 3 different implementations.

```
-----Welcome to StrangerVille Street-----
1-- Edit Mode
2-- View Mode
3-- Exit
Please enter choice: 4
Invalid Choice...
```

Passed

```
----Edit Mode----
1- Add Building
2- Remove Building
3- Back
Please enter choice: 2
Please select the direction you want to delete.
Please select the direction you want to focus on (0-Left Side,1- Right Side,2- Back): 3
Invalid Choice..
Please select the direction you want to delete.
Please select the direction you want to focus on (0-Left Side,1- Right Side,2- Back): |
```

Passed



```

Please enter choice: 1
----Edit Mode----
1- Add Building
2- Remove Building
3- Back
Please enter choice: 1
1- House
2- Market
3- Office
4- Exit
Please select type Building: 1
Please enter position: 0
Please select the direction (0-Left ,1- Right): 1
Please enter the height: 7
Please enter the lenght: 0
Please enter owner: Veli
Please enter number of rooms: 3
Please enter color: White
The area you want to build is not available.

```

adding a building to unoccupied land

delete a nonexistent building

```

----Edit Mode----
1- Add Building
2- Remove Building
3- Back
Please enter choice: 2
Please select the direction you want to delete on (0-Left Side,1- Right Side,2- Back): 1
---Buildings to the right of the street---
0. House--> position= 0, direction= Right, height= 7, weight= 8, owner= 'Tuba Toprak', number_of_rooms= 4, color= 'Pink'
1. House--> position= 12, direction= Right, height= 7, weight= 14, owner= 'Ali Agaoğlu', number_of_rooms= 5, color= 'White'
2. House--> position= 40, direction= Right, height= 5, weight= 5, owner= 'Han Derenoglu', number_of_rooms= 2, color= 'Black'
3. Market--> position= 50, direction= Right, height= 5, weight= 5, owner= 'Carrefoursa As.', open_time= '08:55', close_time= '23:55'
Please select the building you want to delete: 4
You entered the wrong index. The building could not be deleted.

```

Passed

```

----Edit Mode----
1- Add Building
2- Remove Building
3- Back
Please enter choice: 1
1- House
2- Market
3- Office
4- Exit
Please select type Building: 5
Invalid Choice..

```

```

----Viewing Mode----
1- display the total remaining length of lands on the street
2- display the list of buildings on the street.
3- display the number and ratio of lenth of playgrounds in the street.
4- calculate the total length of street occupied by the markets, houses or offices.
5- display the skyline silhouvette of the street
6- Focus Mode
7- Back
Please enter choice: 0
Invalid Choice...

```

## 6. RUNNING AND RESULTS

When the code is run in three different implementations, the outputs are as follows. The experimental runtime of the array implementation in the first assignment was measured.

```
A House has been added in the street.  
A House has been added in the street.  
A House has been added in the street.  
A Market has been added in the street.  
A Playground has been added in the street.  
A Office has been added in the street.  
A Office has been added in the street.
```

Experimental Running Time: 19476400 nano times

```
--Array Different Size Scenarios--
```

The experimental runtime of the **Arraylist** implementation was measured.

```
A House has been added in the street.
A House has been added in the street.
A House has been added in the street.
A Market has been added in the street.
A Playground has been added in the street.
A Office has been added in the street.
A Office has been added in the street.
```

Experimental Running Time: 17821800 nano times

### --Arraylist Different Size Scenarios--

The experimental runtime of the **LinkedList** implementation was measured.

```
A House has been added in the street.  
A House has been added in the street.  
A House has been added in the street.  
A Market has been added in the street.  
A Playground has been added in the street.  
A Office has been added in the street.  
A Office has been added in the street.
```

```
Experimental Running Time: 44153700 nano times
```

--Linkedlist Different Size Scenarios--

The experimental runtime of the **LDLinkedlist** implementation was measured.

```
A House has been added in the street.  
A House has been added in the street.  
A House has been added in the street.  
A Market has been added in the street.  
A Playground has been added in the street.  
A Office has been added in the street.  
A Office has been added in the street.
```

```
Experimental Running Time: 73694799 nano times
```

**Different Size Senerios:** Output of the values specified in the previous table

```
--Array Different Size Scenarios---  
Small Size: 19500  
Middle Size: 25000  
Large Size: 2627900  
  
Process finished with exit code 0  
|
```

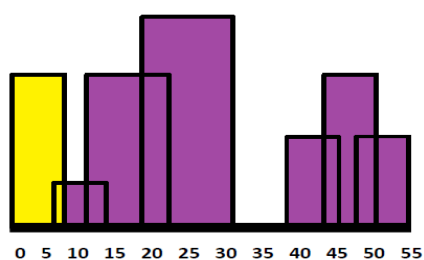
```
--LDLinkedList Different Size Scenarios---  
Small Size: 68600  
Middle Size: 272300  
Large Size: 23809100  
  
Process finished with exit code 0  
|
```

```
--LinkedList Different Size Scenarios---  
Small Size: 44500  
Middle Size: 364499  
Large Size: 18215300  
  
Process finished with exit code 0
```

```
--ArrayList Different Size Scenarios---  
Small Size: 37199 nanoTime  
Middle Size: 166501 nanoTime  
Large Size: 9917601 nanoTime  
  
Process finished with exit code 0
```

### Remove and Add Method LDLinkedList

The building painted in yellow in the picture below was first deleted and added later. The results are below. Then viewing mode was tried and the silhouette image was printed. The same output was obtained in all 3 implementations.



```
-----Welcome to StrangerVille Street-----
1-- Edit Mode
2-- View Mode
3-- Exit
Please enter choice: 1
-----Edit Mode-----
1- Add Building
2- Remove Building
3- Back
Please enter choice: 2
Please select the direction you want to delete on (0-Left Side,1- Right)
---Buildings to the right of the street---
0. House---> position= 0, direction= Right, height= 7, weight= 8, owner=
1. House---> position= 12, direction= Right, height= 7, weight= 14, owner=
2. House---> position= 40, direction= Right, height= 5, weight= 5, owner=
3. Market---> position= 50, direction= Right, height= 5, weight= 5, owner=
Please select the building you want to delete: 0
The building was demolished.
```

```
-----Edit Mode-----
1- Add Building
2- Remove Building
3- Back
Please enter choice: 1
1- House
2- Market
3- Office
4- PlayGround
5- Exit
Please select type Building: 1
Please enter position: 0
Please select the direction (0-Left ,1- Right): 1
Please enter the height: 7
Please enter the lenght: 8
Please enter owner: Mahmut Tuncer
Please enter number of rooms: 2
Please enter color: Red
A House has been added in the street.
1- House
2- Market
3- Office
4- PlayGround
5- Exit
Please select type Building:
```

```
4- calculate the total length of street occupied by the mar
5- display the skyline silhouvette of the street
6- Focus Mode
7- Back
Please enter choice: 5
```

