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

ÇAĞRI ÇAYCI
1901042629

1. SYSTEM REQUIREMENTS

The system provides to design a Street with both sides. The system support following types of Structures: House, Market, Office, and Playground. To create a Structure, some information about Structures is asked for user. For example, position and length is asked for user to create a Playground. If the user enters the information in correct type such as String for name, int for length; Structure is created.

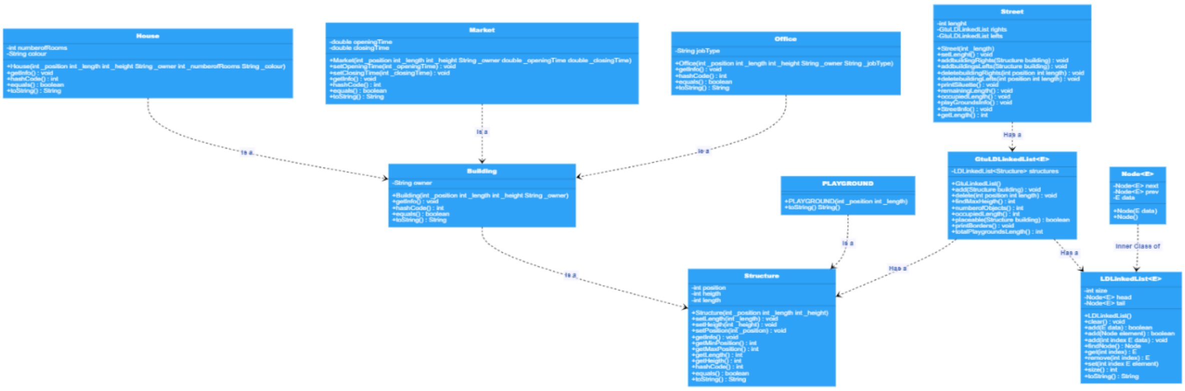
The system provides following features for designing a Street: adding a Structure to Street, removing a Structure from Street, showing information about Street (such as types of Structures on the Street, occupied length of the street etc.), printing the silhouette of the Street and more.

The methods which are developed for the system for performance rather than memory usage. For this reason, to increase the performance of the system more memory may be used.

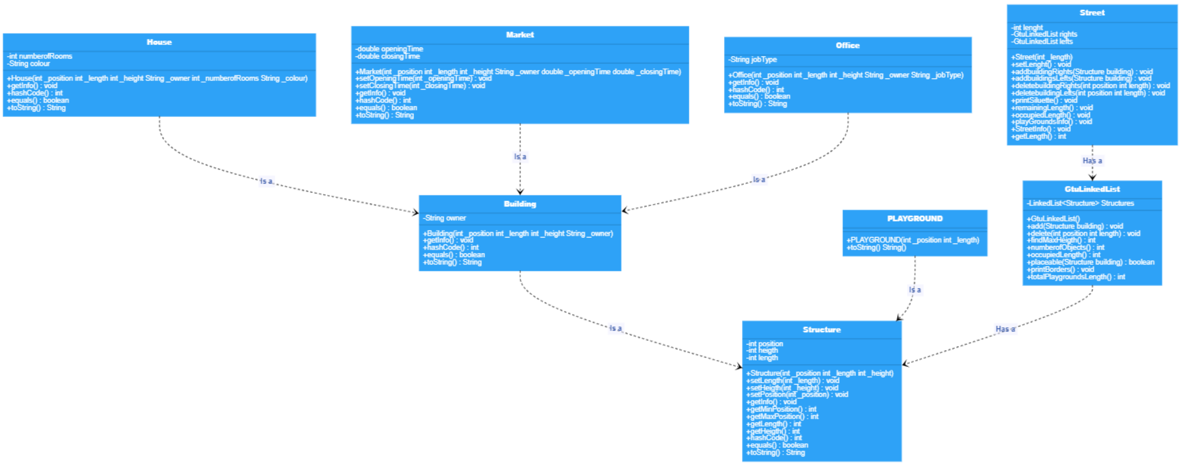
The methods which are developed for the system may throw an exception for some inputs. For example, the length of a structure cannot be less than 0. For this reason, when using this system, reasonable inputs should be used.

2. USE CASE AND CLASS DIAGRAMS

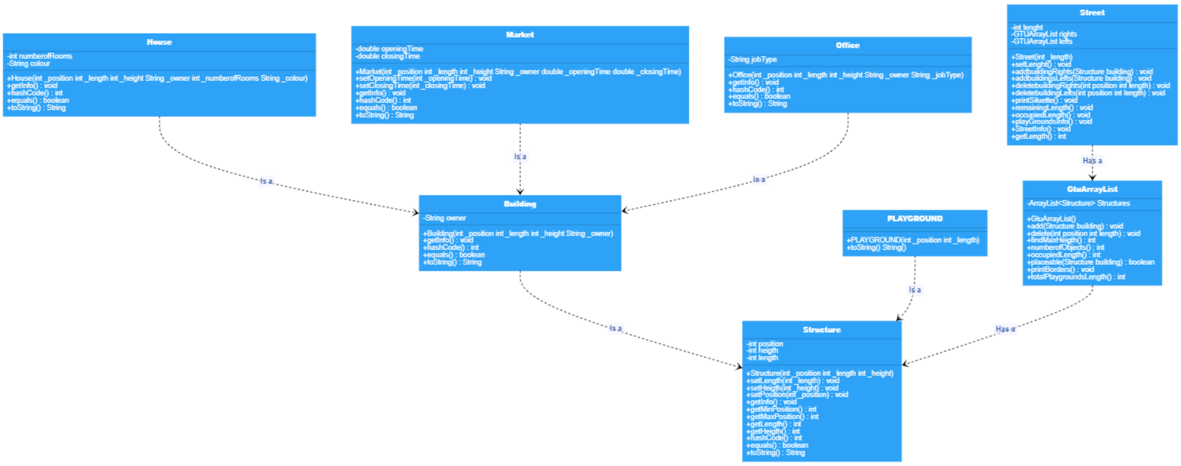
HW3_1



HW3_2


House
-int numberofRooms
-String colour
+House(int _position int _length int _height String _owner int _numberofRooms String _colour)
+getInfo() : void
+hashCode() : int
+equals() : boolean
+toString() : String
Market
-double openingTime
-double closingTime
+Market(int _position int _length int _height String _owner double _openingTime double _closingTime)
+setOpeningTime(int _openingTime) : void
+setClosingTime(int _closingTime) : void
+getInfo() : void
+hashCode() : int
+equals() : boolean
+toString() : String
Office
-String jobType
+Office(int _position int _length int _height String _owner String _jobType)
+getInfo() : void
+hashCode() : int
+equals() : boolean
+toString() : String
Street
-int lenght
-GtuLinkedList rights
-GtuLinkedList lefts
+Street(int _length)
+setLenght() : void
+addbuildingRights(Structure building) : void
+addbuildingsLefts(Structure building) : void
+deletebuildingRights(int position int length) : void
+deletebuildingLefts(int position int length) : void
+printSiluette() : void
+remainingLength() : void
+occupiedLength() : void
+playGroundsInfo() : void
+StreetInfo() : void
+getLength() : int
Is a
Is a
Is a
Has a
Building
-String owner
+Building(int _position int _length int _height String _owner)
+getInfo() : void
+hashCode() : int
+equals() : boolean
+toString() : String
PLAYGROUND
+PLAYGROUND(int _position int _length)
+toString() String()
GtuLinkedList
-LinkedList<Structure> Structures
+GtuLinkedList()
+add(Structure building) : void
+delete(int position int length) : void
+findMaxHeigth() : int
+numberofObjects() : int
+occupiedLength() : int
+placeable(Structure building) : boolean
+printBorders() : void
+totalPlaygroundsLength() : int
Is a
Is a
Has a
Structure
-int position
-int heigth
-int length
+Structure(int _position int _length int _height)
+setLength(int _length) : void
+setHeigth(int _height) : void
+setPosition(int _position) : void
+getInfo() : void
+getMinPosition() : int
+getMaxPosition() : int
+getLength() : int
+getHeigth() : int
+hashCode() : int
+equals() : boolean
+toString() : String

HW3_3


House
-int numberofRooms
-String colour
+House(int _position int _length int _height String _owner int _numberofRooms String _colour)
+getInfo() : void
+hashCode() : int
+equals() : boolean
+toString() : String
Market
-double openingTime
-double closingTime
+Market(int _position int _length int _height String _owner double _openingTime double _closingTime)
+setOpeningTime(int _openingTime) : void
+setClosingTime(int _closingTime) : void
+getInfo() : void
+hashCode() : int
+equals() : boolean
+toString() : String
Office
-String jobType
+Office(int _position int _length int _height String _owner String _jobType)
+getInfo() : void
+hashCode() : int
+equals() : boolean
+toString() : String
Street
-int lenght
-GTUArrayList rights
-GTUArrayList lefts
+Street(int _length)
+setLenght() : void
+addbuildingRights(Structure building) : void
+addbuildingsLefts(Structure building) : void
+deletebuildingRights(int position int length) : void
+deletebuildingLefts(int position int length) : void
+printSiluette() : void
+remainingLength() : void
+occupiedLength() : void
+playGroundsInfo() : void
+StreetInfo() : void
+getLength() : int
Is a
Is a
Is a
Has a
Building
-String owner
+Building(int _position int _length int _height String _owner)
+getInfo() : void
+hashCode() : int
+equals() : boolean
+toString() : String
PLAYGROUND
+PLAYGROUND(int _position int _length)
+toString() String()
GtuArrayList
-ArrayList<Structure> Structures
+GtuArrayList()
+add(Structure building) : void
+delete(int position int length) : void
+findMaxHeigth() : int
+numberofObjects() : int
+occupiedLength() : int
+placeable(Structure building) : boolean
+printBorders() : void
+totalPlaygroundsLength() : int
Is a
Is a
Has a
Structure
-int position
-int heigth
-int length
+Structure(int _position int _length int _height)
+setLength(int _length) : void
+setHeigth(int _height) : void
+setPosition(int _position) : void
+getInfo() : void
+getMinPosition() : int
+getMaxPosition() : int
+getLength() : int
+getHeigth() : int
+hashCode() : int
+equals() : boolean
+toString() : String

3. PROBLEM SOLUTION APPROACH

Write your problem solution approach.

Keeping Structures

3 different containers are used to keep the structures. These are LinkedList, ArrayList, and LDLinkedList.

Creating LDLinkedList

LDLinkedList is designed as doubly linked list to increase performance. In this way, adding a Node to the end of the list, does not require traverse to the end of the list. It also provides to traverse from end of the list to the beginning of the list. LDLinkedList implements some methods which provided by AbstractList class and List interface. It also implements some methods to use lazy deletion strategy such as findNode and add (gets Node as parameter) method.

Creating GtuLDLinkedList

GtuLDLinkedList has 2 data field, one of them is to keep structures as LDLinkedList, the other one is to keep removed structures. When user call add method, the Structure which is taken as parameter is searched in removedElements LDLinkedList. If the removedElements list has this node, it uses the node instead of creating new node. When user call remove method, the element is searched in structures list. If it founds, it is removed from structures list and added to removedElements list.

Reaching elements of GtuLDLinkedList, GtuLinkedList and ArrayList

Reaching elements from linked list has O (n) time complexity. For this reason, methods of GtuLDLinkedList and GtuLinkedList use ListIterator to reach elements to decrease time complexity. Contrary to LinkedList classes ArrayList use get method to reach elements because get method of ArrayList has Theta(1) time complexity.

Designing Class

Class designs are the same as Homework1.

4. RUNNING AND RESULTS

Results are added to homework file.