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

Student Name: Enes Abdülhalik
Student Number: 1901042803

1. PROBLEM SOLUTION APPROACH

The Street classes use an abstract inner class to store buildings in it called “AbstractBuilding”, any class derived from this abstract class can be accepted as a building in the street class.
The following information is the minimum for the program to perform as expected:

1- Length of the street.

2- Length of the building.

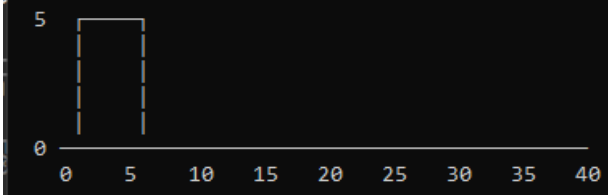
3- Height of the building.

4- Position of the building.

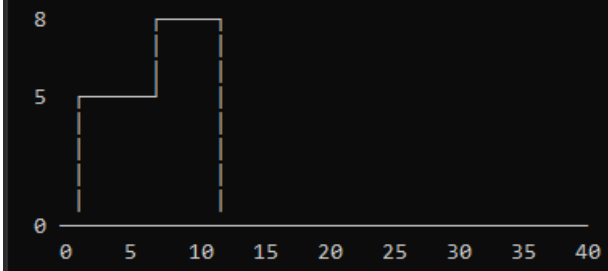
The position of the building in the street is stored inside the AbstractBuilding class but cannot be modified directly, only modified through add and remove methods from the Street class.
The maximum length of the street is not constrained, as well as the maximum height of the buildings.

TEST CASES

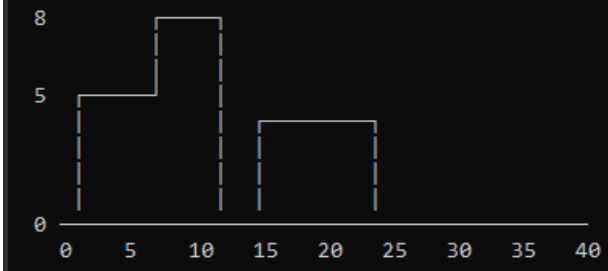
```
A street of length 0 meters is created

Mode is set to Read/Write

Add a house 6 meters long, 5 meters high, 1 meters away from the left of the Front side
```



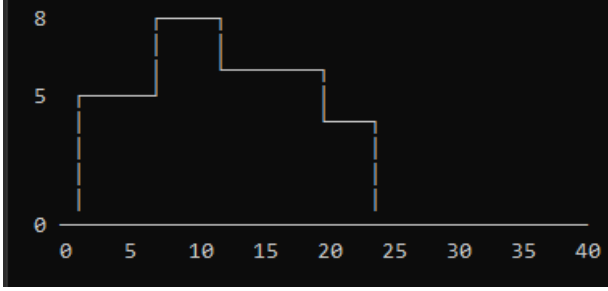

```
Add a market 6 meters long, 8 meters high, 7 meters away from the left of the Front side
```



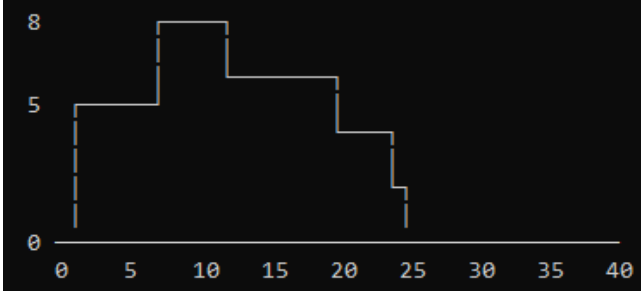

```
Add an office 10 meters long, 4 meters high, 15 meters away from the left of the Front side
```



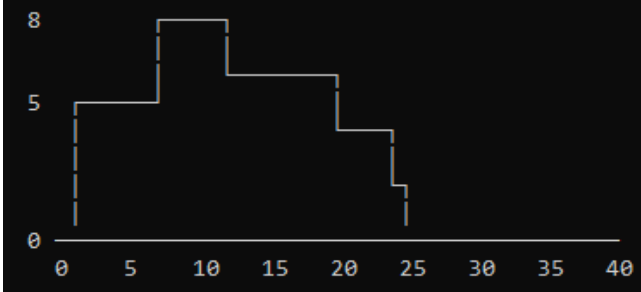

```
Add an office 8 meters long, 4 meters high, 13 meters away from the left of the Back side
```



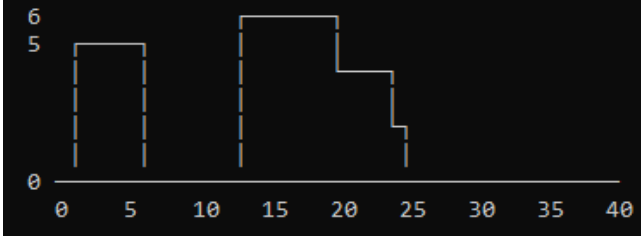
```
Add a playGround 5 meters long, 21 meters away from the left of the Back side
```



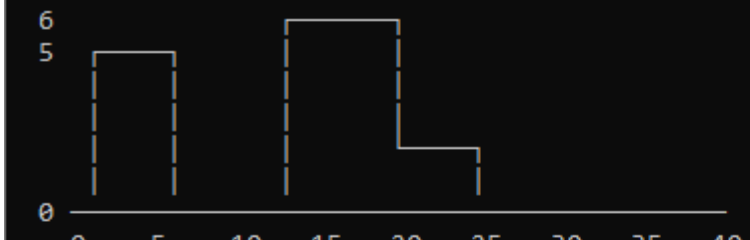

```
Add a playGround 6 meters long, 23 meters away from the left of the Back side
Insuccessful, because the place is occupied
```




```
Mode is set to Read Only

Remove whatever building is in the area 7 meters away from the left of the Front side
Insuccessful, because: Read Only Mode is Selected

Mode is set to Read/Write

Remove whatever building is in the area 7 meters away from the left of the Front side
```



```
Remove the second building from the Front side

```




```
The building 7 meters away from the left of the front side is:

Type: House
Number of rooms: 10
Color: Blue
Owner: owner man
Length (meters): 6
Height (meters): 5
Position (meters away from left): 1

The number of playgrounds in the back side is: 1
The ratio of playgrounds in the street is: 0.062500
Focus on the building in the position 14 in the back side: Useless Job
Length: 40

Front side:
Number of buildings: 1

Type: House
Number of rooms: 10
Color: Blue
Owner: owner man
Length (meters): 6
Height (meters): 5
Position (meters away from left): 1

Back side:
Number of buildings: 2

Type: Office
Owner: owner man
Length (meters): 8
Height (meters): 6
Position (meters away from left): 13

Type: Playgorund
Length (meters): 5
Height (meters): 2
Position (meters away from left): 21
```

2. RUNNING AND RESULTS

	ArrayStreet				ArrayListStreet			
Methods	Theoretical	Experimental (μs)			Theoretical	Experimental (μs)		
L = Length of the street		L = 10000	L = 10000	L = 10000		L = 10000	L = 10000	L = 10000
m = Highest building height		m = 10	m = 100	m = 1000		m = 10	m = 100	m = 1000
n = number of buildings	T(n)	n = 50	n = 500	n = 5000	T(n)	n = 50	n = 500	n = 5000
`boolean add(AbstractBuilding, int, Side)`	Θ(L)				O(n)	8.0 - 9.0	8.5 - 10.0	10.5 - 13.0
AbstractBuilding remove(int, Side)	Θ(l)				O(n)	11.5 - 12.5	14.0 - 16.0	16.0 - 20.0
AbstractBuilding remove_Position(int, Side)	-				O(nlogn)	70.0 - 76.0	76.0 - 86.0	85.0 - 100.0
AbstractBuilding get(int, Side)	Θ(1)				Θ(1)	8.0 - 9.0	8.5 - 10.0	11.5 - 13.0
AbstractBuilding get_Position(int, Side)	-				O(logn)	77.0 - 85.0	115 - 125	105 - 130
String getSilhouette()	Θ(L*m)				Θ(L*m)	90 - 95*10^3	140-150*10^3	550-750*10^3
int countType(String, Side)	O(L)				Θ(n)	85 - 90	374 - 400	1.56-1.70*10^3
int countTypeArea(String, Side)	O(L)				Θ(n)	125 - 140	390 - 480	1.70-1.76*10^3
float playGroundRatio()	O(L)				Θ(n)	85 - 100	400 - 550	1.50-1.60*10^3
int getEmptyArea(Side)	O(L)				O(L*n)	10 - 12*10^3	13.5 - 15.0*10^3	14.0-16.0*10^3
String toString()	Θ(L*n)				Θ(n)	27 - 33*10^3	51.0 - 53.0*10^3	90 - 95*10^3

	LinkedListStreet				LDLinkedStreet			
Methods	Theoretical	Experimental (μs)			Theoretical	Experimental (μs)		
L = Length of the street		L = 10000	L = 10000	L = 10000		L = 10000	L = 10000	L = 10000
m = Highest building height		m = 10	m = 100	m = 1000		m = 10	m = 100	m = 1000
n = number of buildings	T(n)	n = 50	n = 500	n = 5000	T(n)	n = 50	n = 500	n = 5000
`boolean add(AbstractBuilding, int, Side)`	O(n)	7.00 - 8.00	9.00 - 11.5	10.5 - 13.0	O(n)	8.8 - 10.0	9.60 - 12.0	11.0 - 14.0
AbstractBuilding remove(int, Side)	O(n)	11.0 - 13.0	14.0 - 17.3	19.0 - 21.0	O(n)	10.7 - 11.7	14.0 - 18.0	15.0 - 20.0
AbstractBuilding remove_Position(int, Side)	O(n)	24.0 - 26.0	100 - 115	430 - 460	O(n)	120 - 140	620 - 700	40-45*10^3
AbstractBuilding get(int, Side)	O(n)	11.0 - 12.5	15.0 22.0	40.0 - 55.0	O(n)	11.0 -12.7	13.0 - 16.5	40.0 - 47.0
AbstractBuilding get_Position(int, Side)	O(n)	55.0 - 67.0	160 - 190	540 - 580	O(n)	180 - 200	650 - 760	40-45*10^3
String getSilhouette()	Θ(L*m)	87-90*10^3	170-200*10^3	660-960*10^3	Θ(L*m)	97-100*10^3	3*10^6	
int countType(String, Side)	Θ(n)	70 - 93	340 - 500	1.35-1.50*10^3	Θ(n)			
int countTypeArea(String, Side)	Θ(n)	120 - 140	350 - 500	1.40-1.60*10^3	Θ(n)			
float playGroundRatio()	Θ(n)	75 - 90	350 - 450	1.25-1.4*10^3	Θ(n)			
int getEmptyArea(Side)	O(L*n)	10.0-10.5*10^3	36-39*10^3	173-175*10^3	O(L*n)			
String toString()	Θ(n)	27 - 29*10^3	50 - 53*10^3	90 - 96*10^3	Θ(n)			