GIT Department of Computer Engineering
CSE 222/505 - Spring 2022
Homework #03 Report

Mert Gürşimşir
1901042646

PART 1

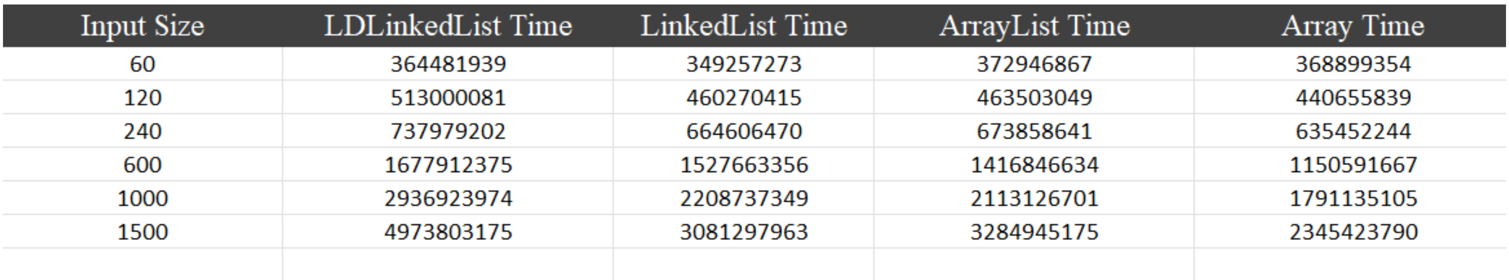
RUNNING TIMES

	LDLinkedList	LinkedList	ArrayList	Array
Constructor	θ(n)	θ(n)	θ(n)	θ(n)
add()	O(n^2)	O(n^2)	O(n)	O(n)
delete()	O(n^2)	O(n^2)	O(n)	O(n)
isEmpty()	θ(n^2)	θ(n^2)	θ(n)	θ(n)
remainingLands()	θ(n^2)	θ(n^2)	θ(n)	θ(n)
buildingsList()	θ(n^2)	θ(n^2)	θ(n)	θ(n)
numberOfPlaygrounds()	θ(n^2)	θ(n^2)	θ(n)	θ(n)
ratioOfPlaygrounds()	θ(n^2)	θ(n^2)	θ(n)	θ(n)
marketLength()	θ(n^2)	θ(n^2)	θ(n)	θ(n)
houseLength()	θ(n^2)	θ(n^2)	θ(n)	θ(n)
officeLength()	θ(n^2)	θ(n^2)	θ(n)	θ(n)
silhouette()	O(n^3)	O(n^3)	O(n^3)	O(n^3)
getBuilding()	O(n)	O(n)	θ(1)	θ(1)

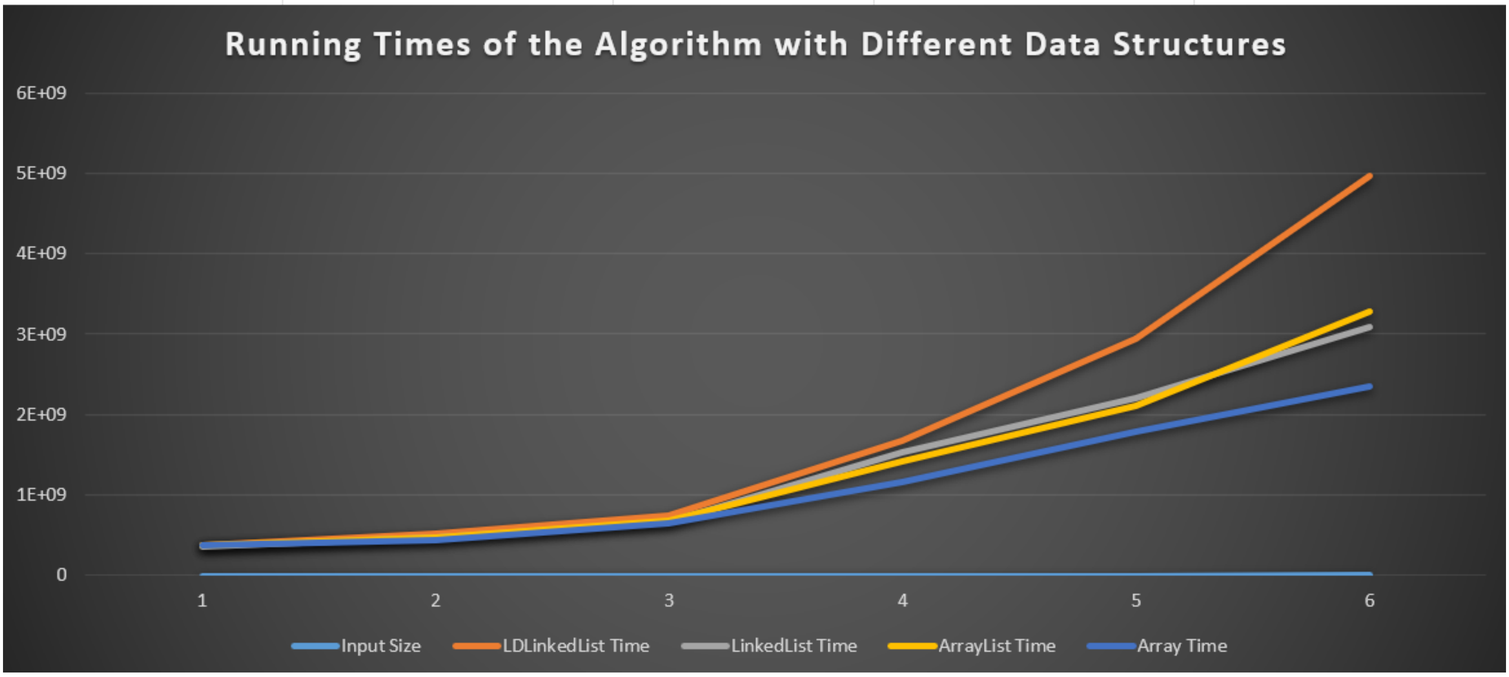
As it seems, there are so many methods that is θ(n^2) in linked lists. That's because, according to my algorithm, I have to traverse through all the elements in the street List and also in this loops I have to get the element at the specified position. get method is linear time in LDLinkedList that I have implemented. Therefore those methods takes quadratic time.

In ArrayList and Array implementations, many methods takes linear time. That's because get method takes constant time in ArrayList and we can access elements in constant time in Arrays too. So all I had to do is traverse through the street (which needs linear time) and access the elements (which needs constant time).

In silhouette method, all the implementations take cubic time. The reason for this is existence of so many nested loops. I have used 3 different data structures in 3 different classes to keep char values. According to my algorithm, I had to traverse through both for street structure and also for data structure that keeps char values (‘-‘ or ‘|’ or ‘ ‘). I also had to check all column whether I can fill that column with ‘|’ or I cannot because there is another building at the other side of the street. For all these checks, silhouette method takes cubic time.



Input Size	LDLinkedList Time	LinkedList Time	ArrayList Time	Array Time
60	364481939	349257273	372946867	368899354
120	513000081	460270415	463503049	440655839
240	737979202	664606470	673858641	635452244
600	1677912375	1527663356	1416846634	1150591667
1000	2936923974	2208737349	2113126701	1791135105
1500	4973803175	3081297963	3284945175	2345423790



Times in nanoseconds.

There are low order terms and they are one of the reasons for different running times. LDLinkedList use lazy deletion strategy, so actually it keeps 2 linked list. To remove an element from it, we have to add that element to the other list that keeps removed elements. So it requires more job to do and obviously takes more time.

We can’t evaluate exact time values because there are some other factors like computer, OS, compiler.

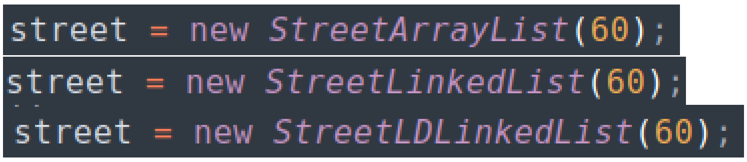
PART 2

1. SYSTEM REQUIREMENTS

Firstly, to run the program, you should choose your data structure and run the desired driver file. There are 3 of them:

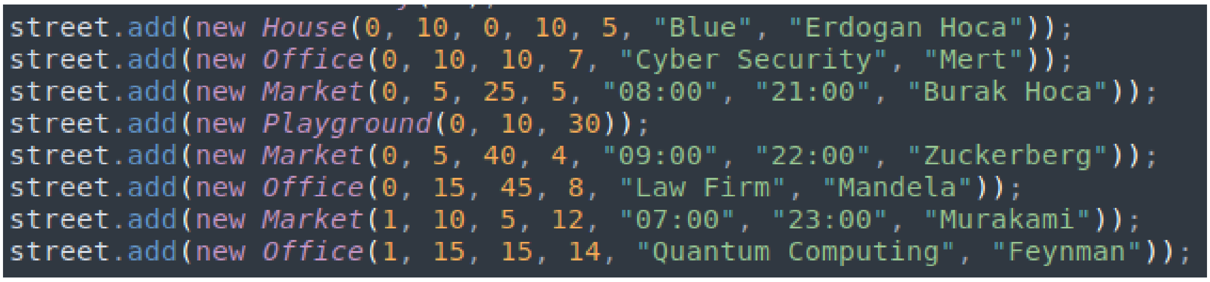
1. DriverArrayList.java
2. DriverLinkedList.java
3. DriverLDLinkedList.java

This is a city planning software that will be used for designing a small one street town. There should be a street in the desired structure:



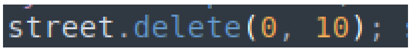
```
street = new StreetArrayList(60);
street = new StreetLinkedList(60);
street = new StreetLDLinkedList(60);
```

Street needs to be given proper length. Proper means it cannot be negative otherwise there will be exception specifies that issue.



```
street.add(new House(0, 10, 0, 10, 5, "Blue", "Erdogan Hoca"));
street.add(new Office(0, 10, 10, 7, "Cyber Security", "Mert"));
street.add(new Market(0, 5, 25, 5, "08:00", "21:00", "Burak Hoca"));
street.add(new Playground(0, 10, 30));
street.add(new Market(0, 5, 40, 4, "09:00", "22:00", "Zuckerberg"));
street.add(new Office(0, 15, 45, 8, "Law Firm", "Mandela"));
street.add(new Market(1, 10, 5, 12, "07:00", "23:00", "Murakami"));
street.add(new Office(1, 15, 15, 14, "Quantum Computing", "Feynman"));
```

Then, system can accept buildings. They can be added to street with their length, position, side, height, and any other type specific properties. Here, buildings that does not fit in the street according to their position and length values will cause to an exception. Any other unproper values (like giving 5 as side value instead of 0 or 1) will also cause to an exception.



```
street.delete(0, 10);
```

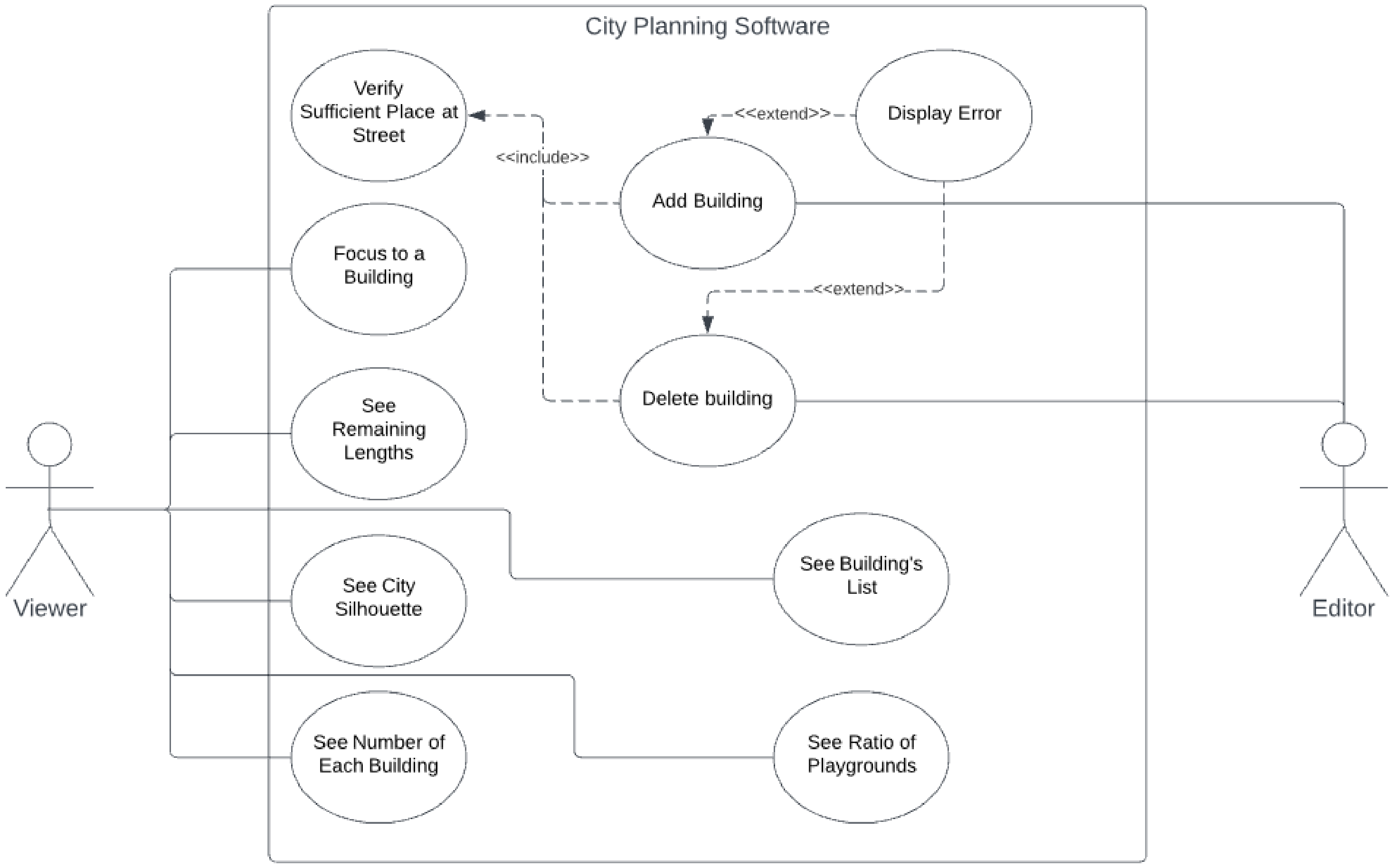
As shown above, system can delete any of the buildings in the street by looking its side, and position value respectively. Here, empty spaces or any other unproper values will cause an exception.

After these processes, your city planning software is ready to use.

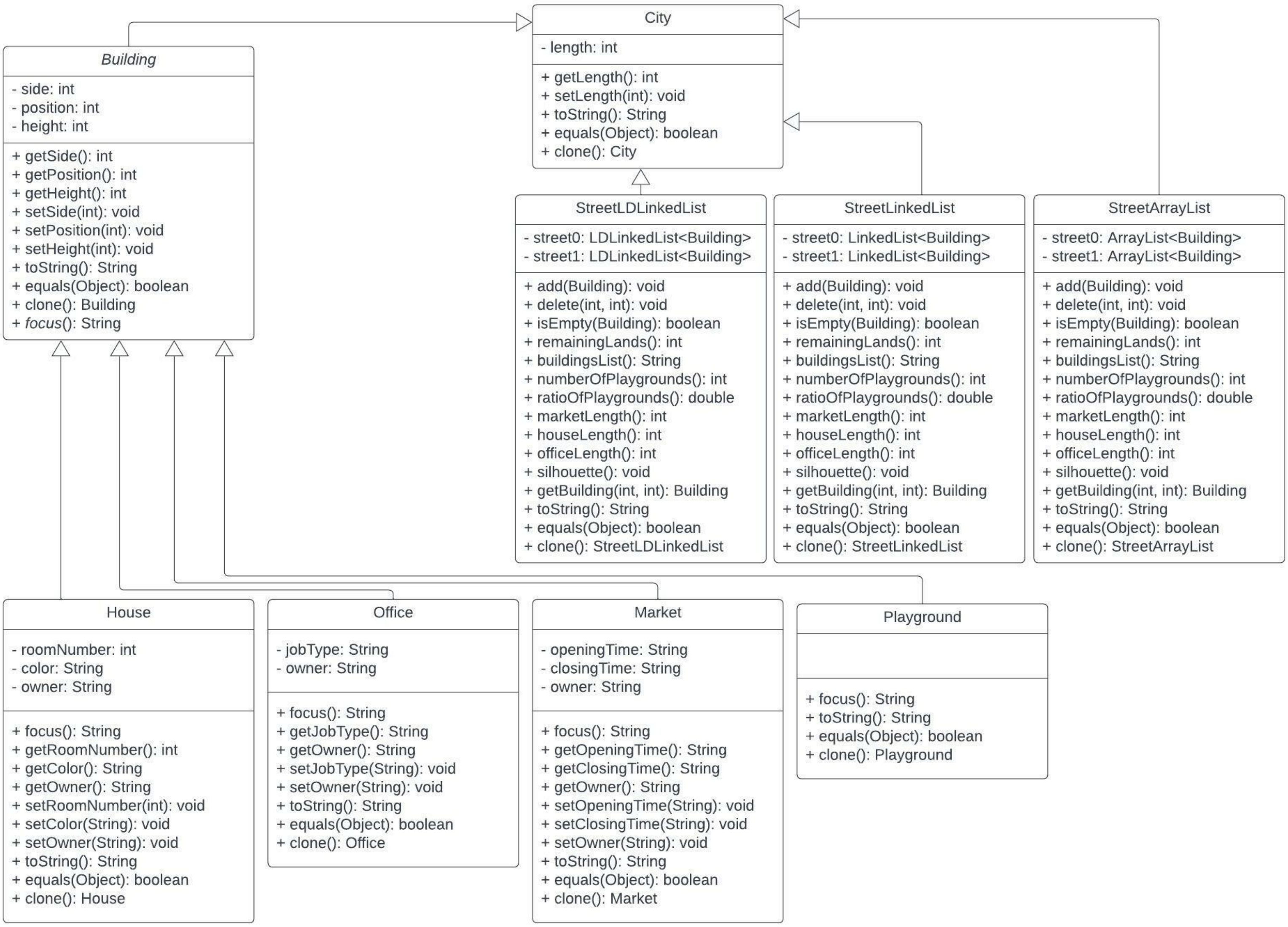
--> You can also run TestLDLinkedList.java to try my LDLinkedList.

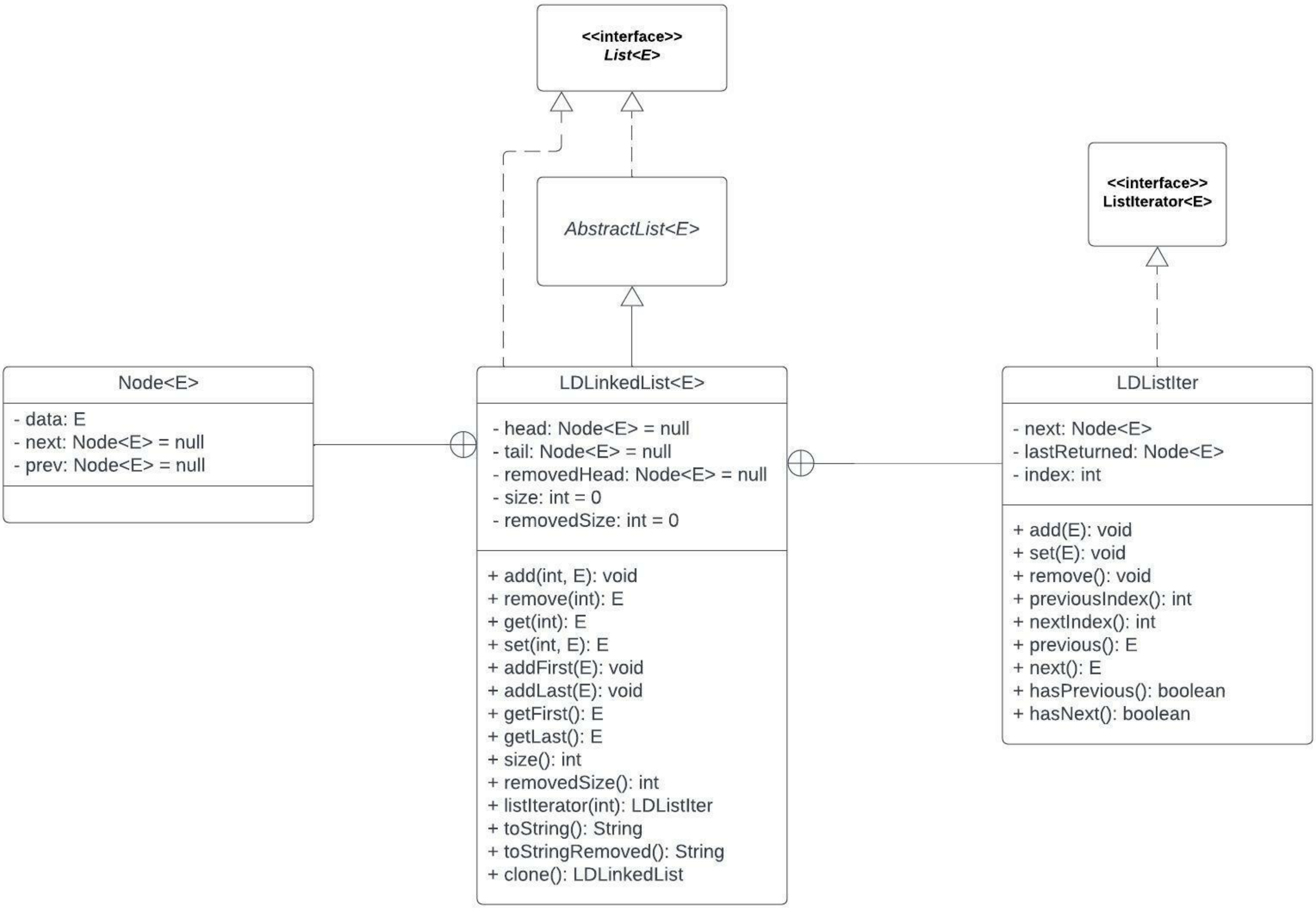
2. USE CASE AND CLASS DIAGRAMS

a. USE CASE DIAGRAM



b. CLASS DIAGRAM





3. PROBLEM SOLUTION APPROACH

Firstly, I have to implement new linked list class named LDLinkedList which uses lazy deletion strategy. That was not easy for me because implementing a data structure in Java is new to me. I have looked at books and online Javadocs in the oracle site. Then I implemented it successfully.

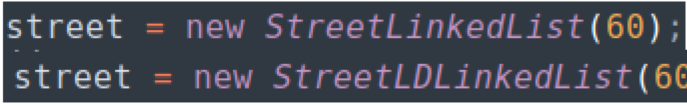
I draw my plan on a blank paper to what to do. I spend my time on algorithm to not do anything wrong. Then I implemented 3 separate container class for the street. One of them keeps elements in LinkedList, other one in ArrayList and last one in LDLinkedList that I have implemented. It is not so hard to adapt my code to three of them.

I have used polymorphism a lot in these 3 classes. I have kept Building structure and assign specified building to it (House, Office, Market, or Playground). Then I have used their own methods.

4. TEST CASES

Create a street

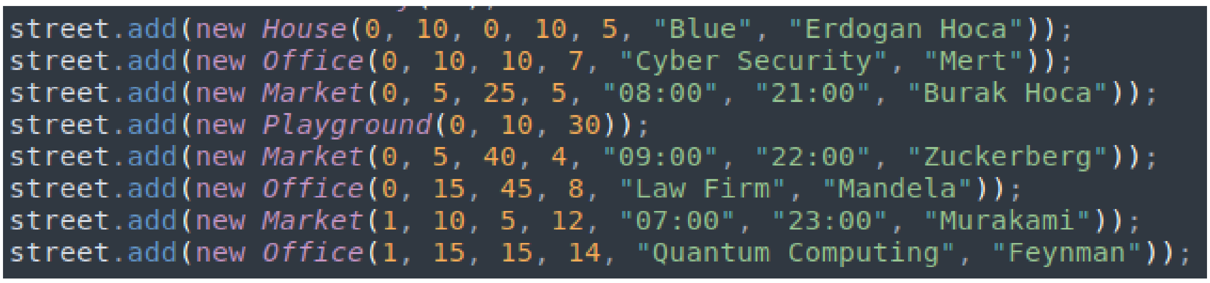
```java
street = new StreetArrayList(60);
```



```java
street = new StreetLinkedList(60);
```

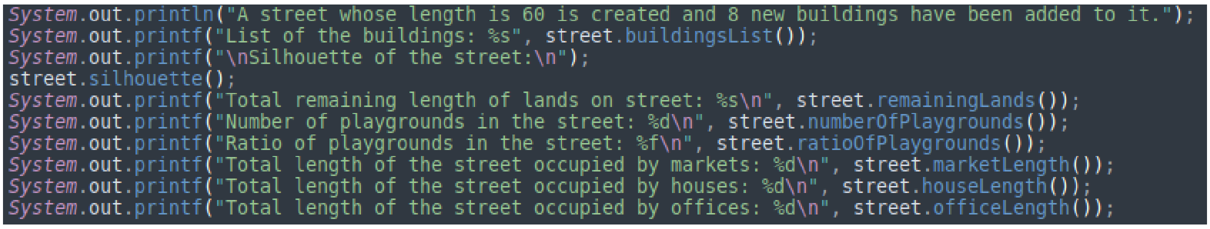
```java
street = new StreetLDLinkedList(60);
```

Adding buildings to street



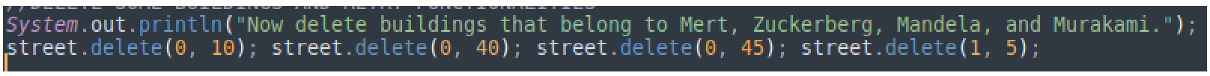
```java
street.add(new House(0, 10, 0, 10, 5, "Blue", "Erdogan Hoca"));
street.add(new Office(0, 10, 10, 7, "Cyber Security", "Mert"));
street.add(new Market(0, 5, 25, 5, "08:00", "21:00", "Burak Hoca"));
street.add(new Playground(0, 10, 30));
street.add(new Market(0, 5, 40, 4, "09:00", "22:00", "Zuckerberg"));
street.add(new Office(0, 15, 45, 8, "Law Firm", "Mandela"));
street.add(new Market(1, 10, 5, 12, "07:00", "23:00", "Murakami"));
street.add(new Office(1, 15, 15, 14, "Quantum Computing", "Feynman"));
```

Trying functionalities



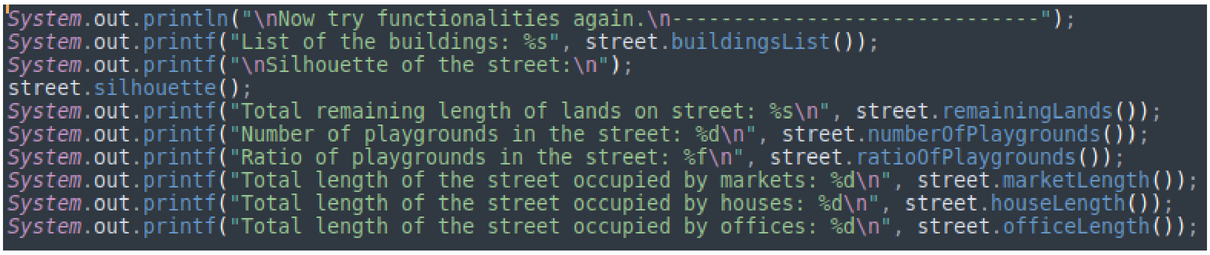
```java
System.out.println("A street whose length is 60 is created and 8 new buildings have been added to it.");
System.out.printf("List of the buildings: %s", street.buildingsList());
System.out.printf("\nSilhouette of the street:\n");
street.silhouette();
System.out.printf("Total remaining length of lands on street: %s\n", street.remainingLands());
System.out.printf("Number of playgrounds in the street: %d\n", street.numberOfPlaygrounds());
System.out.printf("Ratio of playgrounds in the street: %f\n", street.ratioOfPlaygrounds());
System.out.printf("Total length of the street occupied by markets: %d\n", street.marketLength());
System.out.printf("Total length of the street occupied by houses: %d\n", street.houseLength());
System.out.printf("Total length of the street occupied by offices: %d\n", street.officeLength());
```

Delete some buildings



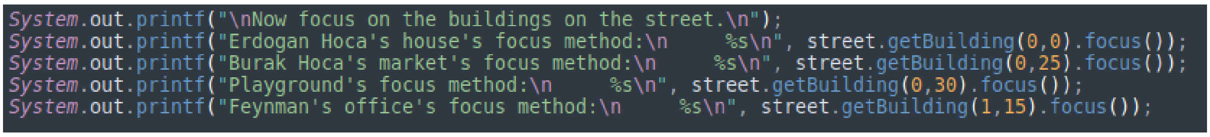
```java
System.out.println("Now delete buildings that belong to Mert, Zuckerberg, Mandela, and Murakami.");
street.delete(0, 10); street.delete(0, 40); street.delete(0, 45); street.delete(1, 5);
```

Retry functionalities



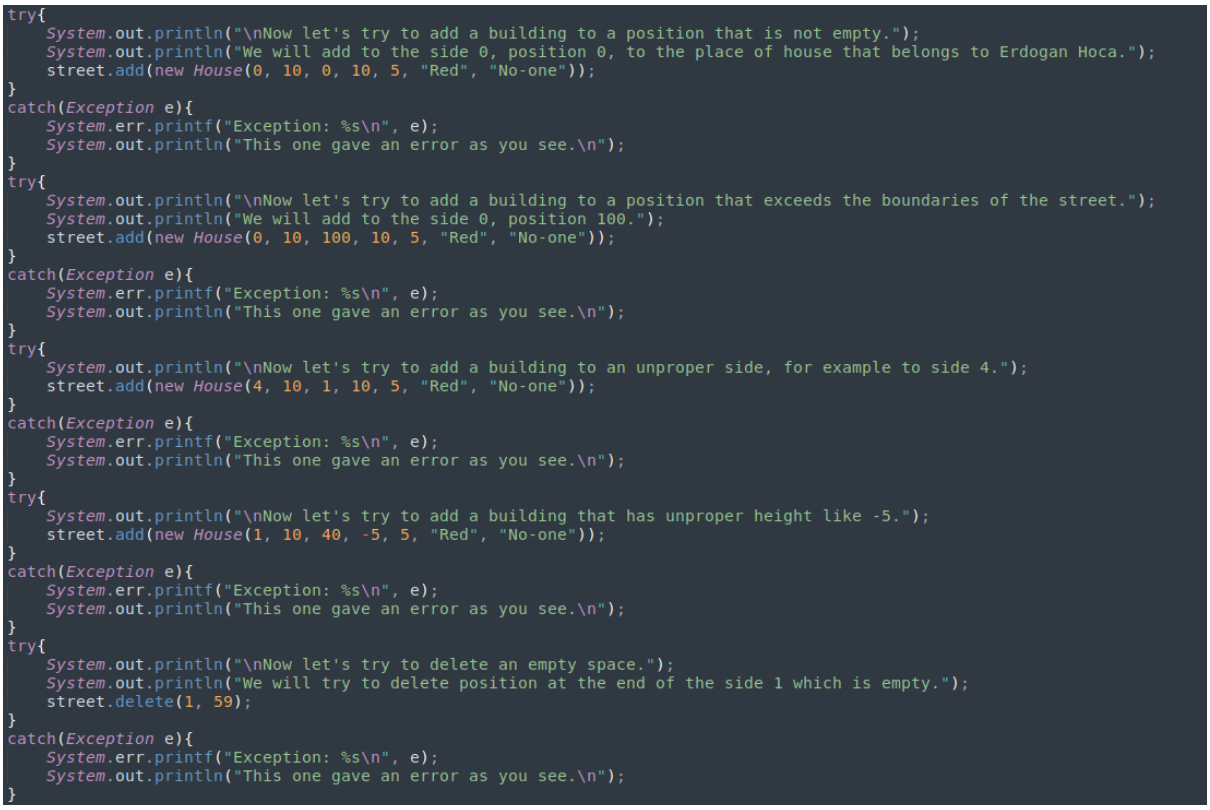
```java
System.out.println("\nNow try functionalities again.\n------------------------------");
System.out.printf("List of the buildings: %s", street.buildingsList());
System.out.printf("\nSilhouette of the street:\n");
street.silhouette();
System.out.printf("Total remaining length of lands on street: %s\n", street.remainingLands());
System.out.printf("Number of playgrounds in the street: %d\n", street.numberOfPlaygrounds());
System.out.printf("Ratio of playgrounds in the street: %f\n", street.ratioOfPlaygrounds());
System.out.printf("Total length of the street occupied by markets: %d\n", street.marketLength());
System.out.printf("Total length of the street occupied by houses: %d\n", street.houseLength());
System.out.printf("Total length of the street occupied by offices: %d\n", street.officeLength());
```

Focusing on the buildings



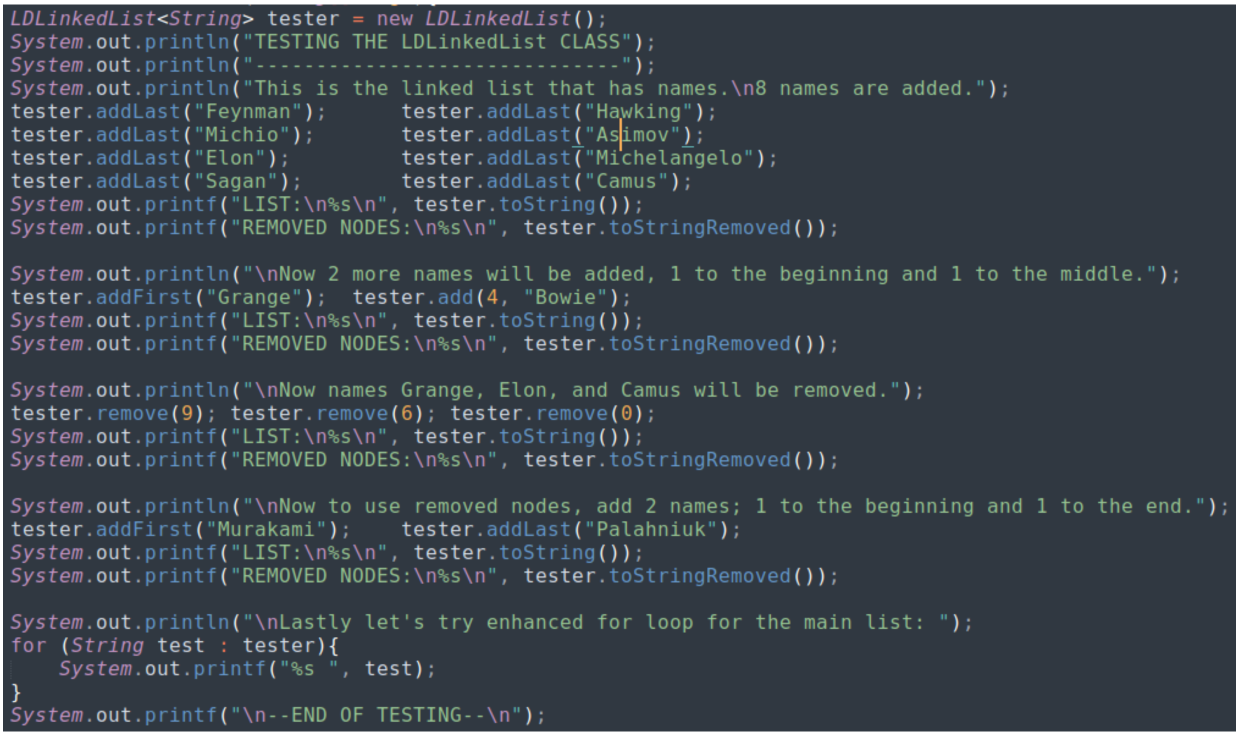
```java
System.out.printf("\nNow focus on the buildings on the street.\n");
System.out.printf("Erdogan Hoca's house's focus method:\n     %s\n", street.getBuilding(0,0).focus());
System.out.printf("Burak Hoca's market's focus method:\n     %s\n", street.getBuilding(0,25).focus());
System.out.printf("Playground's focus method:\n     %s\n", street.getBuilding(0,30).focus());
System.out.printf("Feynman's office's focus method:\n     %s\n", street.getBuilding(1,15).focus());
```

Trying error cases



```java
try{
    System.out.println("\nNow let's try to add a building to a position that is not empty.");
    System.out.println("We will add to the side 0, position 0, to the place of house that belongs to Erdogan Hoca.");
    street.add(new House(0, 10, 0, 10, 5, "Red", "No-one"));
}
catch(Exception e){
    System.err.printf("Exception: %s\n", e);
    System.out.println("This one gave an error as you see.\n");
}
try{
    System.out.println("\nNow let's try to add a building to a position that exceeds the boundaries of the street.");
    System.out.println("We will add to the side 0, position 100.");
    street.add(new House(0, 10, 100, 10, 5, "Red", "No-one"));
}
catch(Exception e){
    System.err.printf("Exception: %s\n", e);
    System.out.println("This one gave an error as you see.\n");
}
try{
    System.out.println("\nNow let's try to add a building to an unproper side, for example to side 4.");
    street.add(new House(4, 10, 1, 10, 5, "Red", "No-one"));
}
catch(Exception e){
    System.err.printf("Exception: %s\n", e);
    System.out.println("This one gave an error as you see.\n");
}
try{
    System.out.println("\nNow let's try to add a building that has unproper height like -5.");
    street.add(new House(1, 10, 40, -5, 5, "Red", "No-one"));
}
catch(Exception e){
    System.err.printf("Exception: %s\n", e);
    System.out.println("This one gave an error as you see.\n");
}
try{
    System.out.println("\nNow let's try to delete an empty space.");
    System.out.println("We will try to delete position at the end of the side 1 which is empty.");
    street.delete(1, 59);
}
catch(Exception e){
    System.err.printf("Exception: %s\n", e);
    System.out.println("This one gave an error as you see.\n");
}
```

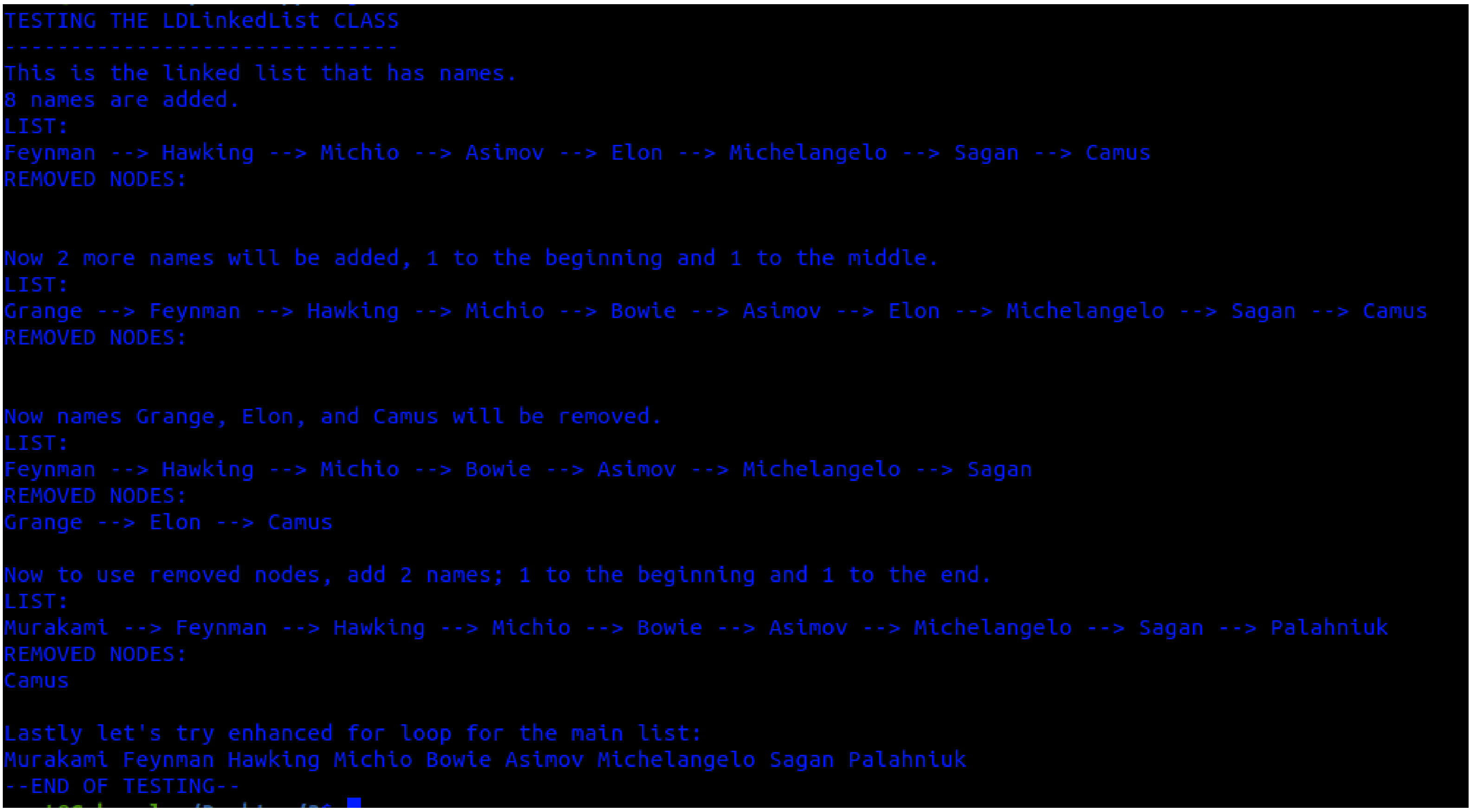
TEST CASES FOR THE LDLINKEDLIST



```java
LDLinkedList<String> tester = new LDLinkedList();
System.out.println("TESTING THE LDLinkedList CLASS");
System.out.println("------------------------------");
System.out.println("This is the linked list that has names.\n8 names are added.");
tester.addLast("Feynman");      tester.addLast("Hawking");
tester.addLast("Michio");       tester.addLast("Asimov");
tester.addLast("Elon");         tester.addLast("Michelangelo");
tester.addLast("Sagan");        tester.addLast("Camus");
System.out.printf("LIST:\n%s\n", tester.toString());
System.out.printf("REMOVED NODES:\n%s\n", tester.toStringRemoved());

System.out.println("\nNow 2 more names will be added, 1 to the beginning and 1 to the middle.");
tester.addFirst("Grange");  tester.add(4, "Bowie");
System.out.printf("LIST:\n%s\n", tester.toString());
System.out.printf("REMOVED NODES:\n%s\n", tester.toStringRemoved());

System.out.println("\nNow names Grange, Elon, and Camus will be removed.");
tester.remove(9); tester.remove(6); tester.remove(0);
System.out.printf("LIST:\n%s\n", tester.toString());
System.out.printf("REMOVED NODES:\n%s\n", tester.toStringRemoved());

System.out.println("\nNow to use removed nodes, add 2 names; 1 to the beginning and 1 to the end.");
tester.addFirst("Murakami");    tester.addLast("Palahniuk");
System.out.printf("LIST:\n%s\n", tester.toString());
System.out.printf("REMOVED NODES:\n%s\n", tester.toStringRemoved());

System.out.println("\nLastly let's try enhanced for loop for the main list: ");
for (String test : tester){
    System.out.printf("%s ", test);
}
System.out.printf("\n--END OF TESTING--\n");
```

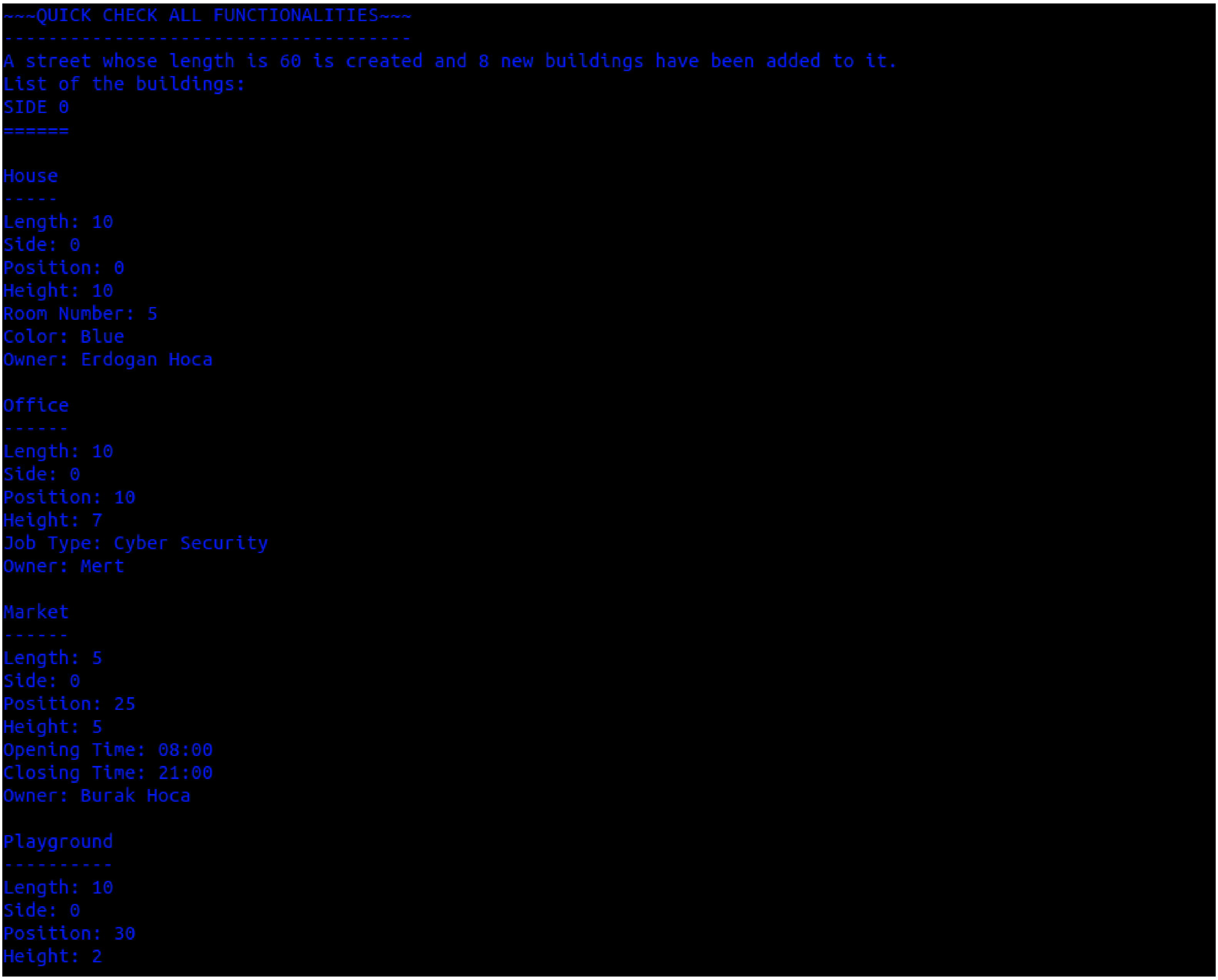
5. RUNNING AND RESULTS

LDLINKEDLIST TEST

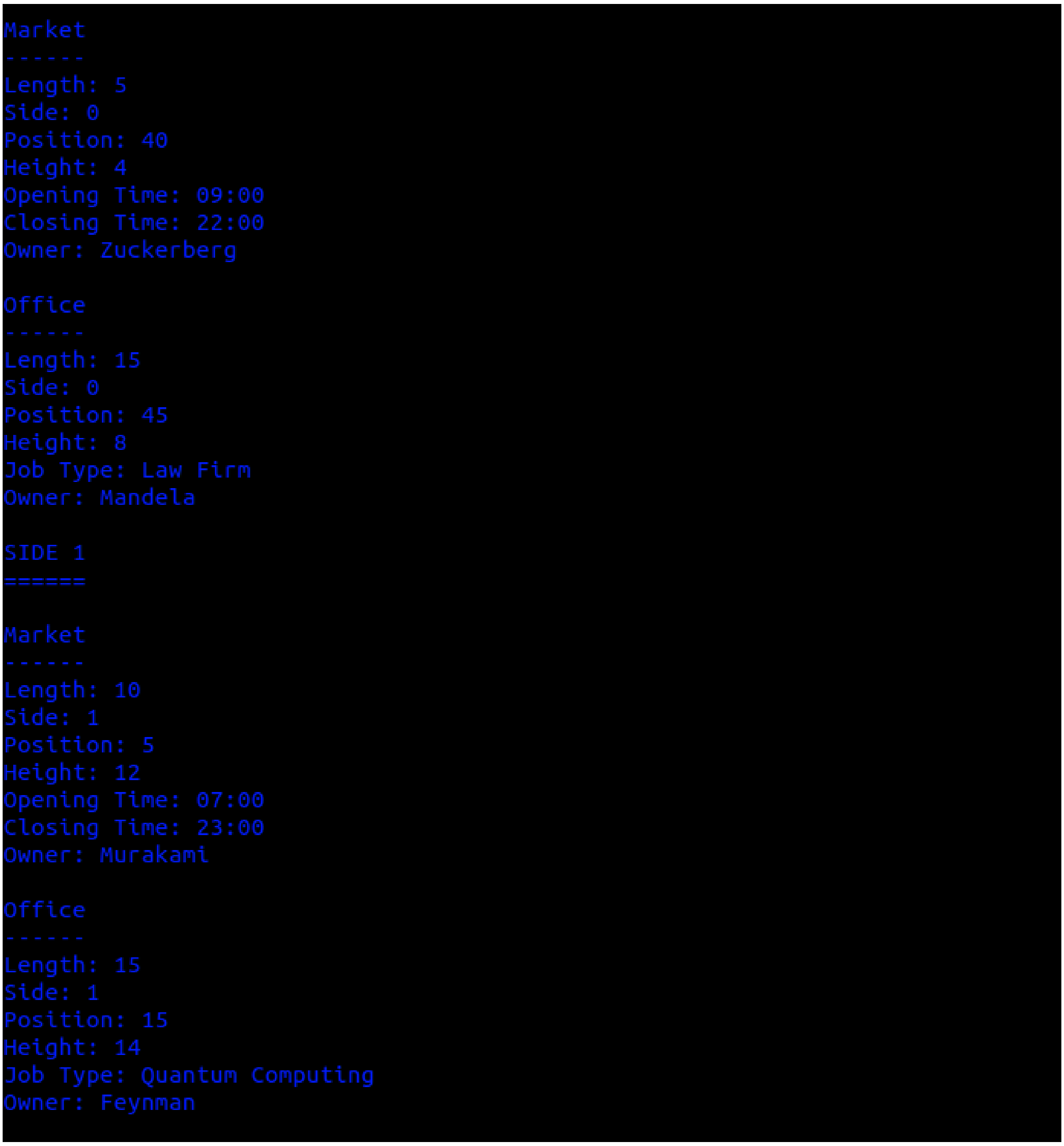


```
TESTING THE LDLinkedList CLASS
------------------------------
This is the linked list that has names.
8 names are added.
LIST:
Feynman --> Hawking --> Michio --> Asimov --> Elon --> Michelangelo --> Sagan --> Camus
REMOVED NODES:


Now 2 more names will be added, 1 to the beginning and 1 to the middle.
LIST:
Grange --> Feynman --> Hawking --> Michio --> Bowie --> Asimov --> Elon --> Michelangelo --> Sagan --> Camus
REMOVED NODES:


Now names Grange, Elon, and Camus will be removed.
LIST:
Feynman --> Hawking --> Michio --> Bowie --> Asimov --> Michelangelo --> Sagan
REMOVED NODES:
Grange --> Elon --> Camus

Now to use removed nodes, add 2 names; 1 to the beginning and 1 to the end.
LIST:
Murakami --> Feynman --> Hawking --> Michio --> Bowie --> Asimov --> Michelangelo --> Sagan --> Palahniuk
REMOVED NODES:
Camus

Lastly let's try enhanced for loop for the main list:
Murakami Feynman Hawking Michio Bowie Asimov Michelangelo Sagan Palahniuk
--END OF TESTING--
```

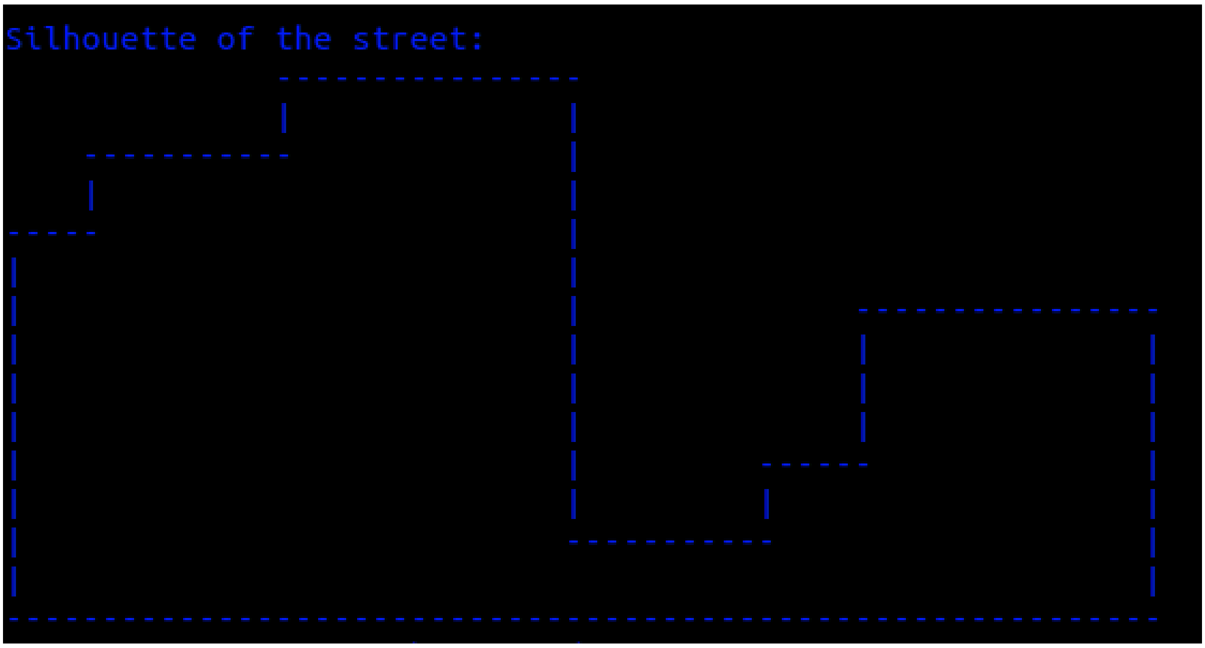
CITY TEST



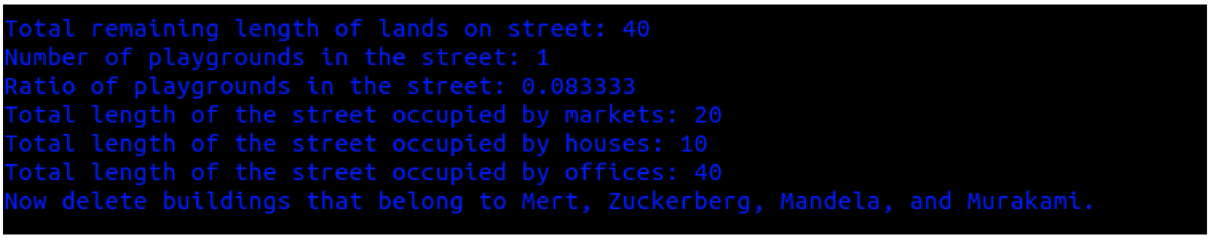
```
~~~QUICK CHECK ALL FUNCTIONALITIES~~~
-------------------------------------
A street whose length is 60 is created and 8 new buildings have been added to it.
List of the buildings:
SIDE 0
======

House
-----
Length: 10
Side: 0
Position: 0
Height: 10
Room Number: 5
Color: Blue
Owner: Erdogan Hoca

Office
------
Length: 10
Side: 0
Position: 10
Height: 7
Job Type: Cyber Security
Owner: Mert

Market
------
Length: 5
Side: 0
Position: 25
Height: 5
Opening Time: 08:00
Closing Time: 21:00
Owner: Burak Hoca

Playground
----------
Length: 10
Side: 0
Position: 30
Height: 2
```



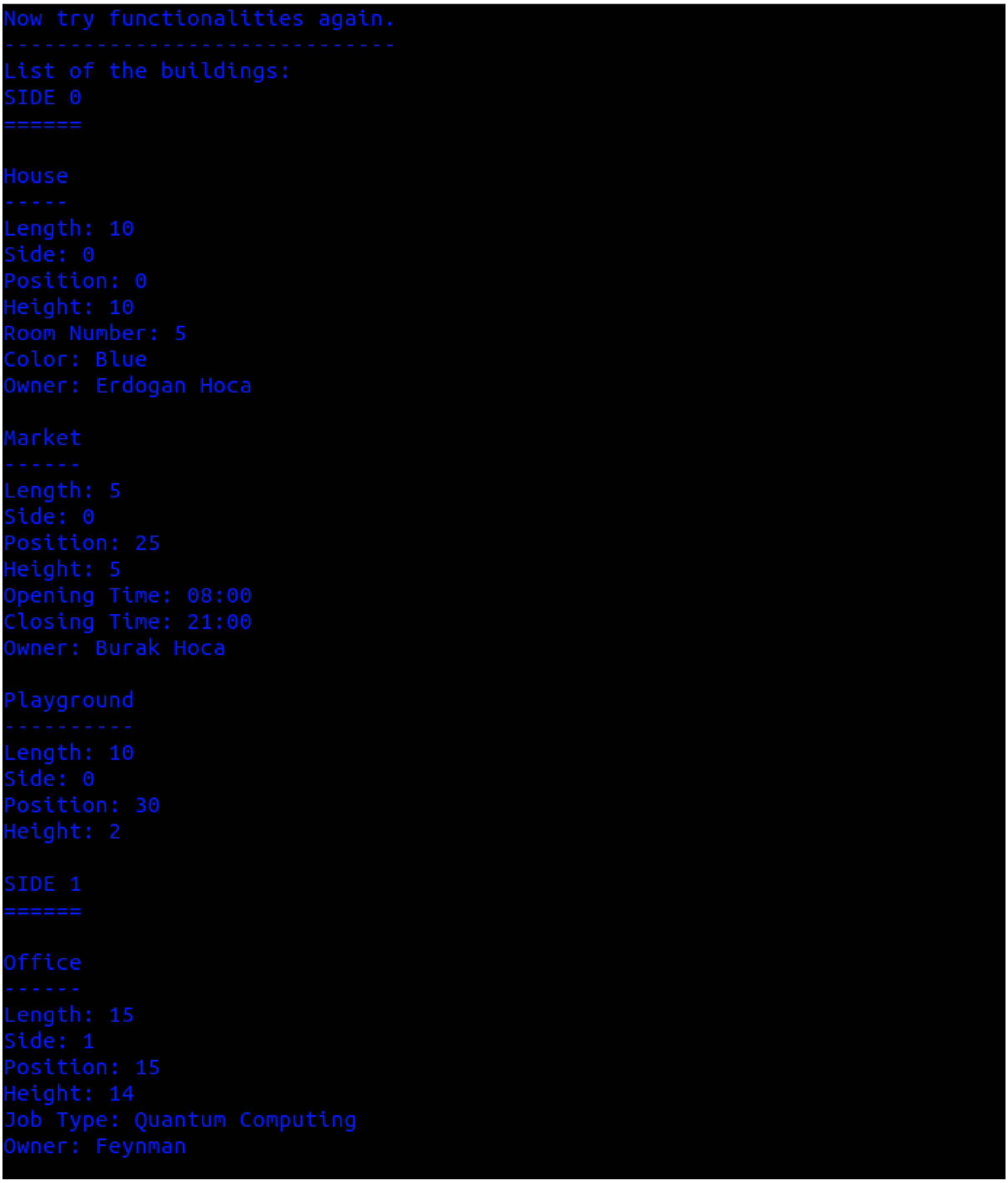
```
Market
------
Length: 5
Side: 0
Position: 40
Height: 4
Opening Time: 09:00
Closing Time: 22:00
Owner: Zuckerberg

Office
------
Length: 15
Side: 0
Position: 45
Height: 8
Job Type: Law Firm
Owner: Mandela

SIDE 1
======

Market
------
Length: 10
Side: 1
Position: 5
Height: 12
Opening Time: 07:00
Closing Time: 23:00
Owner: Murakami

Office
------
Length: 15
Side: 1
Position: 15
Height: 14
Job Type: Quantum Computing
Owner: Feynman
```



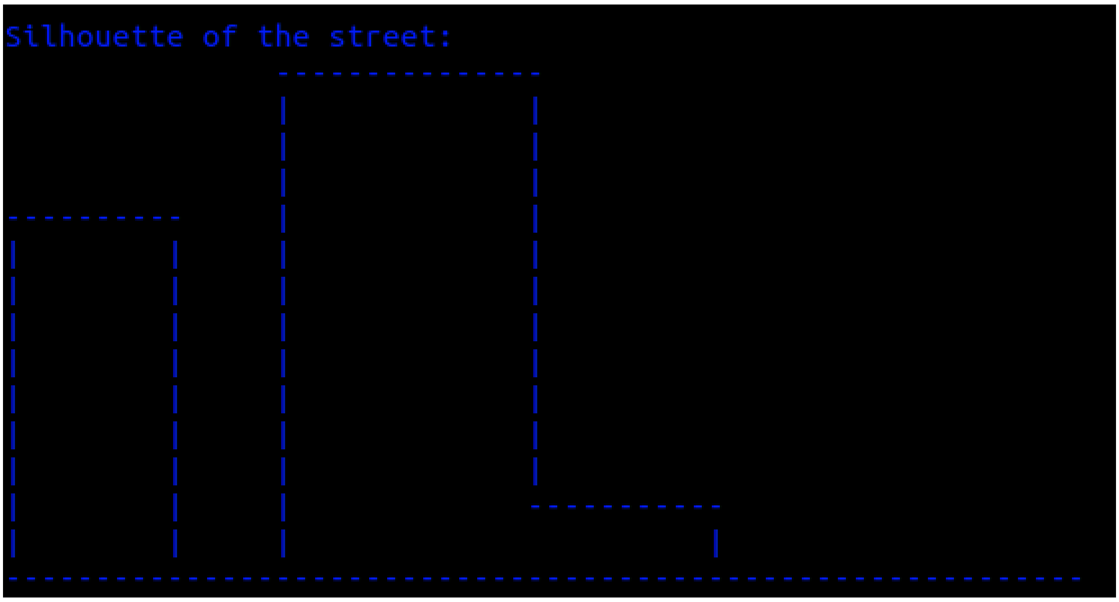
```
Silhouette of the street:
              ----------------
              |              |
    -----------              |
    |                        |
-----                        |
|                            |
|                            |              ----------------
|                            |              |              |
|                            |              |              |
|                            |              |              |
|                            |         ------              |
|                            |         |                   |
|                            -----------                   |
|                                                          |
------------------------------------------------------------
```



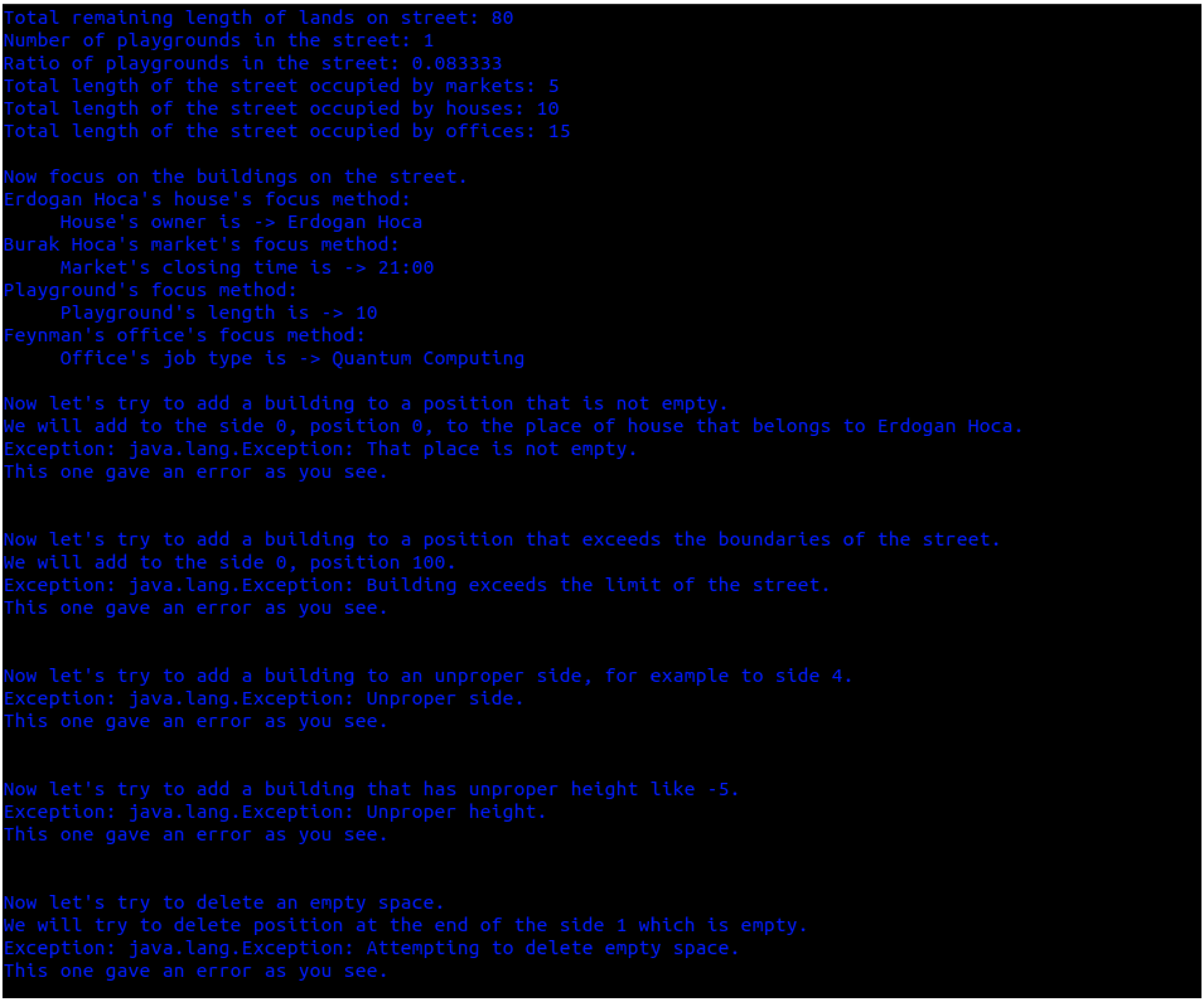
```
Total remaining length of lands on street: 40
Number of playgrounds in the street: 1
Ratio of playgrounds in the street: 0.083333
Total length of the street occupied by markets: 20
Total length of the street occupied by houses: 10
Total length of the street occupied by offices: 40
Now delete buildings that belong to Mert, Zuckerberg, Mandela, and Murakami.
```



```
Now try functionalities again.
------------------------------
List of the buildings:
SIDE 0
======

House
-----
Length: 10
Side: 0
Position: 0
Height: 10
Room Number: 5
Color: Blue
Owner: Erdogan Hoca

Market
------
Length: 5
Side: 0
Position: 25
Height: 5
Opening Time: 08:00
Closing Time: 21:00
Owner: Burak Hoca

Playground
----------
Length: 10
Side: 0
Position: 30
Height: 2

SIDE 1
======

Office
------
Length: 15
Side: 1
Position: 15
Height: 14
Job Type: Quantum Computing
Owner: Feynman
```

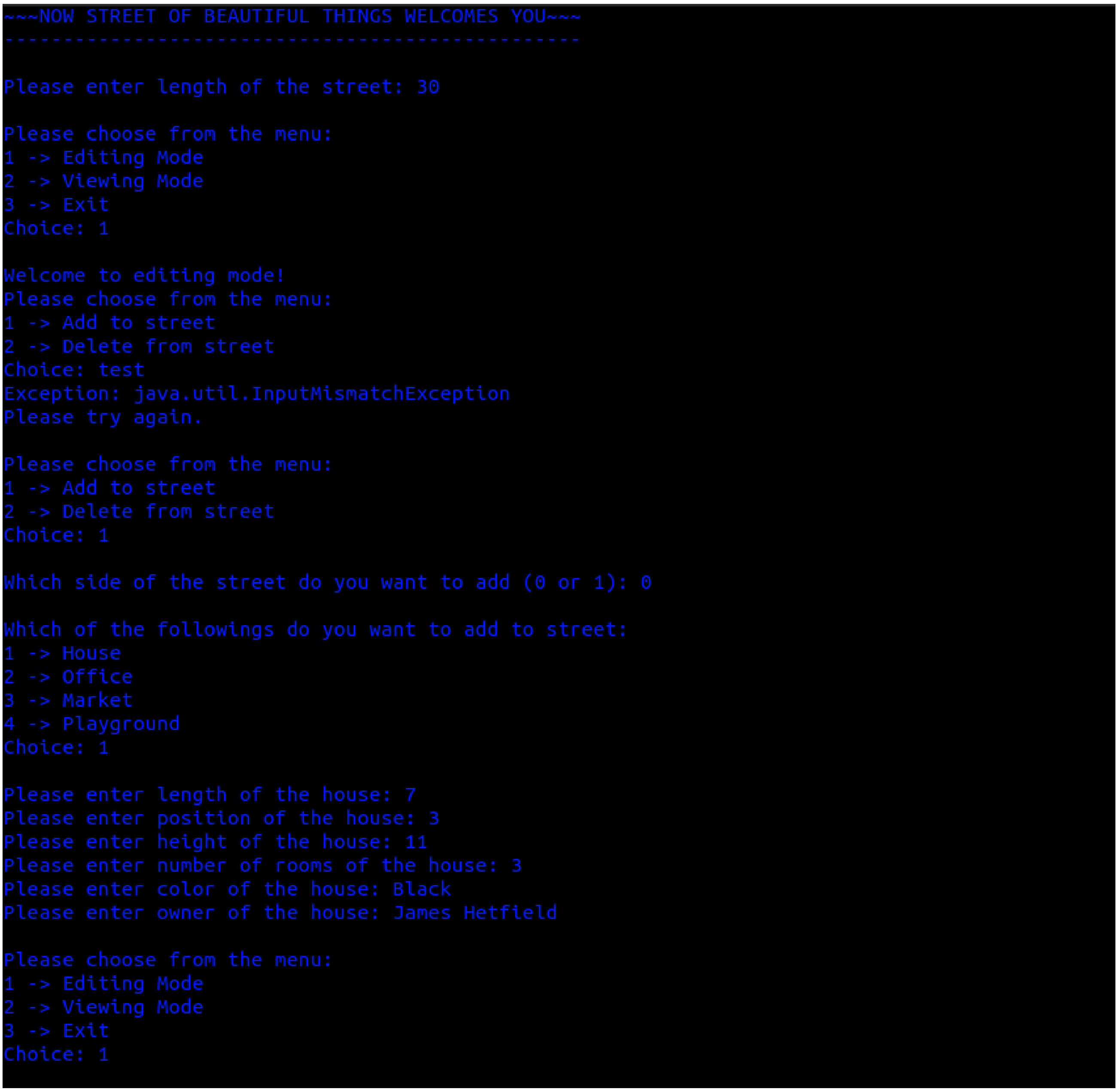


```
Silhouette of the street:
               ---------------
               |             |
               |             |
               |             |
----------     |             |
|        |     |             |
|        |     |             |
|        |     |             |
|        |     |             |
|        |     |             |
|        |     |             |
|        |     |             |
|        |     |             -----------
|        |     |                       |
------------------------------------------------------------
```

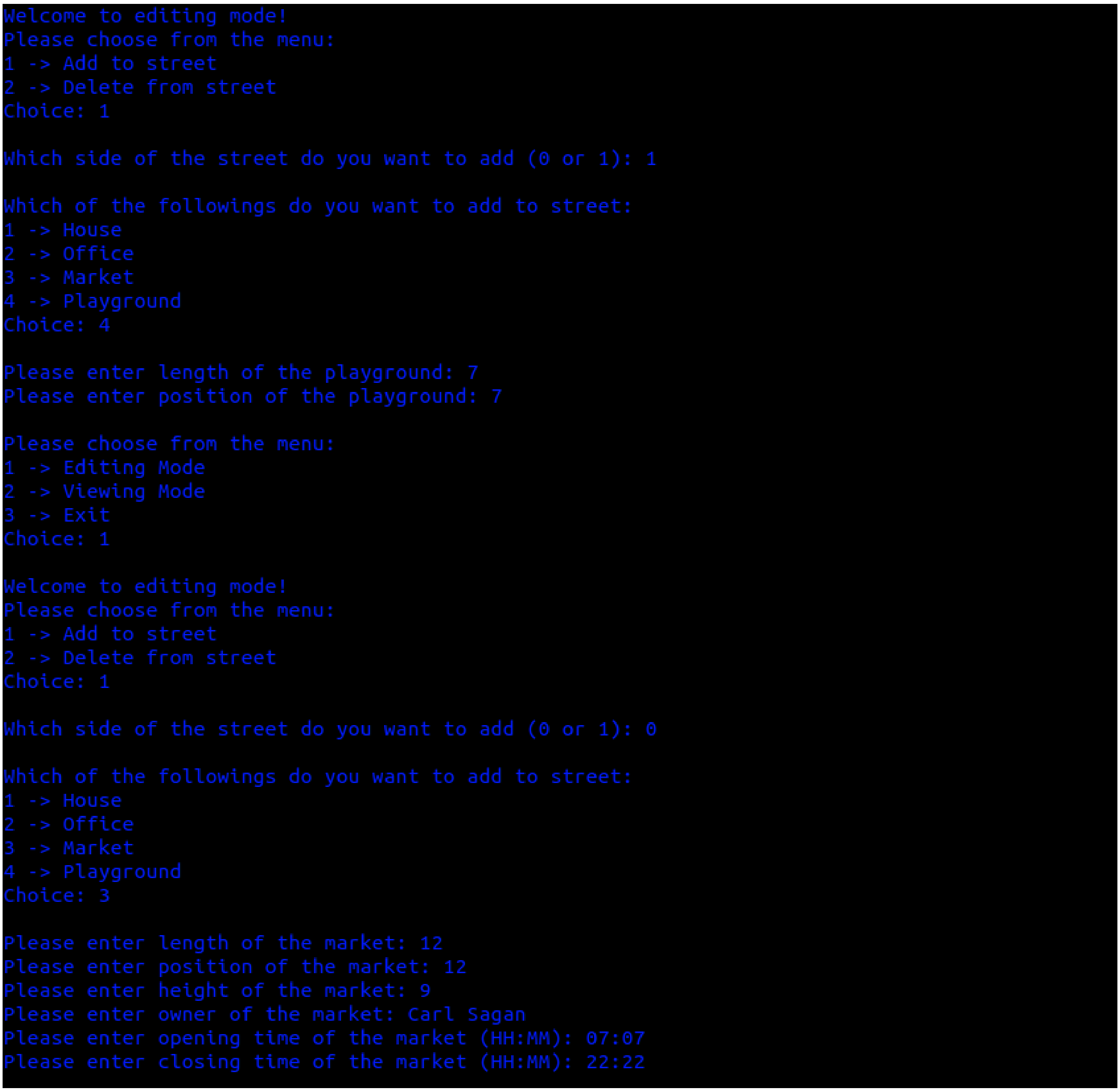


```
Total remaining length of lands on street: 80
Number of playgrounds in the street: 1
Ratio of playgrounds in the street: 0.083333
Total length of the street occupied by markets: 5
Total length of the street occupied by houses: 10
Total length of the street occupied by offices: 15

Now focus on the buildings on the street.
Erdogan Hoca's house's focus method:
     House's owner is -> Erdogan Hoca
Burak Hoca's market's focus method:
     Market's closing time is -> 21:00
Playground's focus method:
     Playground's length is -> 10
Feynman's office's focus method:
     Office's job type is -> Quantum Computing

Now let's try to add a building to a position that is not empty.
We will add to the side 0, position 0, to the place of house that belongs to Erdogan Hoca.
Exception: java.lang.Exception: That place is not empty.
This one gave an error as you see.


Now let's try to add a building to a position that exceeds the boundaries of the street.
We will add to the side 0, position 100.
Exception: java.lang.Exception: Building exceeds the limit of the street.
This one gave an error as you see.


Now let's try to add a building to an unproper side, for example to side 4.
Exception: java.lang.Exception: Unproper side.
This one gave an error as you see.


Now let's try to add a building that has unproper height like -5.
Exception: java.lang.Exception: Unproper height.
This one gave an error as you see.


Now let's try to delete an empty space.
We will try to delete position at the end of the side 1 which is empty.
Exception: java.lang.Exception: Attempting to delete empty space.
This one gave an error as you see.
```

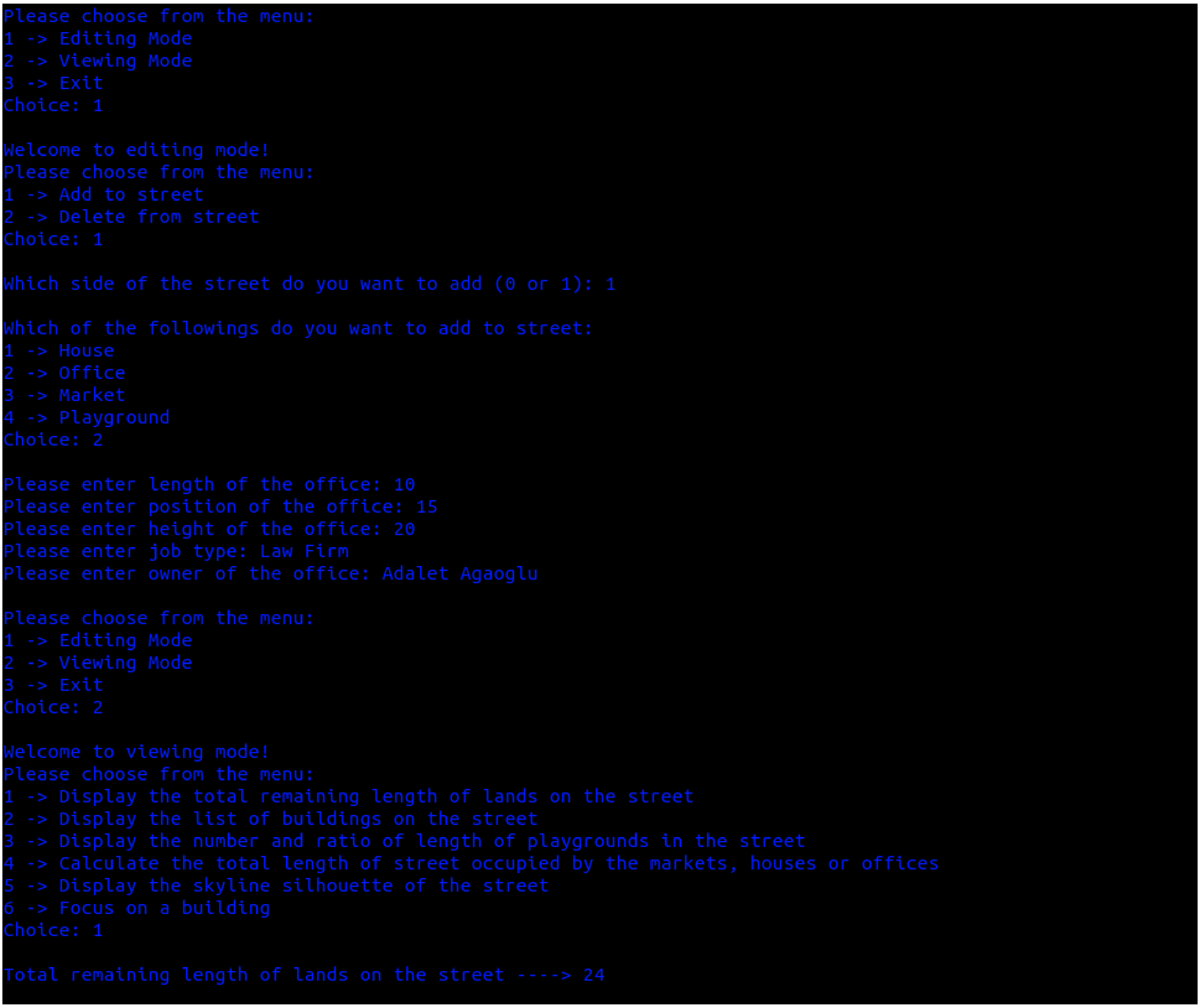
USER TESTS & MENU



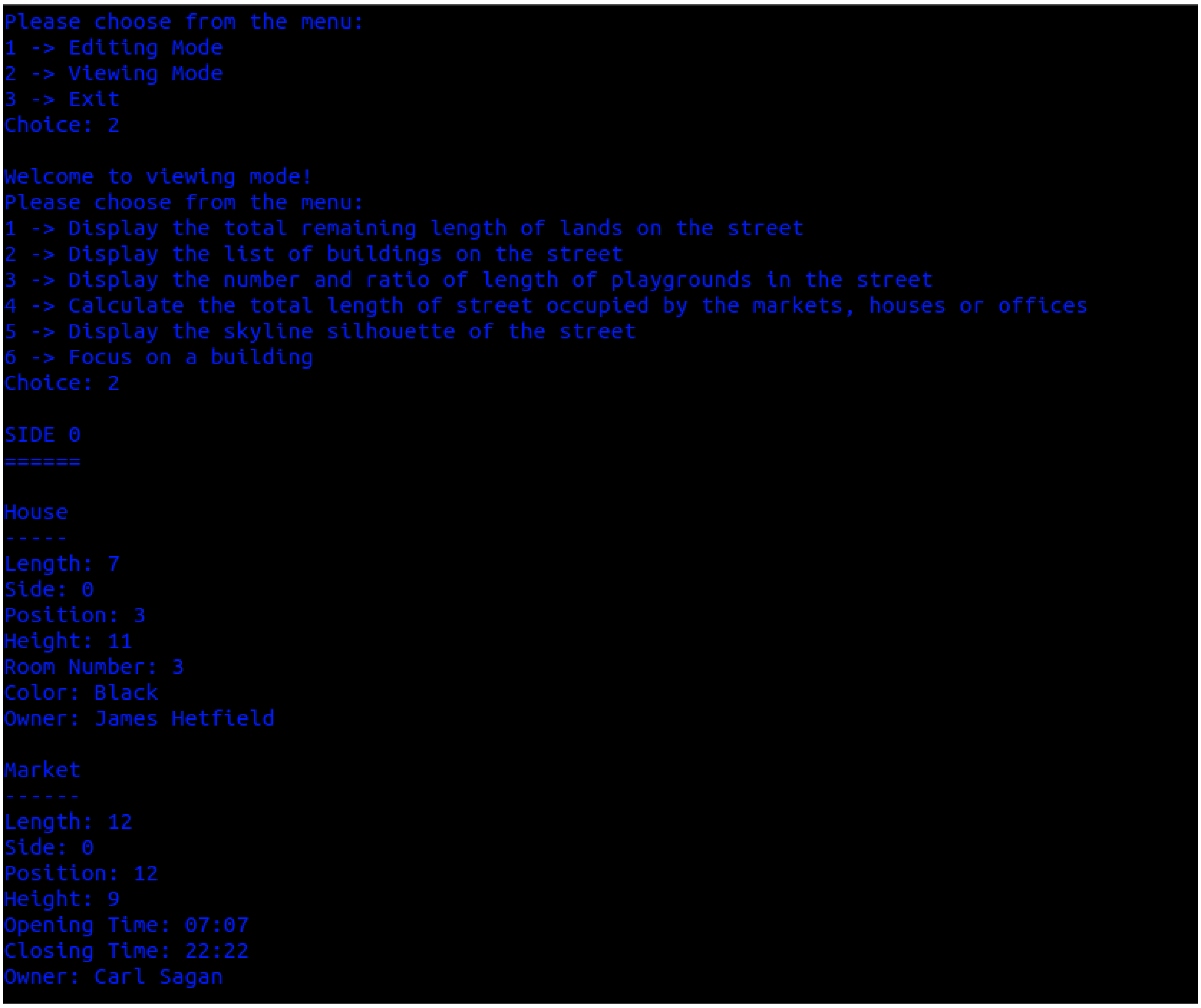
```
~~~NOW STREET OF BEAUTIFUL THINGS WELCOMES YOU~~~
-------------------------------------------------

Please enter length of the street: 30

Please choose from the menu:
1 -> Editing Mode
2 -> Viewing Mode
3 -> Exit
Choice: 1

Welcome to editing mode!
Please choose from the menu:
1 -> Add to street
2 -> Delete from street
Choice: test
Exception: java.util.InputMismatchException
Please try again.

Please choose from the menu:
1 -> Add to street
2 -> Delete from street
Choice: 1

Which side of the street do you want to add (0 or 1): 0

Which of the followings do you want to add to street:
1 -> House
2 -> Office
3 -> Market
4 -> Playground
Choice: 1

Please enter length of the house: 7
Please enter position of the house: 3
Please enter height of the house: 11
Please enter number of rooms of the house: 3
Please enter color of the house: Black
Please enter owner of the house: James Hetfield

Please choose from the menu:
1 -> Editing Mode
2 -> Viewing Mode
3 -> Exit
Choice: 1
```



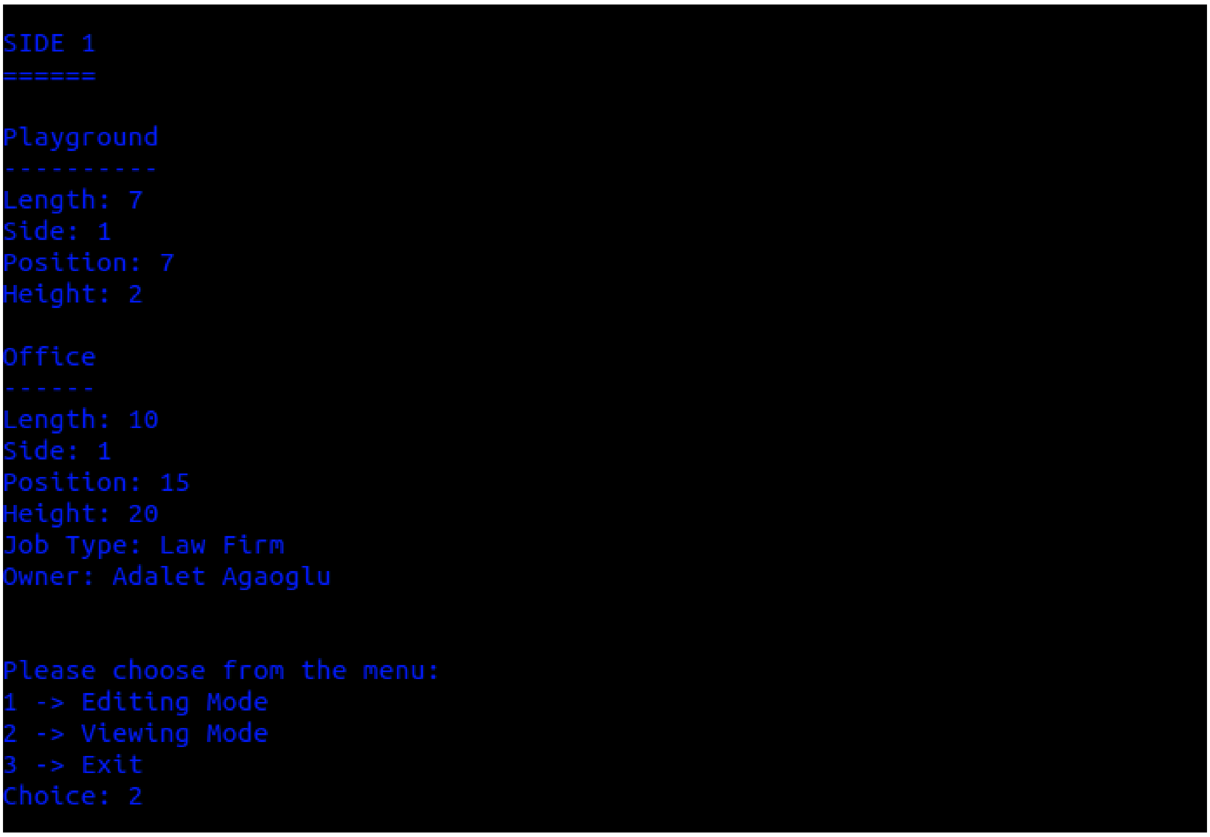
```
Welcome to editing mode!
Please choose from the menu:
1 -> Add to street
2 -> Delete from street
Choice: 1

Which side of the street do you want to add (0 or 1): 1

Which of the followings do you want to add to street:
1 -> House
2 -> Office
3 -> Market
4 -> Playground
Choice: 4

Please enter length of the playground: 7
Please enter position of the playground: 7

Please choose from the menu:
1 -> Editing Mode
2 -> Viewing Mode
3 -> Exit
Choice: 1

Welcome to editing mode!
Please choose from the menu:
1 -> Add to street
2 -> Delete from street
Choice: 1

Which side of the street do you want to add (0 or 1): 0

Which of the followings do you want to add to street:
1 -> House
2 -> Office
3 -> Market
4 -> Playground
Choice: 3

Please enter length of the market: 12
Please enter position of the market: 12
Please enter height of the market: 9
Please enter owner of the market: Carl Sagan
Please enter opening time of the market (HH:MM): 07:07
Please enter closing time of the market (HH:MM): 22:22
```



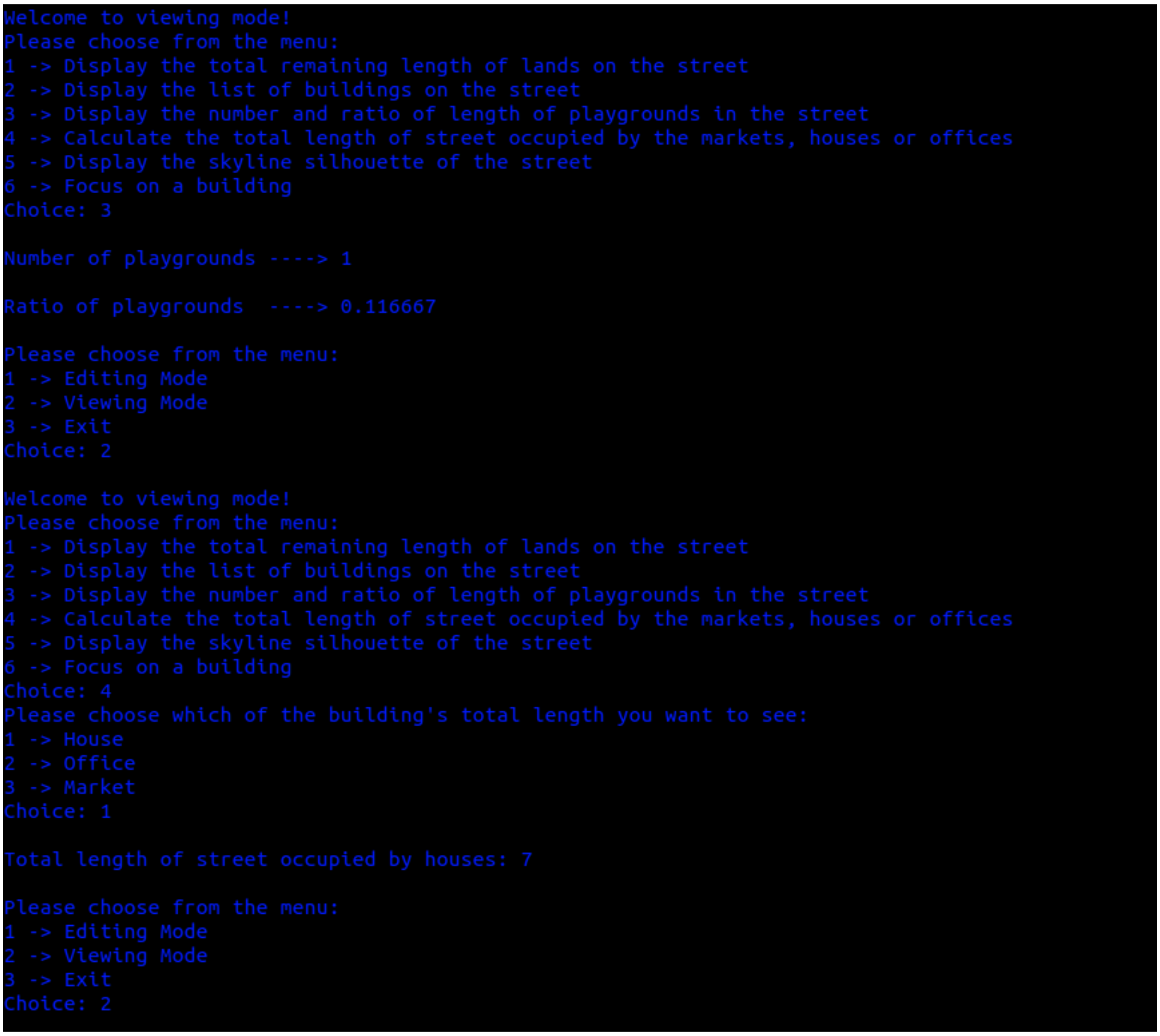
```
Please choose from the menu:
1 -> Editing Mode
2 -> Viewing Mode
3 -> Exit
Choice: 1

Welcome to editing mode!
Please choose from the menu:
1 -> Add to street
2 -> Delete from street
Choice: 1

Which side of the street do you want to add (0 or 1): 1

Which of the followings do you want to add to street:
1 -> House
2 -> Office
3 -> Market
4 -> Playground
Choice: 2

Please enter length of the office: 10
Please enter position of the office: 15
Please enter height of the office: 20
Please enter job type: Law Firm
Please enter owner of the office: Adalet Agaoglu

Please choose from the menu:
1 -> Editing Mode
2 -> Viewing Mode
3 -> Exit
Choice: 2

Welcome to viewing mode!
Please choose from the menu:
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 length of playgrounds in the street
4 -> Calculate the total length of street occupied by the markets, houses or offices
5 -> Display the skyline silhouette of the street
6 -> Focus on a building
Choice: 1

Total remaining length of lands on the street ----> 24
```



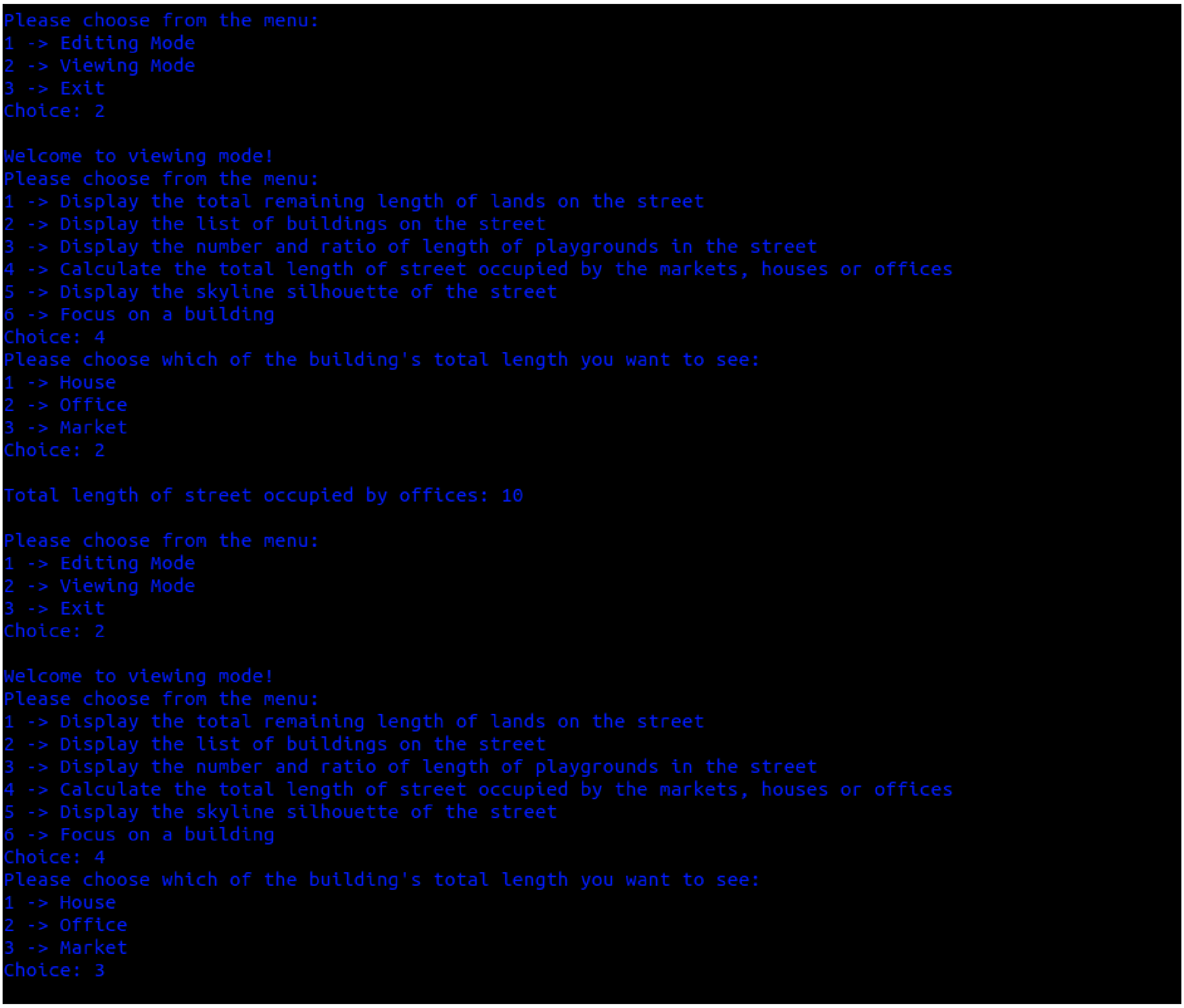
```
Please choose from the menu:
1 -> Editing Mode
2 -> Viewing Mode
3 -> Exit
Choice: 2

Welcome to viewing mode!
Please choose from the menu:
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 length of playgrounds in the street
4 -> Calculate the total length of street occupied by the markets, houses or offices
5 -> Display the skyline silhouette of the street
6 -> Focus on a building
Choice: 2

SIDE 0
======

House
-----
Length: 7
Side: 0
Position: 3
Height: 11
Room Number: 3
Color: Black
Owner: James Hetfield

Market
------
Length: 12
Side: 0
Position: 12
Height: 9
Opening Time: 07:07
Closing Time: 22:22
Owner: Carl Sagan
```



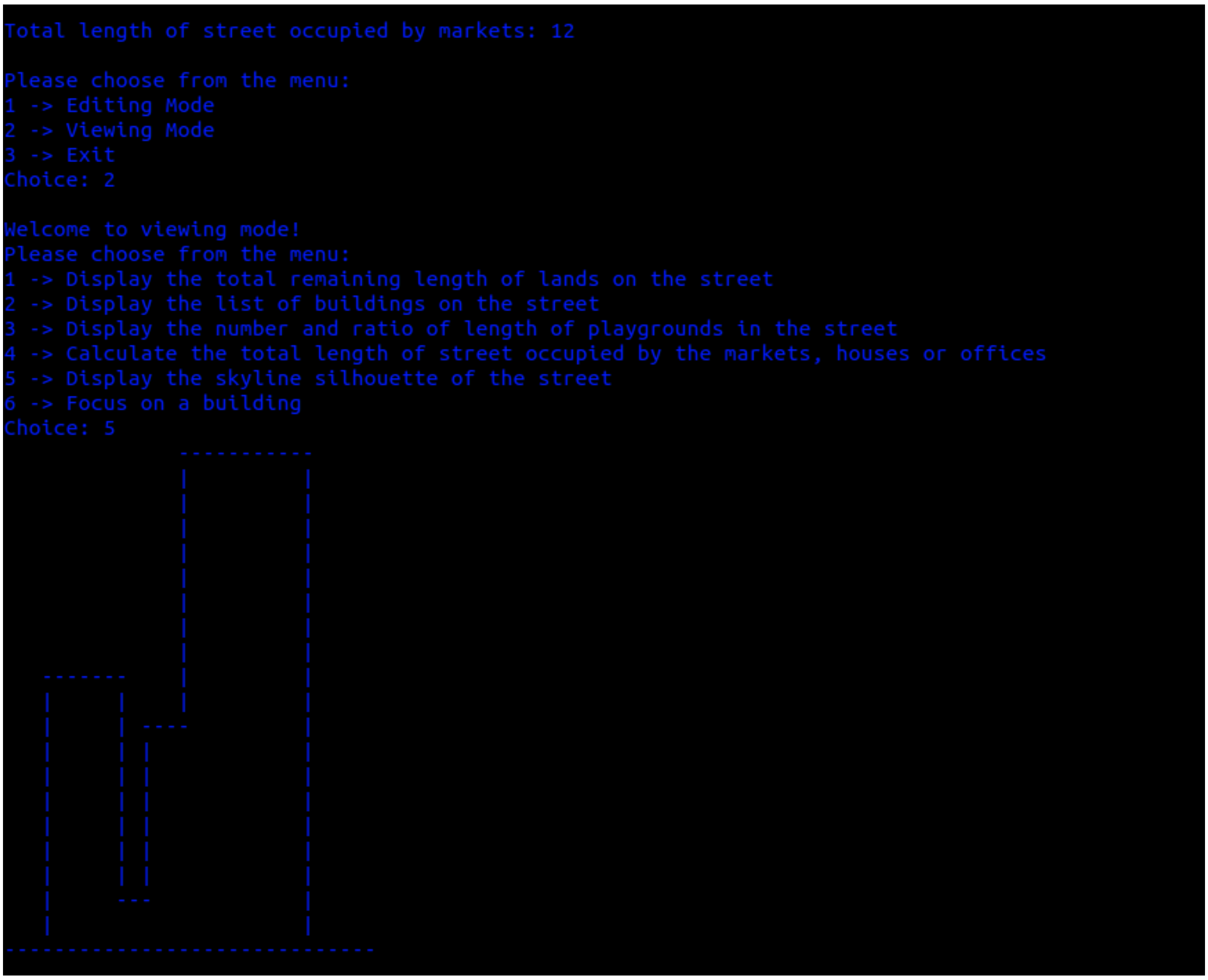
```
SIDE 1
======

Playground
----------
Length: 7
Side: 1
Position: 7
Height: 2

Office
------
Length: 10
Side: 1
Position: 15
Height: 20
Job Type: Law Firm
Owner: Adalet Agaoglu


Please choose from the menu:
1 -> Editing Mode
2 -> Viewing Mode
3 -> Exit
Choice: 2
```



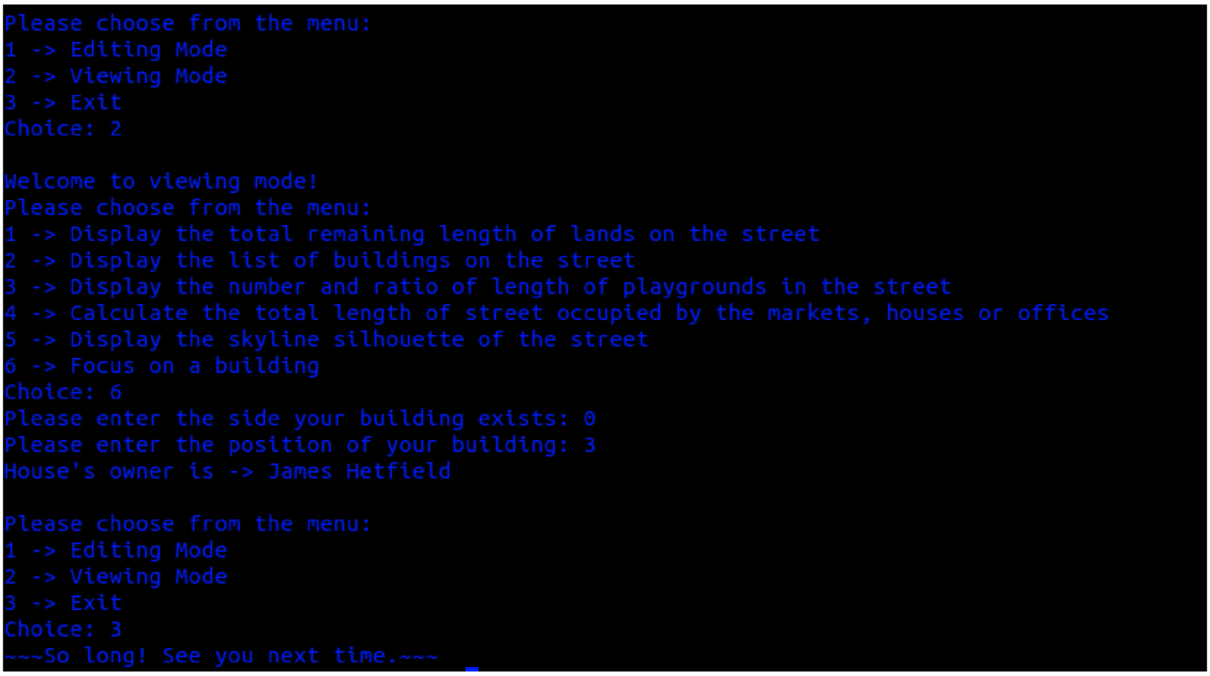
```
Welcome to viewing mode!
Please choose from the menu:
1 -> Display the total remaining length of lands on the street
2 -> Display the list of buildings on the street
3 -> Display the number and ratio of length of playgrounds in the street
4 -> Calculate the total length of street occupied by the markets, houses or offices
5 -> Display the skyline silhouette of the street
6 -> Focus on a building
Choice: 3

Number of playgrounds ----> 1

Ratio of playgrounds  ----> 0.116667

Please choose from the menu:
1 -> Editing Mode
2 -> Viewing Mode
3 -> Exit
Choice: 2

Welcome to viewing mode!
Please choose from the menu:
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 length of playgrounds in the street
4 -> Calculate the total length of street occupied by the markets, houses or offices
5 -> Display the skyline silhouette of the street
6 -> Focus on a building
Choice: 4
Please choose which of the building's total length you want to see:
1 -> House
2 -> Office
3 -> Market
Choice: 1

Total length of street occupied by houses: 7

Please choose from the menu:
1 -> Editing Mode
2 -> Viewing Mode
3 -> Exit
Choice: 2
```



```
Please choose from the menu:
1 -> Editing Mode
2 -> Viewing Mode
3 -> Exit
Choice: 2

Welcome to viewing mode!
Please choose from the menu:
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 length of playgrounds in the street
4 -> Calculate the total length of street occupied by the markets, houses or offices
5 -> Display the skyline silhouette of the street
6 -> Focus on a building
Choice: 4
Please choose which of the building's total length you want to see:
1 -> House
2 -> Office
3 -> Market
Choice: 2

Total length of street occupied by offices: 10

Please choose from the menu:
1 -> Editing Mode
2 -> Viewing Mode
3 -> Exit
Choice: 2

Welcome to viewing mode!
Please choose from the menu:
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 length of playgrounds in the street
4 -> Calculate the total length of street occupied by the markets, houses or offices
5 -> Display the skyline silhouette of the street
6 -> Focus on a building
Choice: 4
Please choose which of the building's total length you want to see:
1 -> House
2 -> Office
3 -> Market
Choice: 3
```



```
Total length of street occupied by markets: 12

Please choose from the menu:
1 -> Editing Mode
2 -> Viewing Mode
3 -> Exit
Choice: 2

Welcome to viewing mode!
Please choose from the menu:
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 length of playgrounds in the street
4 -> Calculate the total length of street occupied by the markets, houses or offices
5 -> Display the skyline silhouette of the street
6 -> Focus on a building
Choice: 5
              -----------
              |         |
              |         |
              |         |
              |         |
              |         |
              |         |
              |         |
              |         |
   -------    |         |
   |     |    |         |
   |     | ----         |
   |     | |            |
   |     | |            |
   |     | |            |
   |     | |            |
   |     | |            |
   |     | |            |
   |     ---            |
   |                    |
------------------------------
```



```
Please choose from the menu:
1 -> Editing Mode
2 -> Viewing Mode
3 -> Exit
Choice: 2

Welcome to viewing mode!
Please choose from the menu:
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 length of playgrounds in the street
4 -> Calculate the total length of street occupied by the markets, houses or offices
5 -> Display the skyline silhouette of the street
6 -> Focus on a building
Choice: 6
Please enter the side your building exists: 0
Please enter the position of your building: 3
House's owner is -> James Hetfield

Please choose from the menu:
1 -> Editing Mode
2 -> Viewing Mode
3 -> Exit
Choice: 3
~~~So long! See you next time.~~~
```