CSE 222/CSE505 SPRING 2022
HOMEWORK 1 REPORT

Burcu Sultan Orhan

1901042667

1- Detailed System Requirements

First there needs to be Street object to perform all the necesary things, whose length is set by the user.

```
public Street(int length){
    this.length1 = length;
    this.length2 = length;
    this.length = length;

    this.houses1 = new MyList<>();
    this.houses2 = new MyList<>();
    this.offices1 = new MyList<>();
    this.offices2 = new MyList<>();
    this.markets1 = new MyList<>();
    this.markets2 = new MyList<>();
    this.playgrounds1 = new MyList<>();
    this.playgrounds2 = new MyList<>();
```

Then the user can add buildings (houses, offices and markets) and playgrounds on each of the two sides of the Street.

```
public void addHouseSide1(House house){
```

```
public void addHouseSide2(House house){
```

These functions take house/Office/market/playground as parameter

```
public void addOfficeSide1(Office office){
```

```
public void addOfficeSide2(Office office){
```

```
public void addMarketSide1(Market market){
```

```
public void addMarketSide2(Market market){
```

```
public void addPlaygroundSide1(Playground playground){
```

```
public void addPlaygroundSide2(Playground playground){
```

Like these, user can delete buildings (houses, offices, markets) or playgrounds from the each two sides of the Street.

```
*/
  public void deleteHouseSide1(House house){
```

```
*/
  public void deleteHouseSide2(House house){
```

Likewise, these functions take buildings or playgrounds as parameter.

```
*/
   public void deleteOfficeSide1(Office office){
```

```
*/
  public void deleteOfficeSide2(Office office){
```

```
*/
  public void deleteMarketSide1(Market market){
```

```
*/
  public void deleteMarketSide2(Market market){
```

```
public void deletePlaygroundSide1(Playground playground){
```

```
*/
  public void deletePlaygroundSide2(Playground playground){
```

Then, user can see the total remaining length on the Street.

```
*/
   public int getRemainingLength(){
```

Then, user can see the list of the buildings on the Street.
This function takes the Street object as parameter

```
*/
  public static void displayBuildings(Street street){
```

User also can see the number and ratio of the length of lands occupied by playgrounds.

```
    public int getSizePlaygrounds1(){
        return(this.playgrounds1.size());
    }
/**
 * This function returns the number of playgrounds on side 2 of the street
 * @return
 */
    public int getSizePlaygrounds2(){
        return(this.playgrounds2.size());
    }
```

These functions returns the number of the playgrounds on each side of the Street. size() function returns the element number of the playgrounds lists.

```
 */
    public static double ratio(Street street){
```

And this function calculates the ratio of the length of lands occupied by playgrounds. It takes Street object as parameter.

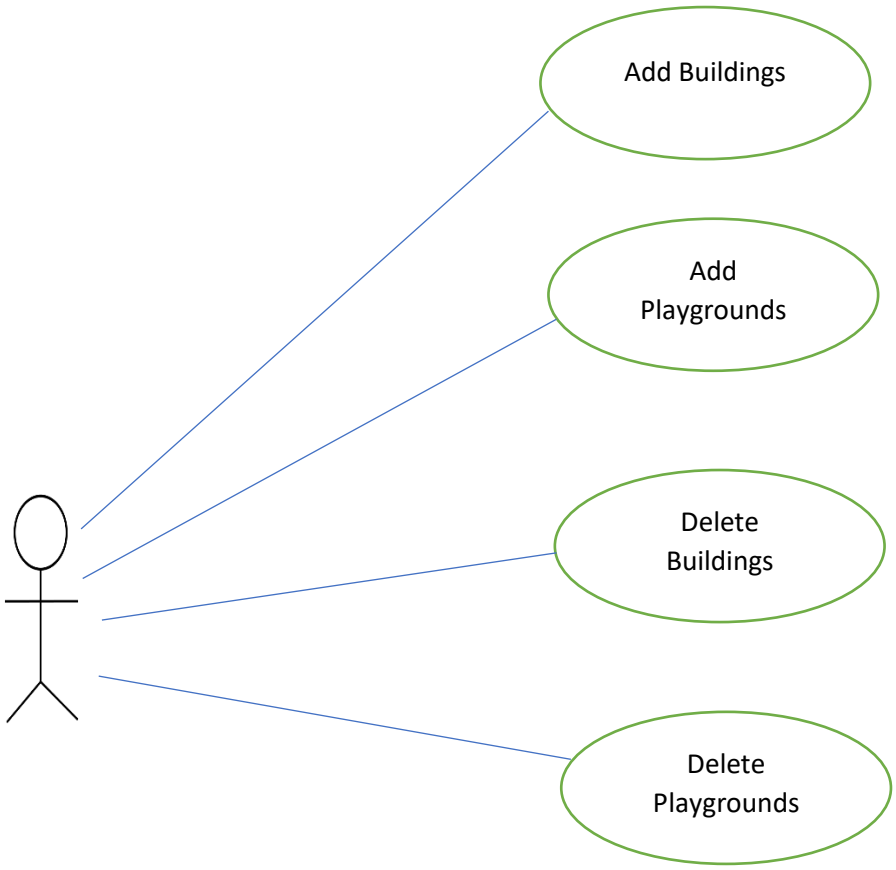
User also can see the total length of lands occupied by buildings (houses, offices and markets)

```
 */
    public int getTotal(){
```

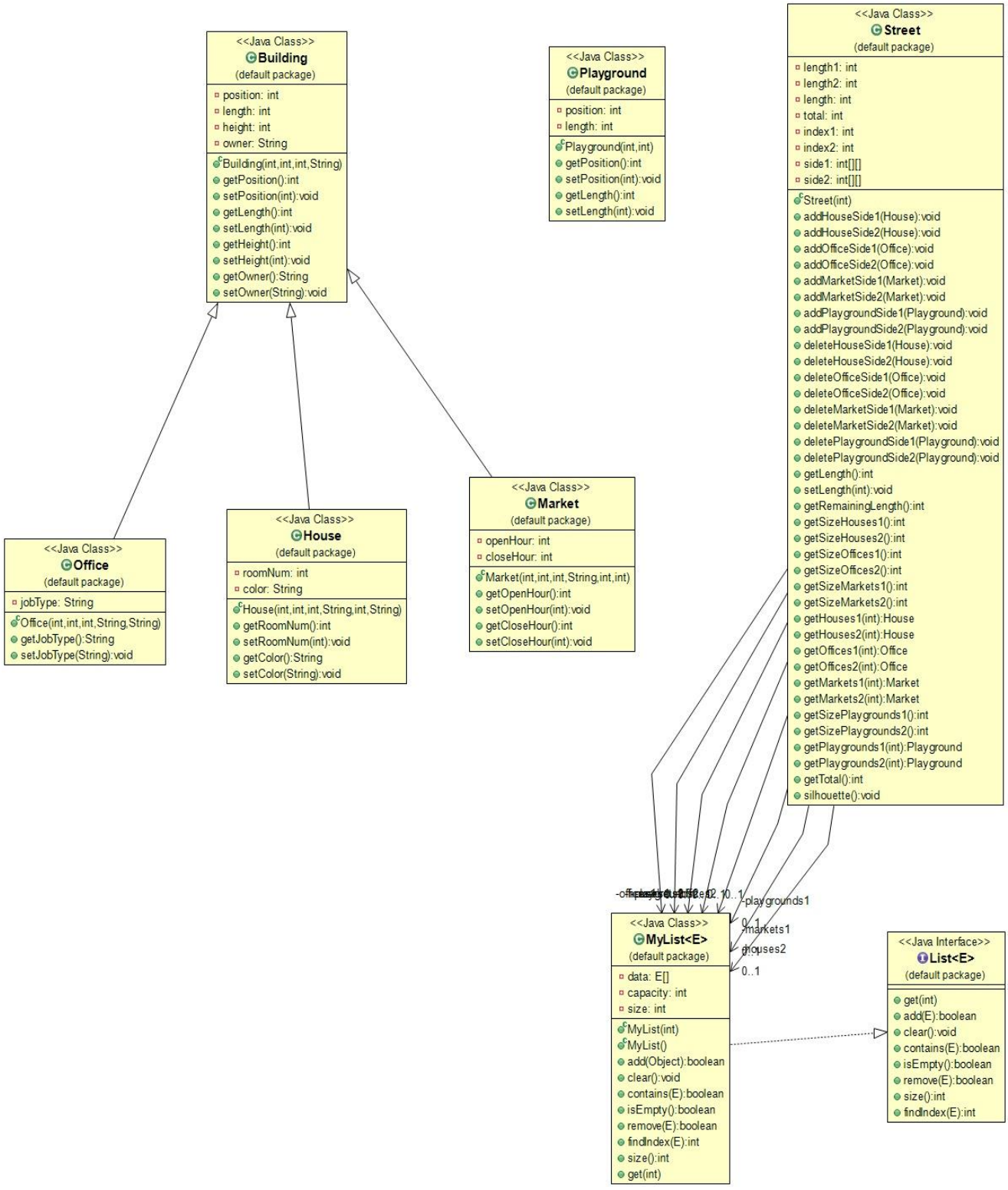
And finally, user can see the skyline silhouette of the Street

```
 */
    public void silhouette(){
```

2- Use Case Diagrams



3- Class Diagrams



4- Problem Solving Approach

I implemented my own version of lists using arrays, which helped me a lot when adding/deleting objects (as in houses, offices, markets, playgrounds). After that, I simplified the Street as two sides, I designed them completely independent because they had no business with each other, aside from the silhouette mode. As for silhouette mode, I collected every position, length and height info within two 2D arrays for both sides, and these were independent from the object arrays, and used these arrays when printing the silhouette.

5- Test Cases

```
Street driverStreet = new Street(30);
driverStreet.setLength(30);
House house1 = new House(5, 5, 10, "burcu", 5, "pink");
driverStreet.addHouseSide1(house1);
System.out.print("Total remaining length: ");
System.out.println(driverStreet.getRemainingLength());
displayBuildings(driverStreet);
System.out.print("Number of playgrounds: ");
System.out.println(driverStreet.getSizePlaygrounds1()+driverStreet.getSizePlaygrounds2());
System.out.print("Ratio of length of playgrounds: %");
System.out.println(ratio(driverStreet));
System.out.print("Total length occupied by buildings: ");
System.out.println(driverStreet.getTotal());
driverStreet.silhouette();
```

I created one Street, and added only one house. Showed all the functionality of all the methods.

```
Office office1 = new Office(8, 5, 15, "sultan", "advertising");
driverStreet.addOfficeSide2(office1);
System.out.print("Total remaining length: ");
System.out.println(driverStreet.getRemainingLength());
displayBuildings(driverStreet);
System.out.print("Number of playgrounds: ");
System.out.println(driverStreet.getSizePlaygrounds1()+driverStreet.getSizePlaygrounds2());
System.out.print("Ratio of length of playgrounds: %");
System.out.println(ratio(driverStreet));
System.out.print("Total length occupied by buildings: ");
System.out.println(driverStreet.getTotal());
driverStreet.silhouette();

Playground playground1 = new Playground(11, 3);
driverStreet.addPlaygroundSide1(playground1);
System.out.print("Total remaining length: ");
System.out.println(driverStreet.getRemainingLength());
displayBuildings(driverStreet);
System.out.print("Number of playgrounds: ");
System.out.println(driverStreet.getSizePlaygrounds1()+driverStreet.getSizePlaygrounds2());
System.out.print("Ratio of length of playgrounds: %");
System.out.println(ratio(driverStreet));
System.out.print("Total length occupied by buildings: ");
System.out.println(driverStreet.getTotal());
driverStreet.silhouette();
```

I added one new Office and one new market, then again, showed all the functionality of all the methods.

```
Market market1 = new Market(15, 5, 10, "orhan", 8, 22);
driverStreet.addMarketSide1(market1);
System.out.print("Total remaining length: ");
System.out.println(driverStreet.getRemainingLength());
displayBuildings(driverStreet);
System.out.print("Number of playgrounds: ");
System.out.println(driverStreet.getSizePlaygrounds1()+driverStreet.getSizePlaygrounds2());
System.out.print("Ratio of length of playgrounds: %");
System.out.println(ratio(driverStreet));
System.out.print("Total length occupied by buildings: ");
System.out.println(driverStreet.getTotal());
driverStreet.silhouette();

driverStreet.deleteOfficeSide1(office1);
System.out.print("Total remaining length: ");
System.out.println(driverStreet.getRemainingLength());
displayBuildings(driverStreet);
System.out.print("Number of playgrounds: ");
System.out.println(driverStreet.getSizePlaygrounds1()+driverStreet.getSizePlaygrounds2());
System.out.print("Ratio of length of playgrounds: %");
System.out.println(ratio(driverStreet));
System.out.print("Total length occupied by buildings: ");
System.out.println(driverStreet.getTotal());
driverStreet.silhouette();
```

Lastly, I added one market and after that, deleted one Office. I showed all the functionality of all the methods.

6- Running Command and Results

First, program runs driver and Works initialized methods.

```
eview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\burcu\Desktop\ilk\bin' 'Driver'
House added successfully!
Total remaining length: 55
Houses on side 1 of the street:
1- 5 5 10 burcu
Houses on side 2 of the street:
Offices on side 1 of the street:
Offices on side 2 of the street:
Markets on side 1 of the street:
Markets on side 2 of the street:
Number of playgrounds: 0
Ratio of length of playgrounds: %0.0
Total length occupied by buildings: 5
     _____
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
```

```
Office added successfully!
Total remaining length: 50
Houses on side 1 of the street:
1- 5 5 10 burcu
Houses on side 2 of the street:
Offices on side 1 of the street:
Offices on side 2 of the street:
1- 8 5 15 sultan
Markets on side 1 of the street:
Markets on side 2 of the street:
Number of playgrounds: 0
Ratio of length of playgrounds: %0.0
Total length occupied by buildings: 10
```

```
Playground added successfully!
Total remaining length: 47
Houses on side 1 of the street:
1- 5 5 10 burcu
Houses on side 2 of the street:
Offices on side 1 of the street:
Offices on side 2 of the street:
1- 8 5 15 sultan
Markets on side 1 of the street:
Markets on side 2 of the street:
Number of playgrounds: 1
Ratio of length of playgrounds: %5.0
Total length occupied by buildings: 10
```

```
Market added successfully!
Total remaining length: 42
Houses on side 1 of the street:
1- 5 5 10 burcu
Houses on side 2 of the street:
Offices on side 1 of the street:
Offices on side 2 of the street:
1- 8 5 15 sultan
Markets on side 1 of the street:
1- 15 5 10 orhan
Markets on side 2 of the street:
Number of playgrounds: 1
Ratio of length of playgrounds: %5.0
Total length occupied by buildings: 15
```

And when the program is done with it, it directs to the menu, and gets interactive with the user