

Street.java

```
1 package game.fields;
2
3 import java.util.Arrays;
4
5
6
7
8
9 public class Street extends Ownable {
10
11     public enum Group {BLUE, PINK, GREEN, GREY, RED, WHITE, YELLOW, PURPLE}
12     private Group group;
13     public static final int MAX_NUMBER_OF_BUILDINGS = 5;
14     private int buildings;
15     private int buildingPrice;
16     int[] buildingRents;
17
18     public Street(String title, int price, Group group) {
19         super(title, price);
20         this.group = group;
21         this.buildings = 0;
22         buildingRents = new int[MAX_NUMBER_OF_BUILDINGS+1];
23         setBuildingRentsFromProperties();
24     }
25
26     private void setBuildingRentsFromProperties(){
27         Properties props = getFieldProperties();
28         for (int i = 0;i<buildingRents.length;i++){
29             String key = title + "Rent"+ String.valueOf(i);
30             String propString = props.getProperty(key);
31             if (propString!=null) {
32                 int rent = Integer.parseInt(propString);
33                 buildingRents[i] = rent;
34             } else {
35                 System.err.println("No property for " + key);
36             }
37         }
38         buildingPrice = Integer.parseInt(props.getProperty(title +
39 "BuildingPrice"));
40     }
41
42
43     public String[] decoratorMessage(Player p){
44         if (owner==null){
45             return StringTools.add(super.decoratorMessage(p),new String[]
46 {"RentMessage", String.valueOf(getBaseRent()),"InRent"});
47         }
48         return super.decoratorMessage(p);
49     }
50
51
52     public Group getGroup() {
```

Street.java

```
53     return group;
54 }
55
56
57 public void setGroup(Group group) {
58     this.group = group;
59 }
60
61
62 @Override
63 public String toString() {
64     return "Street [group=" + group + ", buildings=" + buildings
65         + ", buildingRents=" + Arrays.toString(buildingRents)
66         + ", price=" + price + ", owner=" + owner + ", pawned="
67         + pawned + ", title=" + title + "]";
68 }
69
70
71 @Override
72 public int getRent() {
73     return buildingRents[buildings];
74 }
75
76
77
78 public int getBuildings() {
79     return buildings;
80 }
81 private void setBuildings(int i) {
82     this.buildings = i;
83 }
84 public void addBuilding(){
85     setBuildings(getBuildings() + 1);
86 }
87 public void removeBuilding(){
88     setBuildings(getBuildings() - 1);
89 }
90 public static void main(String[] args){
91     Street testStreet = new Street("Rødovrevej", 10000, Group.BLUE);
92     testStreet.setBuildingRentsFromProperties();
93     System.out.println(testStreet.toString());
94 }
95
96 @Override
97 public int getBaseRent() {
98     return buildingRents[0];
99 }
100 public int getBuildingPrice(){
101     return buildingPrice;
102 }
```

Street.java

```
103
104     public int getBuildingSellValue() {
105         int buildingSellValue = (buildingPrice/2)*buildings;
106         return buildingSellValue;
107     }
108     public int getBuildingBuyValue(){
109         return buildingPrice*buildings;
110     }
111
112 }
113
```