

```
1 namespace RepasoCurso
2 {
3     public class Community
4     {
5         private class Person
6         {
7             private int _age;
8             private string _name;
9         }
10
11        private class Property
12        {
13            private int _floor;
14            private char _letter = ' ';
15            private List<Person> _listPersons = new List<Person>();
16
17            public int Floor
18            {
19                get { return _floor; }
20                set { _floor = value; }
21            }
22
23            public char Letter
24            {
25                get { return _letter; }
26                set { _letter = value; }
27            }
28
29            public int PersonsCount => _listPersons.Count;
30
31            public List<Person> CloneList()
32            {
33                List<Person> cloneList = new List<Person>();
34                for(int i = 0; i < _listPersons.Count; i++)
35                {
36                    cloneList[i] = _listPersons[i];
37                }
38                return cloneList;
39            }
40            public Property Clone()
41            {
42                Property clone = new Property();
43                clone.Floor = _floor;
44                clone.Letter = _letter;
45                clone._listPersons = _listPersons;
46
47                return clone;
48            }
49        }
50
51        private class Community
52        {
53            private List<Property> _listProperties = new List<Property> ➤
```

```
    );
54
55     public int PropertiesCount => _listProperties.Count;
56     public int PersonsCount
57     {
58         get
59         {
60             int result = 0;
61             for (int i = 0; i < _listProperties.Count; i++)
62             {
63                 result += _listProperties[i].PersonsCount;
64             }
65             return result;
66         }
67     } // TODO predicate ALL
68
69     public void AddProperty(Property property)
70     {
71         if (property == null)
72             return;
73
74         _listProperties.Add(property);
75     }
76
77     public void RemovePropertyAt(int index)
78     {
79
80     }
81
82     public void Clear()
83     {
84         _listProperties = new List<Property>();
85     }
86
87     public List<Property> Clone()
88     {
89         List<Property> clone = new List<Property>();
90
91         for(int i = 0; i < _listProperties.Count; i++)
92         {
93             clone[i] = _listProperties[i].Clone();
94         }
95         return clone;
96     }
97 }
98 }
99 }
100
```