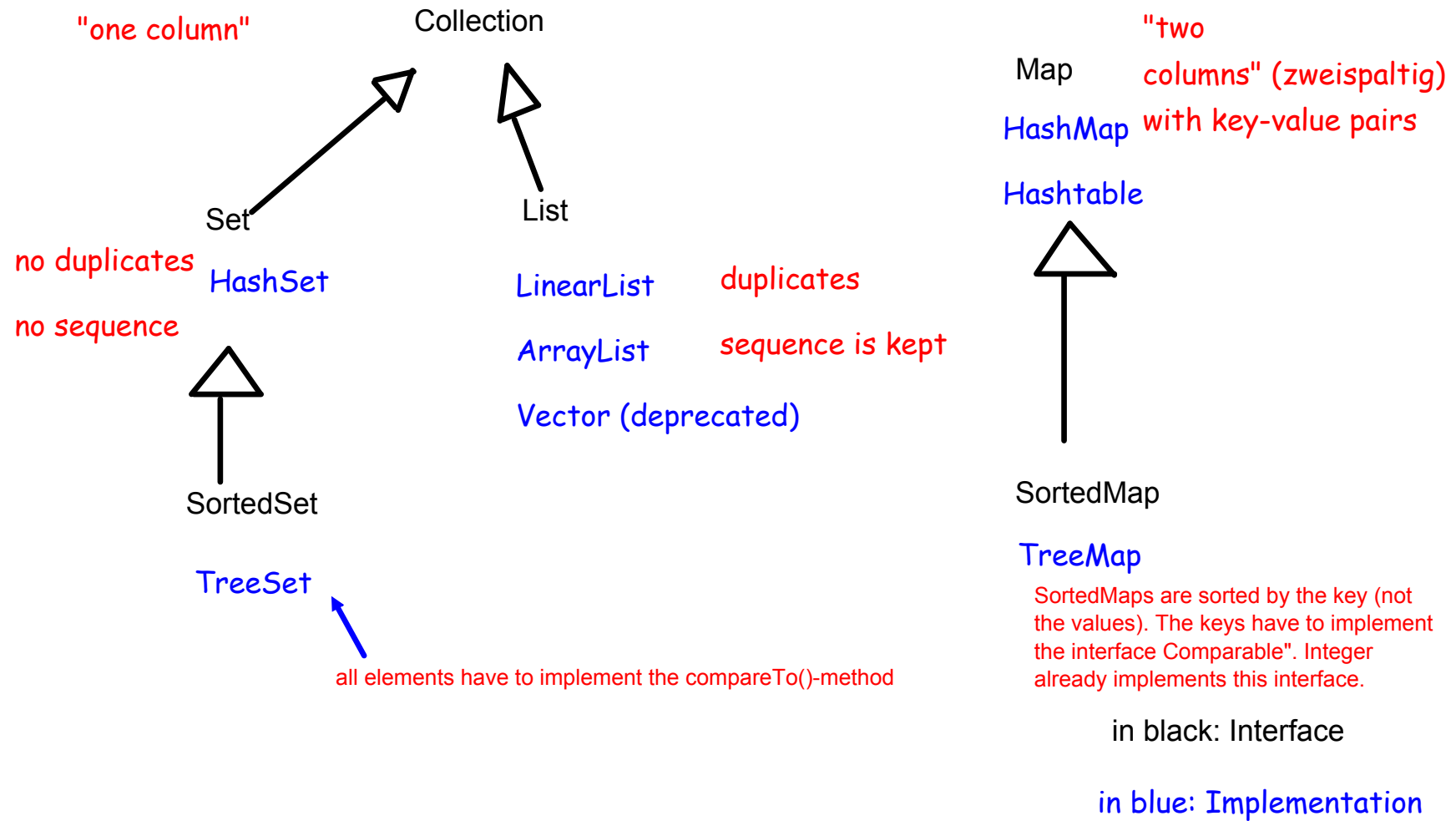


Java Collection Framework



```
Company.java Client.java StartClass.java ✖
1 package uml2code;
2
3 public class StartClass {
4     public static void main(String[] args) {
5         Client justin = new Client("Justin", 1);
6         Client alex = new Client("Alex", 2);
7
8         Company aldi = new Company("Aldi");
9         aldi.addClient(justin);
10        aldi.addClient(alex);
11
12        justin.setCompany(aldi);
13        alex.setCompany(aldi);
14
15        for(Client client: aldi.getClients()) {
16            System.out.println("Aldi has Client " + client.getClientName() -
17        }
18
19        // aldi.deleteClient(alex);
20        // for(Client client: aldi.getClients()) {
21        //     System.out.println("Aldi has Client " + client.getClientName() -
22        // }
23    }
24 }
25
```

```

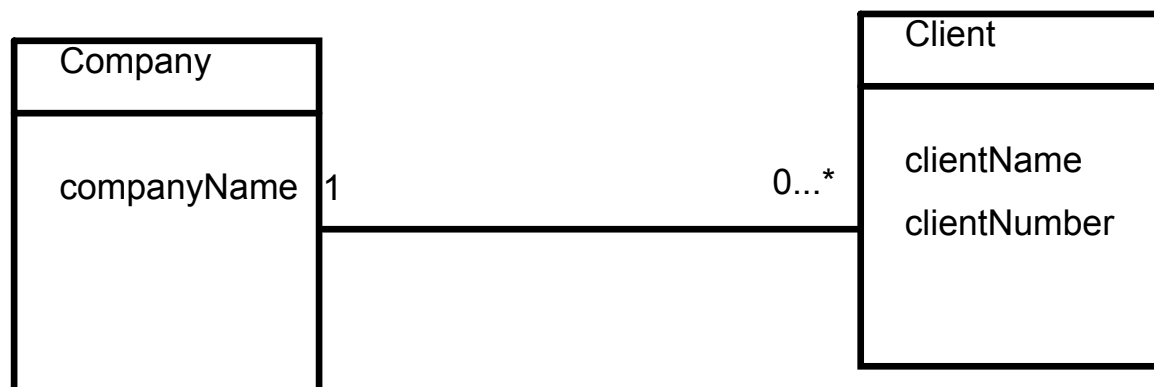
Company.java Client.java StartClass.java
1 package uml2code;
2
3 import java.util.Collection;
4 import java.util.TreeSet;
5
6 public class Company {
7     private String companyName;
8     private Collection<Client> clients = new TreeSet<Client>();
9
10    public Company(String companyName) {
11        super();
12        this.companyName = companyName;
13    }
14
15    public void addClient(Client c) {
16        clients.add(c);
17    }
18
19    public boolean doesCompanyHaveClient(Client c) {
20        return clients.contains(c);
21    }
22
23    public Collection<Client> getClients() {
24        return clients;
25    }
26
27    public void deleteClient(Client c) {
28        clients.remove(c);
29    }
30
31    public String getCompanyName() {
32        return companyName;
33    }
34

```

```

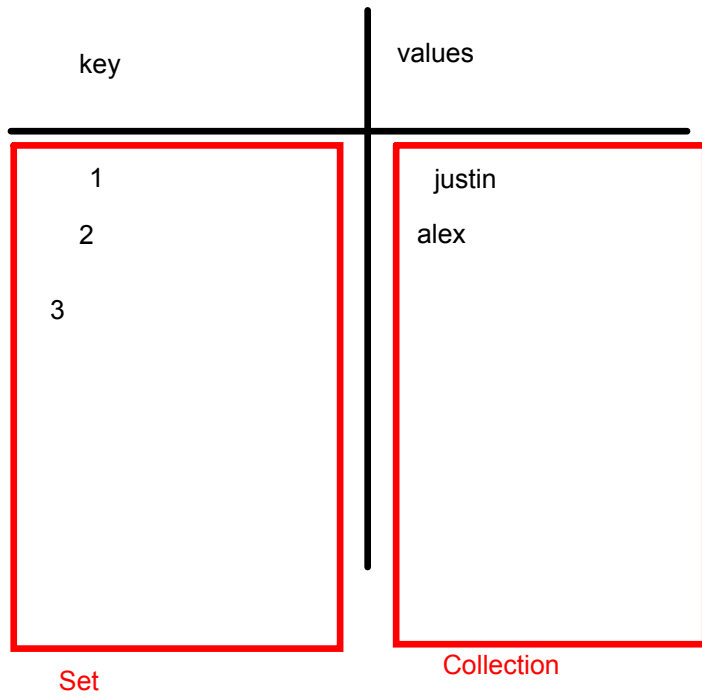
Company.java Client.java StartClass.java
1 package uml2code;
2
3 public class Client implements Comparable<Client> {
4     private String clientName;
5     private int clientNumber;
6     private Company company;
7
8     public Client(String clientName, int number) {
9         super();
10        this.clientName = clientName;
11        this.clientNumber = number;
12    }
13
14    public String getClientName() {
15        return clientName;
16    }
17
18    public void setClientName(String clientName) {
19        this.clientName = clientName;
20    }
21
22    public Company getCompany() {
23        return company;
24    }
25
26    public void setCompany(Company company) {
27        this.company = company;
28    }
29
30    public int getClientNumber() {
31        return clientNumber;
32    }
33

```



```
1 package uml2code;
2
3 public class Client implements Comparable<Client> {
4     private String clientName;
5     private int clientNumber;
6     private Company company;
7
8     public Client(String clientName, int number) {}
9
10
11
12
13     public String getClientName() {}
14
15
16
17     public void setClientName(String clientName) {}
18
19
20
21     public Company getCompany() {}
22
23
24
25     public void setCompany(Company company) {}
26
27
28
29     public int getClientNumber() {}
30
31
32
33     public void setClientNumber(int clientNumber) {}
34
35
36
37
38     @Override
39     public int compareTo(Client arg0) {
40         //return -1 * (this.clientNumber - arg0.getClientNumber());
41
42         return -1 * (this.clientName.compareTo(arg0.getClientName()));
43     }
44
45
46 }
47
```

If the instance of a class (here Client) are supposed to be used in a SortedSet, then this class has to implement Comparable and thereby the method compareTo()



API of the Java Collection Framework

Collection

- void add(Object o)
- void remove(Object o)
- boolean contains(Object o)
- int size()

List (additional methods)

- void add(int index, Object o)
- void remove(int index)
- Object get(int index)

Map

- void put(Object key, Object value)
- Object get(Object key)
- void remove(Object key)
- boolean containsKey(Object key)
- boolean containsValue(Object value)
- Collection values()
- Set keySet()