Container Classes

Sets, Maps

COMP2603
Object Oriented Programming 1

Week 10

Outline

- Java Collections Framework
- Collection Interface ✓
  - List Interface
    - Linked List ✓
    - ArrayList ✓
    - Vector ✓
  - Set Interface ✓
    - SortedSet ✓
  - Map
    - SortedMap

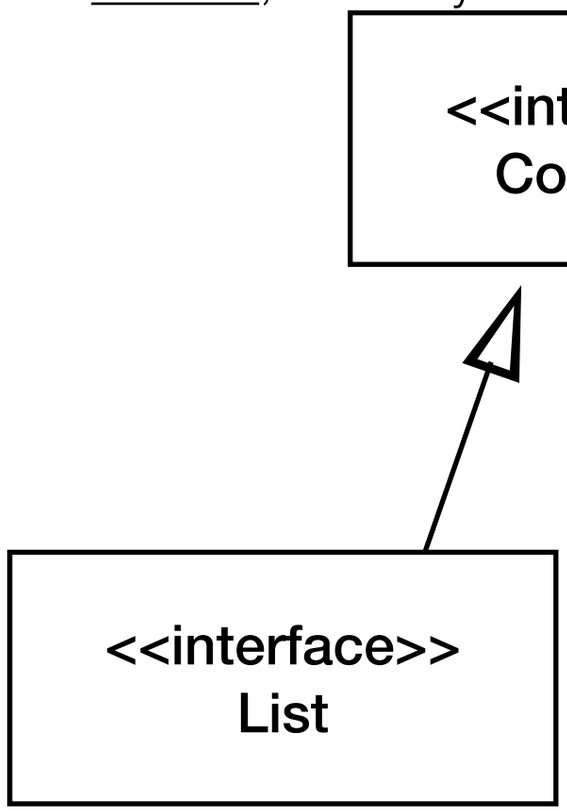
Interfaces in the Java Collections Framework

Interfaces in the Java Collections Framework

A group or collection of objects that may or may not be ordered, and may or may not contain duplicate objects.

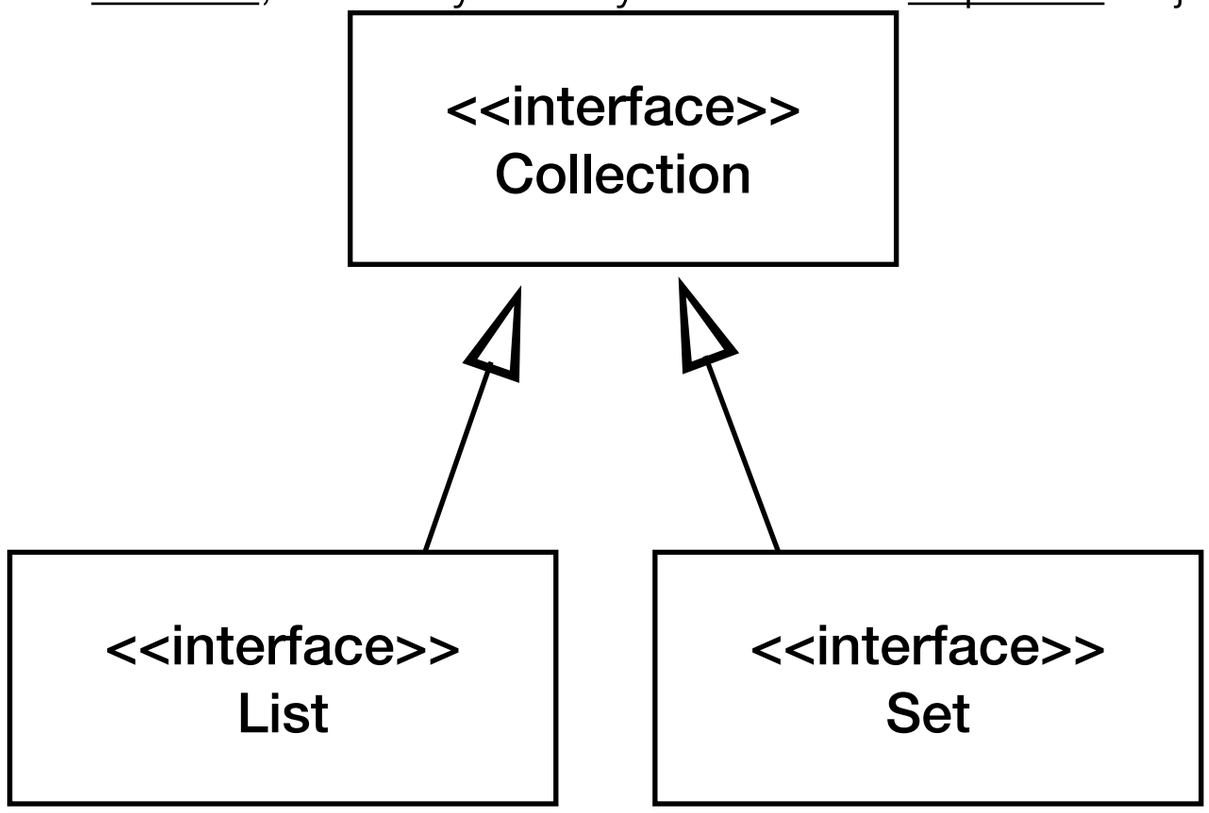
Interfaces in the Java Collections Framework

A group or collection of objects that may or may not be ordered, and may or may not contain duplicate objects.



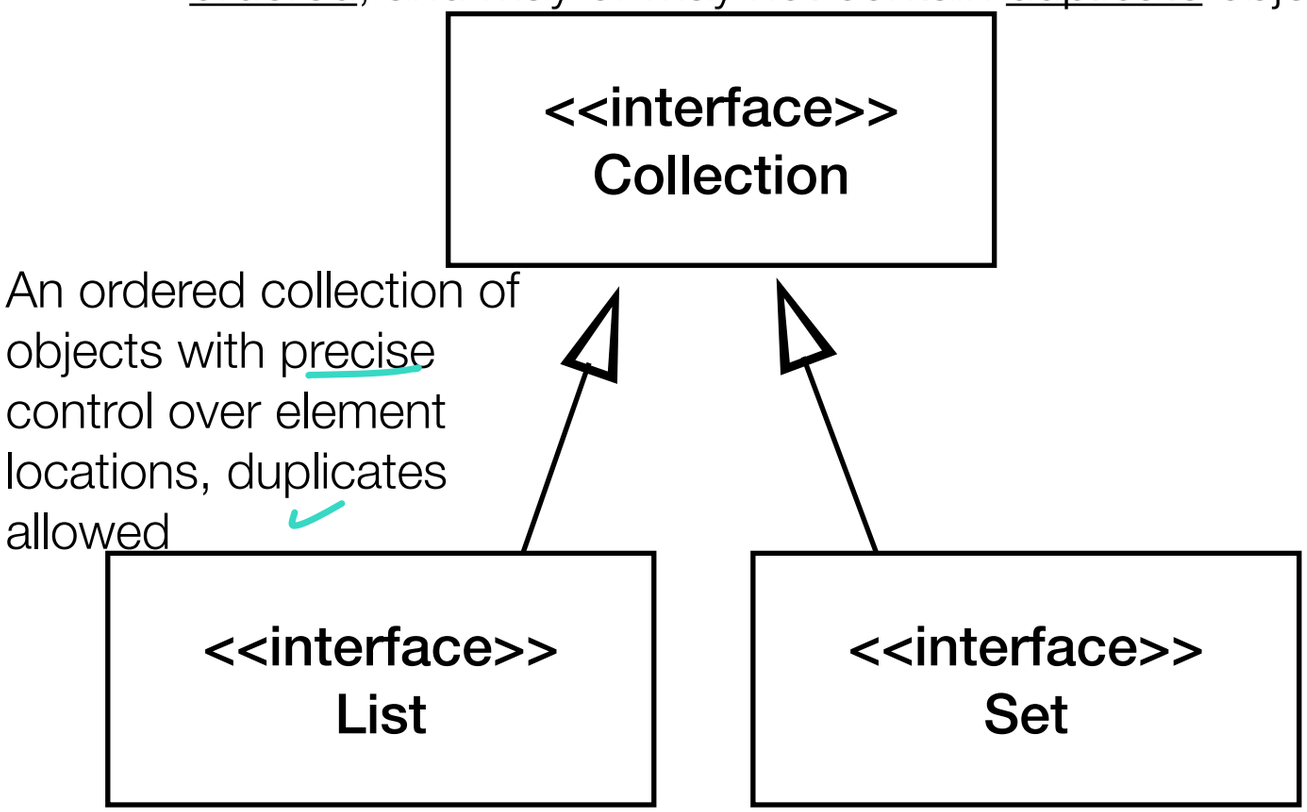
Interfaces in the Java Collections Framework

A group or collection of objects that may or may not be ordered, and may or may not contain duplicate objects.



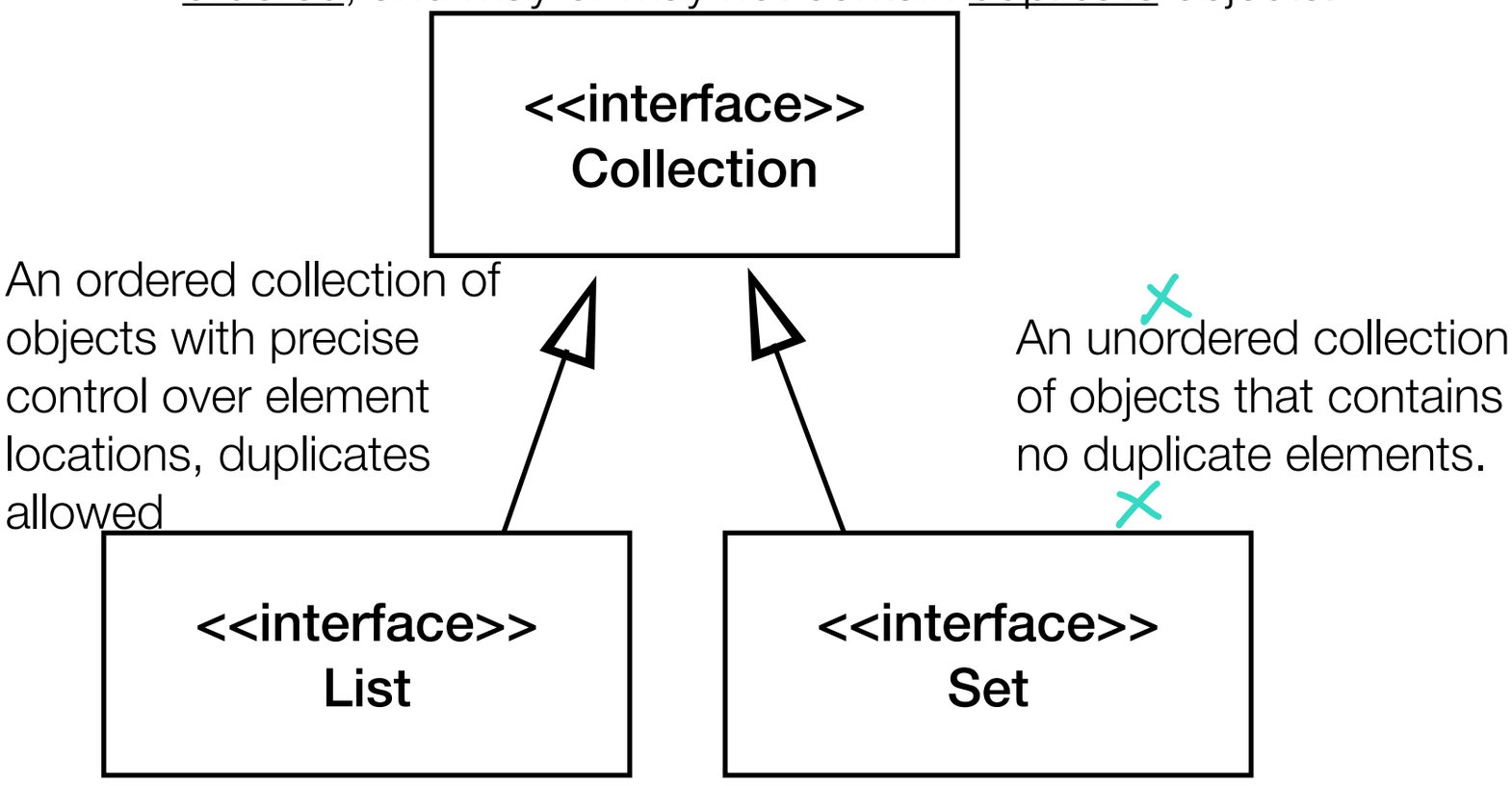
Interfaces in the Java Collections Framework

A group or collection of objects that may or may not be ordered, and may or may not contain duplicate objects.



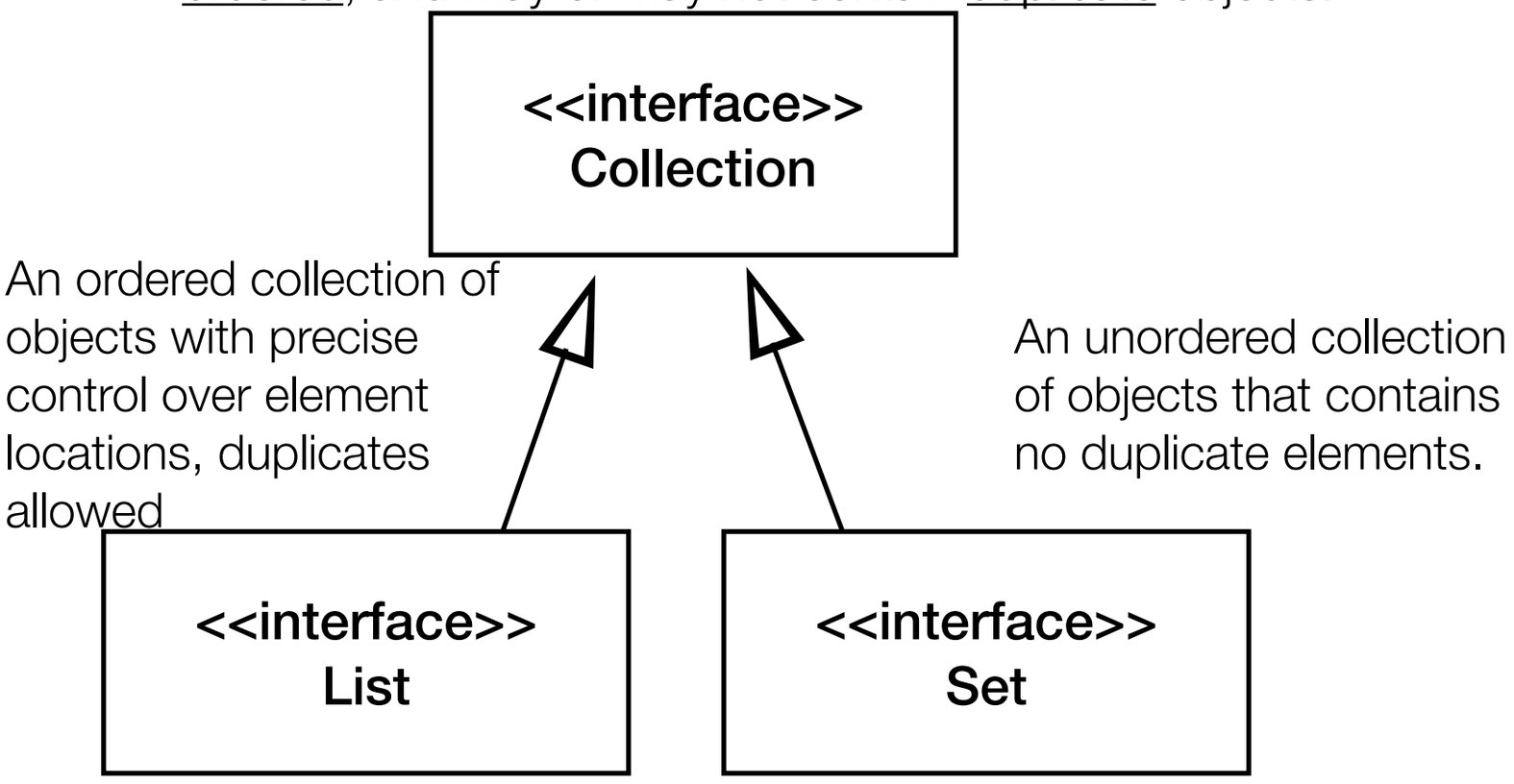
Interfaces in the Java Collections Framework

A group or collection of objects that may or may not be ordered, and may or may not contain duplicate objects.



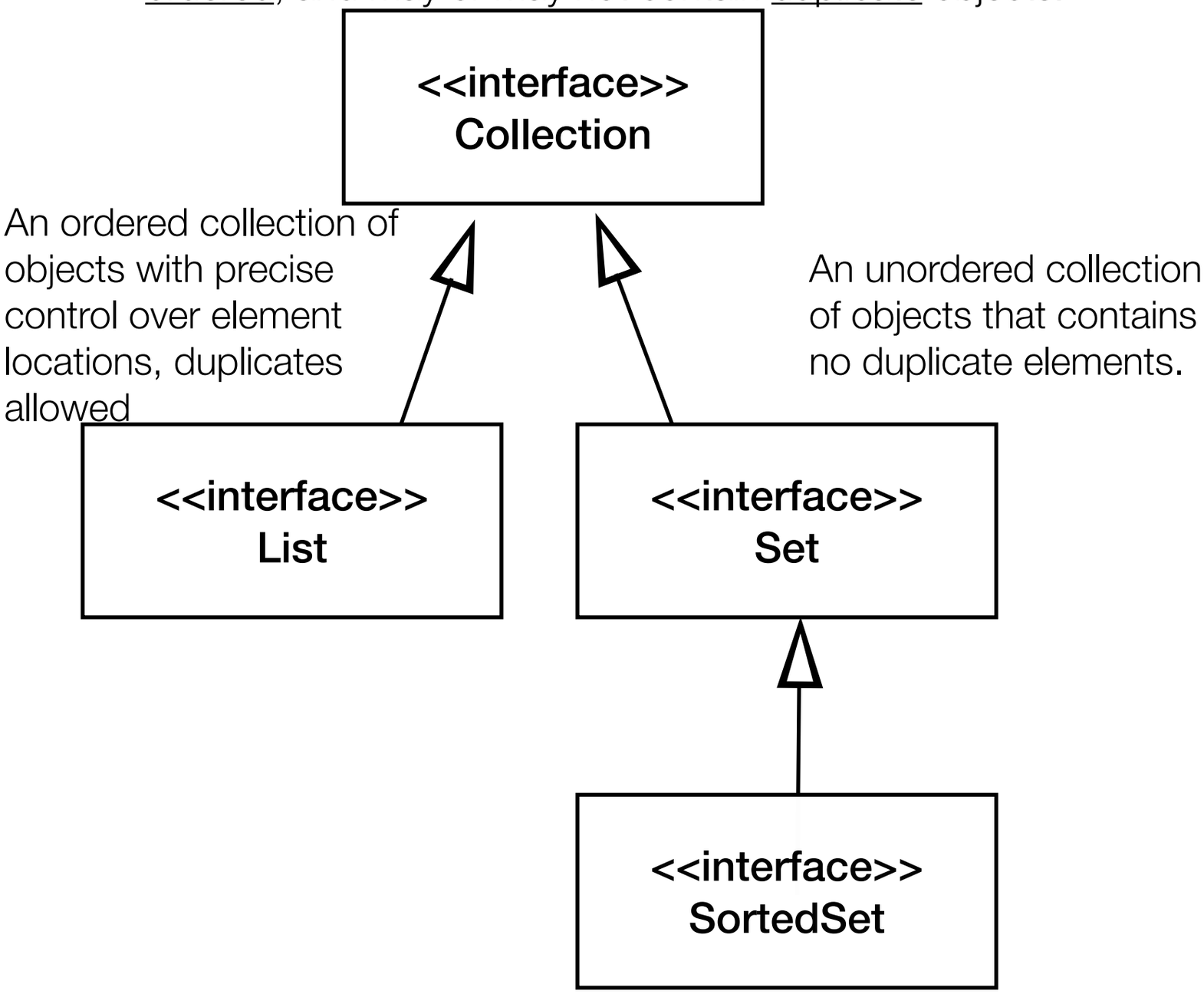
Interfaces in the Java Collections Framework

A group or collection of objects that may or may not be ordered, and may or may not contain duplicate objects.



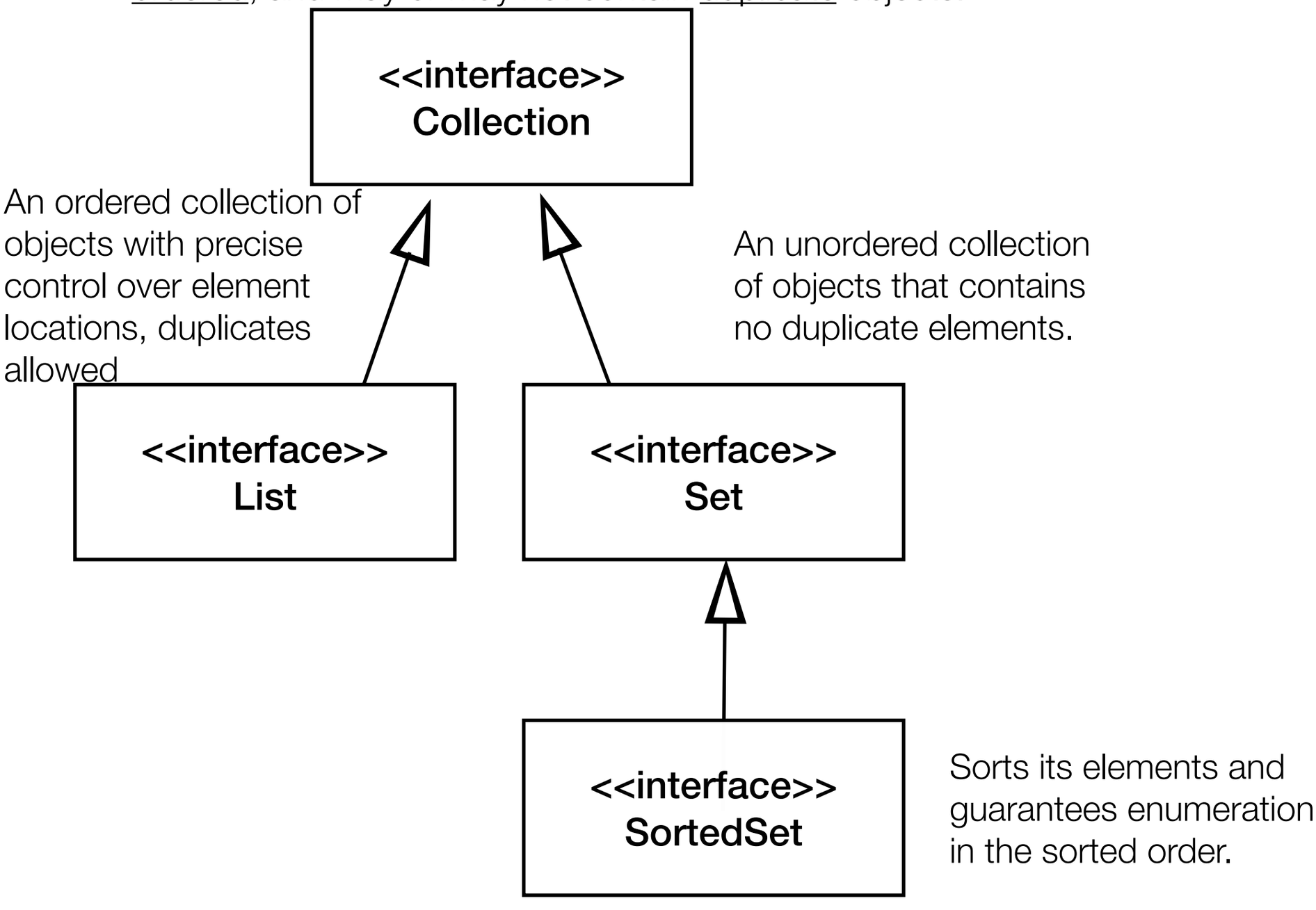
Interfaces in the Java Collections Framework

A group or collection of objects that may or may not be ordered, and may or may not contain duplicate objects.



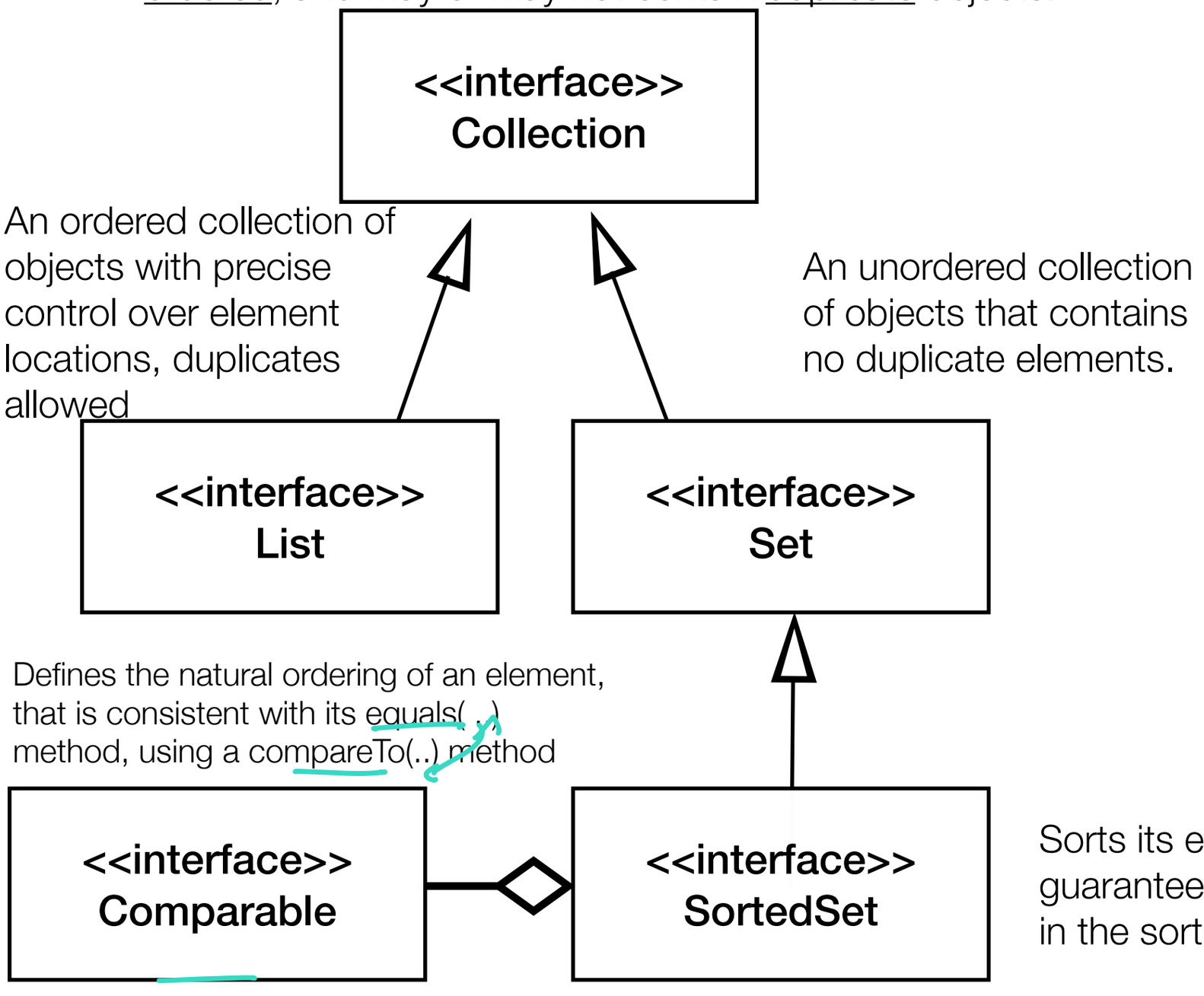
Interfaces in the Java Collections Framework

A group or collection of objects that may or may not be ordered, and may or may not contain duplicate objects.



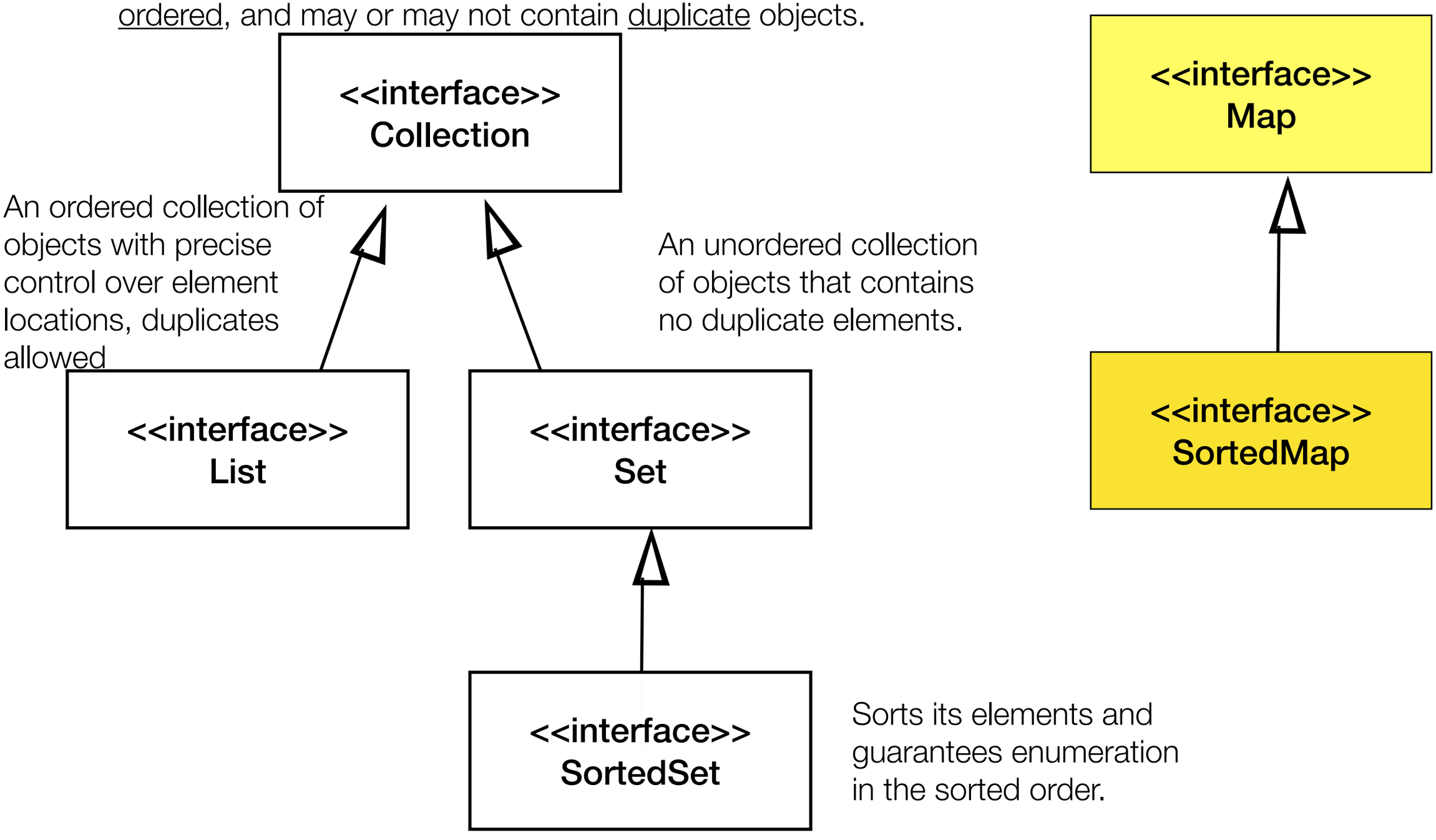
Interfaces in the Java Collections Framework

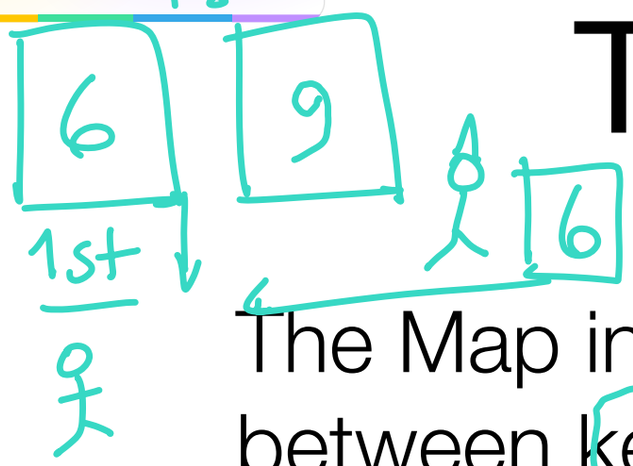
A group or collection of objects that may or may not be ordered, and may or may not contain duplicate objects.



Interfaces in the Java Collections Framework

A group or collection of objects that may or may not be ordered, and may or may not contain duplicate objects.

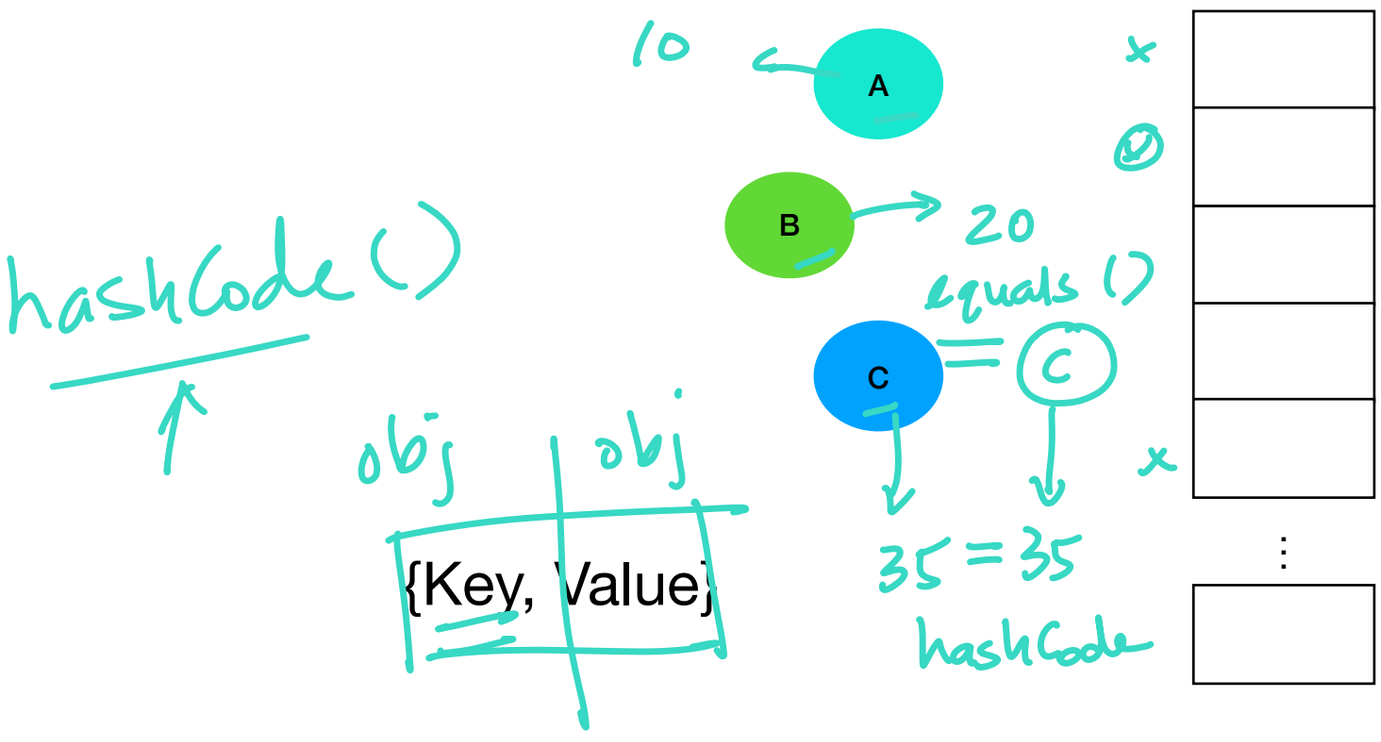




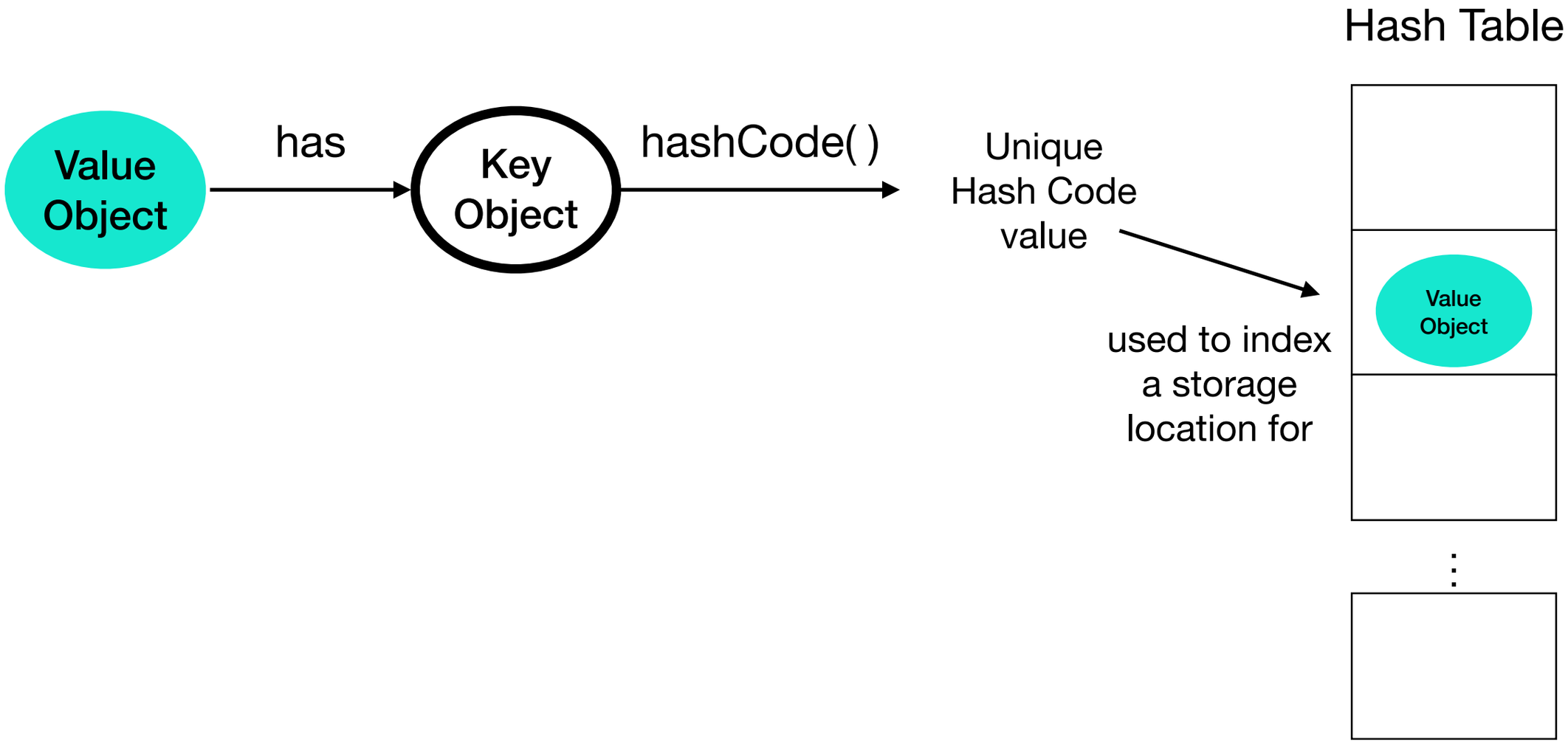
The Map Interface

The Map interface represents a collection of mappings between key objects and value objects. Hash tables are examples of maps.

- The set of key objects in a Map must not have any duplicates. However, the collection of value objects may contain duplicates.



Functionality



The Map Interface

Method	Description
boolean containsKey(Object key)	Returns true if the Map contains a mapping for the specified key, and false otherwise
V get (Object key)	Returns the value object associated with the specified key or null if there is no mapping for the key
Set<E> keySet( )	Returns a Set of all the key objects in the Map
V put(K key, V value)	Creates a key/value mapping in the Map. If the key already exists in the Map, put( ) replaces the value currently in the Map with the value supplied as an argument and returns the value replaced; otherwise it returns the value.
Collection<V> values( )	Returns a Collection of all the value objects in the Map

https://docs.oracle.com/javase/7/docs/api/java/util/Map.html

Example - Map

```
public class Plant{
    private String name;

    public Plant(String n){
        name = n;
    }
    public String getName(){
        return name;
    }
    public String toString(){
        return getName();
    }

}
```

Example - Map
Declaring Map objects

```
1 public class MapDemo{
    2 public static void main(String[] args){
      3. private Map<String,Plant> plants;
                //  {Key, Value}
        //assume the Map (plants) is initialised
    }
}
```

Example - Map
Adding objects

```
public class MapDemo{
    public static void main(String[] args){
      private Map<String,Plant> plants;
       //assume the Map (plants) is initialised

      Plant aloe = new Plant("Aloe"); // value object
      String plantName = aloe.getName(); // key object
      plants.put(plantName, aloe);  //adding {key, value} to Map

      Plant basil = new Plant("Basil");
      plants.put(basil.getName(), basil); //adding {key, value} to Map

    }
  }
```

Example - Map Getting objects

```
public class MapDemo{
    public static void main(String[] args){
1     private Map<String,Plant> plants;
       //assume the Map (plants) is initialised with Aloe and Basil

2.  Plant aloe = plants.get("Aloe"); //getting value from Map
    using the key as input

3.  Plant someBasil = new Plant("Basil");
4   Plant basil =plants.get(someBasil.getName()); //getting value
    from Map using the key as input

  }
}
```

The Map Interface

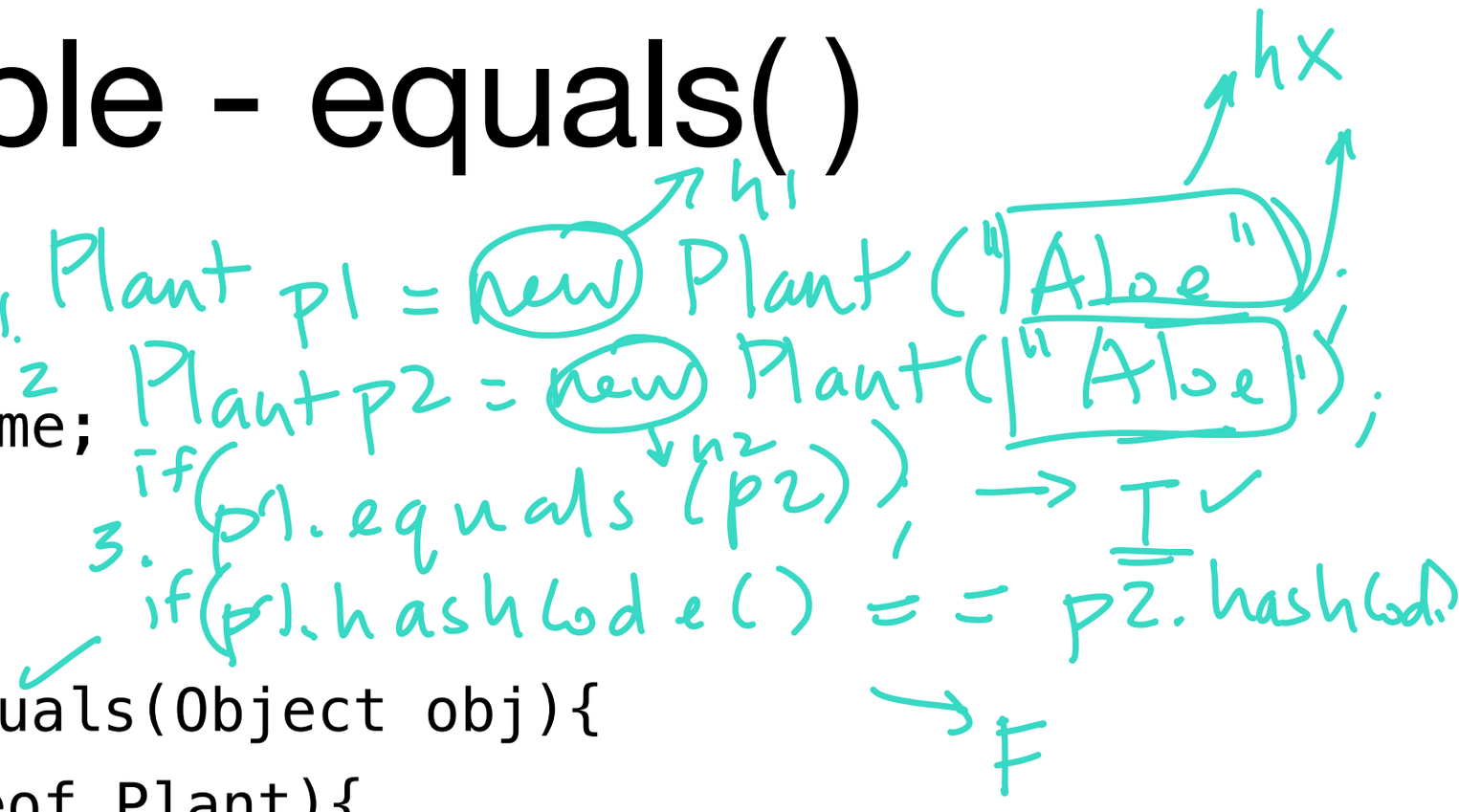
Preventing duplicate keys

To ensure that the set of key objects in the Map does not contain duplicates, it is important that the Key objects override the equals( ) method of the Object class, based on the content of the key.

Duplicates depend on how the equals( ) method is defined.

Example - equals( )

```
public class Plant{
    private String name;
    …

    public boolean equals(Object obj){
       if(obj instanceof Plant){
          Plant p = (Plant) obj;
          if(this.name.equals(p.name))
             return true;
       }
       return false;
    }

}
```



Example - hashCode( )

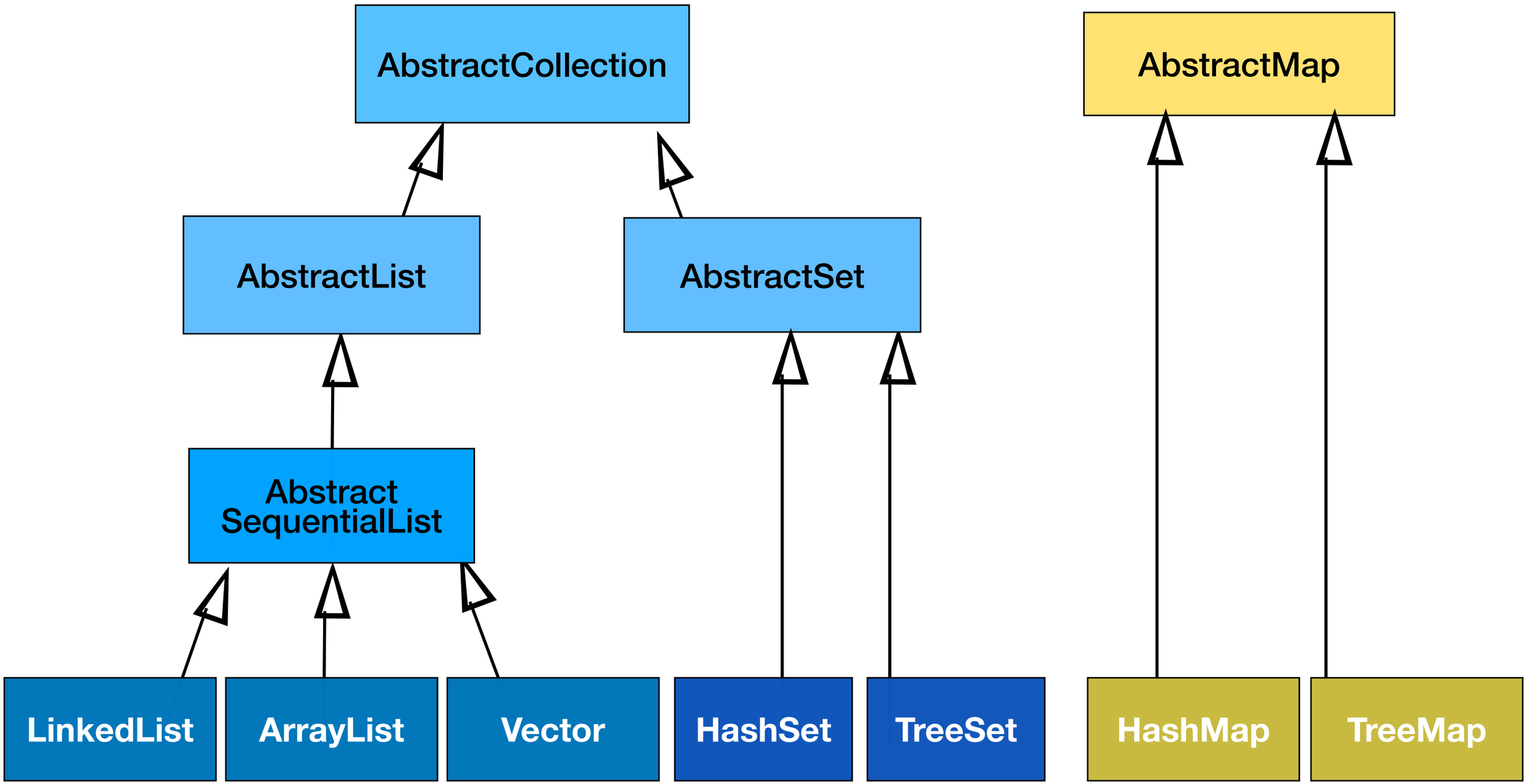
```
public class Plant{
    private String name;
    …

    public int hashCode( ){
       //use the String class hashCode
       return name.hashCode( );
    }

}
```

The Sorted Map Interface

The SortedMap interface represents a Map object that keeps its set of key objects in sorted order. Its keySet( ) and values( ) methods inherited from Map return collection that can be traversed in sorted order of the key.

It also declares methods of its own such as firstKey( ) and lastKey() that return the lowest and highest key values in the SortedMap.

Classes in the Java Collections Framework



Comparisons

Collection	Ordering	Random Access	Key-Value	Duplicate Elements	Null Element	Thread
ArrayList	Yes	Yes	No	Yes	Yes	No
LinkedList	Yes	No	No	Yes	Yes	No
HashSet	No	No	No	No	Yes	No
TreeSet	Yes	No	No	No	No	No
HashMap	No	Yes	Yes	No	Yes	No
TreeMap	Yes	Yes	Yes	No	No	No
Vector	Yes	Yes	No	Yes	Yes	Yes
Hashtable	No	Yes	Yes	No	No	Yes
Properties	No	Yes	Yes	No	No	Yes
Stack	Yes	No	No	Yes	Yes	Yes
CopyOnWriteArrayList	Yes	Yes	No	Yes	Yes	Yes
ConcurrentHashMap	No	Yes	Yes	No	Yes	Yes
CopyOnWriteArraySet	No	No	No	No	Yes	Yes

Source: http://www.journaldev.com/1260/java-collections-framework-tutorial

https://cdn.journaldev.com/wp-content/uploads/2013/01/java-collections-framework.pdf