

Verteilte Software - Java - Collections 1

- Bausteine zur Verwaltung von Objektsammlungen
- enthalten Gruppe von Objekten (eigentlich Referenzen)
- sind dynamisch
- haben unterschiedliche Verarbeitungslogiken (zum Hinzufügen, Entfernen, Zugreifen)
 - List
 - Set
 - Queue
 - Map
- implementieren Interface Collection (meisten)
- seit Java 5 generisch

```
ArrayList<String> cs=new ArrayList<String>();
```

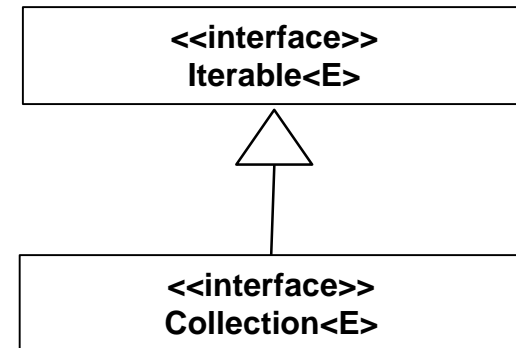
<<interface>> Collection<E>
add (obj:E): boolean remove (obj:Object): boolean clear (): void contains (ob:Object): boolean size (): int iterator(): Iterator<E> ...

Interface Collection

- 15 abstrakte Methoden
- die Implementierungen nutzen die equals-Methode
- iterator() zum Iterieren mit foreach-Schleife, vererbt von Iterable<E>

Interface Iterable<E>

```
public interface Collection<E>  
    extends Iterable<E>
```



- stellt `iterator()` zur Verfügung
- `iterator()` liefert ein `Iterator<E>` über Elemente von Typ `E` (`java.util.Iterator`)
- Interface `Iterator<E>` verfügt über die Methoden zum Iterieren

`hasNext,`
`next,`
`remove`

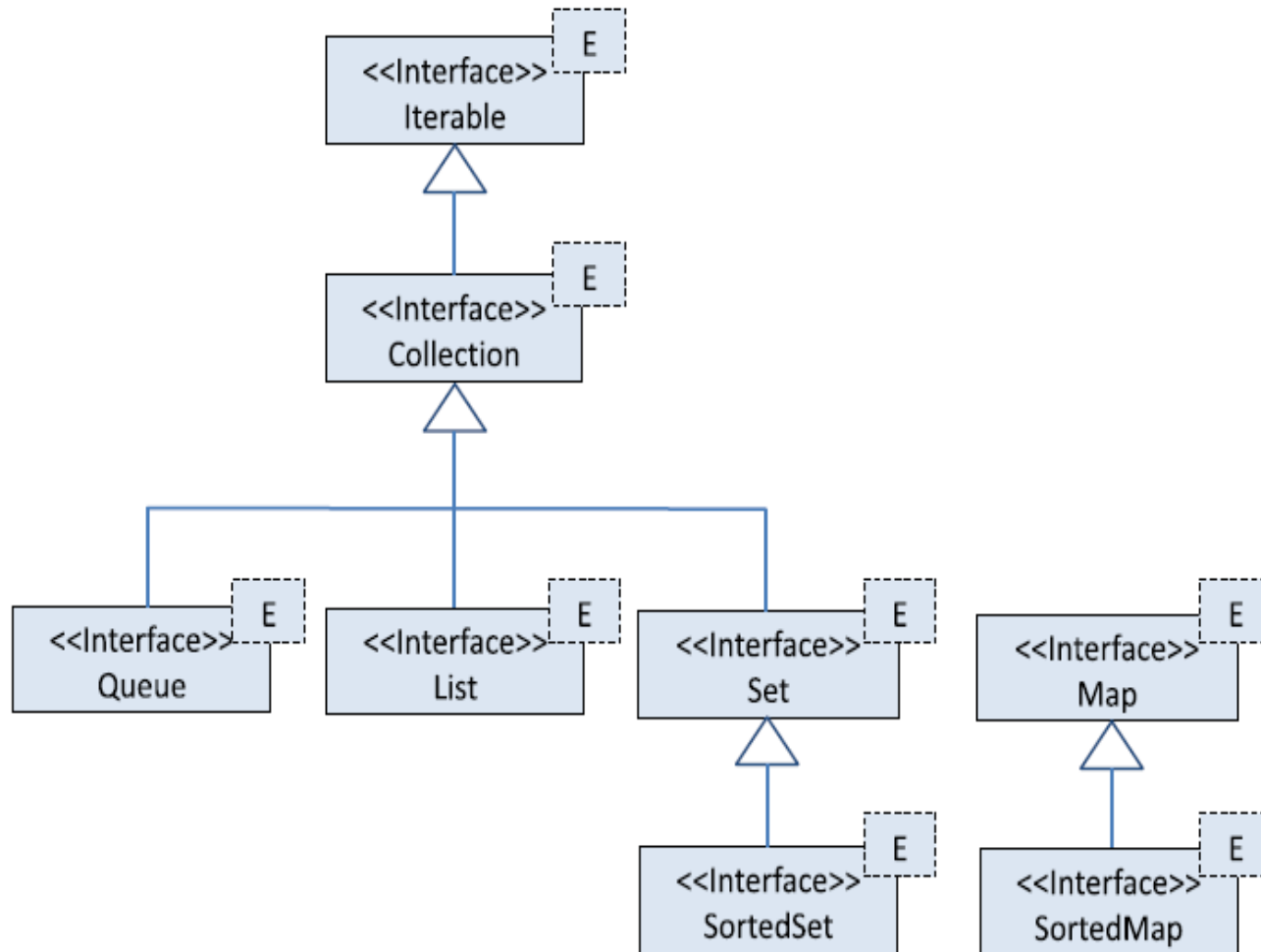
Wichtig für selbst definierte Container-Klassen!

```
ArrayList<String> cs=new ArrayList<String>();  
cs.add("Ilse");  
cs.add("Koch");
```

```
//möglich für Collections und Array  
for (String s:cs) System.out.println(s);
```

```
//möglich auch:  
for (Iterator<String> it=cs.iterator();it.hasNext();){  
    String str=it.next();  
    System.out.println(str);  
}
```

```
//bei ArrayList ausserdem:  
for (int i=0;i<cs.size();i++) {  
    String str=cs.get(i);  
    System.out.println(str);  
}
```



Interfaces und Klassen

Iterable, Iterator

Collection

Set (HashSet, CopyOnWriteArraySet, ...)

Methoden wie Collection

keine Doppelgänger

Queue (ArrayBlockingQueue, PriorityQueue,...)

weitere Methoden, z.B. peek (Abrufen des obersten Elements)

FIFO: first in, first out

außer bei PriorityQueue<E>

List (ArrayList, CopyOnWriteArrayList,...)

weitere Methoden, z. B. add, set , get, indexOf

Position der Elemente gekennzeichnet durch Indizes

Comparable, Comparator

stellen Vergleichsmethoden zur Verfügung:

int compareTo(object) (Comparable<E>)

int compare(object1, object2) (Comparator<E>)

Klasse Collections

static-Methoden zum Bearbeiten von Collections: Sortieren, Suchen, Mischen, etc.

ArrayList - Beispiel

```
public enum Priorität {NORMAL, DRINGEND}

public class Akte {
    public String regisrierNummer;
    public Priorität priorität;
    public Akte(String regisrierNummer, Priorität priorität){
        super();
        this.regisrierNummer = regisrierNummer;
        this.priorität = priorität;
    }
    @Override
    public String toString() {
        return "Akte [" + regisrierNummer + " " + priorität + "]";
    }
}
```

ArrayList - Beispiel

```
import java.util.ArrayList;
public class Ordnen {
    public static void main(String[] args) {
        ArrayList<Akte> haufen=new ArrayList<Akte>();
        haufen.add(new Akte("A01",Priorität.NORMAL));
        haufen.add(new Akte("A02",Priorität.NORMAL));
        haufen.add(new Akte("A03",Priorität.DRINGEND));
        haufen.add(new Akte("A04",Priorität.DRINGEND));

        ArrayList<Akte> geordnet=new ArrayList<Akte>();
        int ind=0;
        for (Akte a:haufen)
            if (a.priorität==Priorität.DRINGEND){
                geordnet.add(ind, a); ind++;
            }
            else geordnet.add(a);

        for (Akte a:geordnet) System.out.println(a);
    }
}
```

Akte	[A03	DRINGEND]
Akte	[A04	DRINGEND]
Akte	[A01	NORMAL]
Akte	[A02	NORMAL]

Interface Map<K,V>

- Assoziativer Speicher
- schnelles Auffinden von Elementen über einen Key
- jedes Element muss einen eindeutigen Key bekommen
- Hashtable, TreeMap,...

Methoden:

```
V put (K key,  V value);  
V get (Object key);  
V remove (Object key);  
boolean containsKey(Object key);  
boolean containsValue(Object value);  
int size();  
boolean isEmpty();  
void clear();  
//...
```

Rückgabewert V: der vorherige Wert,
der dem Schlüssel zugeordnet ist, oder
null, wenn es keine Zuordnung gab


```
Hashtable<String,Double> spenden =  
    new Hashtable<String,Double>();  
spenden.put("Donald", 10.0);  
spenden.put("Mickey", 100.00);  
  
for(String key : spenden.keySet())  
    System.out.println(key+": "+spenden.get(key));  
  
if(spenden.get("Dagobert") == null){  
    System.out.println("Dagobert hat nichts gespendet");  
}
```
