



# Übersicht Java-API-Methoden

## LinkedList

LinkedList.add(ELEMENT) OPTIONAL: LinkedList.add(INDEX,ELEMENT) ⇒ Fügt Wert zur Liste hinzu

LinkedList.addFirst(ELEMENT) ⇒ Element vorne in die Liste einfügen

LinkedList.addLast(ELEMENT) ⇒ Element hinten in die Liste einfügen

LinkedList.get(INDEX) ⇒ Gibt Element an Position X aus

LinkedList.size() ⇒ Größe der LinkedList

LinkedList.getFirst() ⇒ gibt erstes Element aus

LinkedList.isEmpty() ⇒ prüft ob Liste leer ist

LinkedList.clone() ⇒ Liste kopieren (eine == Zuweisung wäre nur eine Referenzzuweisung und keine exakte Kopie)

LinkedList.contains(ELEMENT) ⇒ Check, ob Element in Liste

## Stack

Stack.empty() ⇒ Prüft ob Stack leer ist

Stack.pop() ⇒ Entfernt das Objekt oben vom Stapel

Stack.push() ⇒ Fügt Objekt oben auf dem Stapel ein

## HashSet (Alle Werte unique ⇒ z.B. Warteschlange ohne doppelte Werte)

HashSet.add(Value) ⇒ Wert hinzufügen

HashSet.contains(value) ⇒ checken ob Wert in HashSet

HashSet.remove(value) ⇒ Wert entfernen

HashSet.clear() ⇒ Alle Werte entfernen

HashSet.size() ⇒ Größe des HashSets

## HashSet iterieren:

```
for (String i : HashSet) {  
    System.out.println(i);  
}
```

# HashMap

HashMap.put(KEY,VALUE) ⇒ Wert hinzufügen

HashMap.get(KEY) ⇒ Wert, der Key zugeordnet ist auslesen

HashMap.remove(KEY) ⇒ Wert der dem Key zugeordnet ist entfernen

HashMap.size() ⇒ Größe der HashMap

HashMap.clear() ⇒ Alle Element entfernen

## HashMap iterieren:

HashMap.keySet() ⇒ gibt alle Keys aus

HashMap.values() ⇒ gibt alle Werte aus

```
for (String i : HashMap.keySet()/values()) {  
    System.out.println(i);  
}
```

# Iterator (wird Collections verwendet, z.B. HashSet)

Erstellen: `Iterator<DATENTYP> it = HashSet.iterator();`

`it.hasNext()` ⇒ Checkt, ob es noch ein weiteres Element gibt

`it.next()` ⇒ gibt das nächste Element aus