

## List-Methoden (z. B. ArrayList)

E ... Element

boolean **add**(E e)

*Appends the specified element to the end of this list.*

void **add**(int index, E element)

*Inserts the specified element at the specified position in this list.*

void **clear**()

*Removes all of the elements from this list.*

E **get**(int index)

*Returns the element at the specified position in this list.*

boolean **isEmpty**()

*Returns true if this list contains no elements.*

E **remove**(int index)

*Removes the element at the specified position in this list.*

E **set**(int index, E element)

*Replaces the element at the specified position in this list with the specified element.*

int **size**()

*Returns the number of elements in this list.*

Iterator<E> **iterator**()

*Returns an iterator over the elements in this list in proper sequence.*

boolean **contains**(Object o)

*Returns true if this list contains the specified element. \**

int **indexOf**(Object o)

*Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element. \**

int **lastIndexOf**(Object o)

*Returns the index of the last occurrence of the specified element in this list, or -1 if this list does not contain the element. \**

boolean **remove**(Object o)

*Removes the first occurrence of the specified element from this list, if it is present. \**

---

\* nur mit String-Parameter

### **Set-Methoden (z. B. HashSet)**

E ... Element (nur String)

boolean **add**(E e)

*Adds the specified element to this set if it is not already present.*

**void clear()**

*Removes all of the elements from this set.*

**boolean isEmpty()**

*Returns true if this set contains no elements.*

**Iterator<E> iterator()**

*Returns an iterator over the elements in this set.*

**int size()**

*Returns the number of elements in this set (its cardinality).*

**boolean contains**(Object o)

*Returns true if this set contains the specified element. \**

**boolean remove**(Object o)

*Removes the specified element from this set if it is present. \**

### **Map-Methoden (z. B. HashMap)**

K ... Key (nur String)

V ... Value

**void clear()**

*Removes all of the mappings from this map.*

**boolean isEmpty()**

*Returns true if this map contains no key-value mappings.*

**Set<K> keySet()**

*Returns a Set view of the keys contained in this map.*

**V put**(K key, V value)

*Associates the specified value with the specified key in this map.*

**int size()**

*Returns the number of key-value mappings in this map.*

**boolean containsKey**(Object key)

*Returns true if this map contains a mapping for the specified key. \**

**boolean containsValue**(Object value)

*Returns true if this map maps one or more keys to the specified value. \**

**V get**(Object key)

*Returns the value to which the specified key is mapped, or null if this map contains no mapping for the key. \**

**V remove**(Object key)

*Removes the mapping for a key from this map if it is present (optional operation). \**