

Übersicht Collections

	Array (ist keine Collection)	List	Set	Map
Verwendung	(1) Wenn die Anzahl der zu speichernden Elemente bekannt ist und sich nicht ändert. (2) Wenn Geschwindigkeit wichtig ist.	(1) Wenn die Anzahl der zu speichernden Elemente unbekannt ist und sich ändern kann.	(1) Wie List. (2) Wenn kein zu speicherndes Element mehr als einmal vorkommen soll. Die Gleichheit wird mit hashCode() und equals() oder compareTo() bestimmt.	(1) Wie List. (2) Wenn Elemente individuelle Indizes/Keys haben sollen. (3) Keys wie Set.
Deklaration	<code>int[] a;</code>	<code>List<Integer> l;</code>	<code>Set<Integer> s;</code>	<code>Map<String, Integer> m;</code>
Instanziierung	(1) <i>ohne Wertzuweisung:</i> <code>a = new int[2];</code> (2) <i>mit Wertzuweisung:</i> <code>a = new int[]{47,11}</code>	(1) <i>viele Indexzugriffe, bekannte Größe:</i> <code>l = new ArrayList<>();</code> (2) <i>wenig Indexzugriffe, unbekannte Größe:</i> <code>l = new LinkedList<>();</code>	(1) <i>Gleichheit mit hashCode() + equals():</i> <code>s = new HashSet<>();</code> (2) <i>sortiert, Gleichheit mit compareTo():</i> <code>s = new TreeSet<>();</code>	(1) <i>unsortiert, schneller:</i> <code>m = new HashMap<>();</code> (2) <i>sortiert, langsamer:</i> <code>m = new TreeMap<>();</code>
Einfügen	(1) <i>Einfügen an Stelle 0 ≤ i < a.length:</i> <code>a[i] = 11;</code>	(1) <i>Einfügen ans Ende der Liste:</i> <code>l.add(11);</code> (2) <i>Einfügen an Stelle 0 ≤ i < l.size():</i> <code>l.add(i, 11);</code>	(1) <i>Einfügen in das Set:</i> <code>s.add(11);</code>	(1) <i>Einfügen in die Map:</i> <code>m.put("11", 11);</code>
Auslesen	<code>int v = a[1];</code>	<code>int v = l.get(1);</code>	Kein Indexzugriff möglich.	<code>int v = m.get("11");</code>
Durchlaufen (Einfügen)	<code>for (int i = 0; i < a.length; ++i)</code> { <code>a[i] = i;</code> }	<code>for (int i = 0; i < 47; ++i)</code> { <code>l.add(i);</code> }	<code>for (int i = 0; i < 47; ++i)</code> { <code>s.add(i);</code> }	<code>for (int i = 0; i < 47; ++i)</code> { <code>m.put("" + i, i);</code> }
Durchlaufen (Auslesen)	<code>for (int i = 0; i < a.length; ++i)</code> { <code>System.out.println(a[i]);</code> }	(1) <i>ArrayL-/LinkedList:</i> <code>for (Integer i : l)</code> { <code>System.out.println(i);</code> } (2) <i>ArrayList (ist bei LinkedList langsam):</i> <code>for (int i = 0; i < l.size(); ++i)</code> { <code>System.out.println(l.get(i));</code> }	<code>for (Integer i : s)</code> { <code>System.out.println(i);</code> }	(1) <i>Elemente unsortiert auslesen:</i> <code>for (Integer i : m.values())</code> { <code>System.out.println(i);</code> } (2) <i>Indizes/Key un-/sortiert auslesen:</i> <code>for (String s : m.keySet())</code> { <code>System.out.println(m.get(s));</code> } (2) <i>Indizes/Key und Elemente auslesen:</i> <code>for (Entry<String, Integer> e : m.entrySet())</code> { <code>System.out.println(e.getValue());</code> }
Besonderheiten	(1) <i>Sortieren:</i> <code>Arrays.sort(a);</code> (2) <i>Suchen:</i> <code>int Arrays.binarySearch(a, 11);</code> (3) <i>In List umwandeln:</i> <code>List Arrays.asList(a);</code> (4) <i>Unit-Tests:</i> <code>assertArrayEquals(exp, act);</code>	(1) <i>Sortieren:</i> <code>Collections.sort(l);</code> (2) <i>Suchen:</i> <code>int Collections.binarySearch(l, 11);</code> (3) <i>In Array umwandeln:</i> <code>int[] l.toArray(new int[l.size()]);</code> (4) <i>Weitere Methoden:</i> <code>contains, indexOf, isEmpty, remove, set, clear</code>	(1) <i>Sortieren:</i> <code>new TreeSet<>() // compareTo()</code> <code>new TreeSet<>(Comparator c) // compareTo()</code> (2) <i>In List umwandeln:</i> <code>List<Integer> l;</code> <code>l = new ArrayList<>(s);</code> (3) <i>Weitere Methoden:</i> <code>size, contains, isEmpty, remove, clear, first, last</code>	(1) <i>Sortieren:</i> <code>new TreeMap<>() // compareTo()</code> <code>new TreeMap<>(Comparator c) // compareTo()</code> (2) <i>In List umwandeln:</i> <code>List<Integer> l;</code> <code>l = new ArrayList<>(m.values());</code> (3) <i>Weitere Methoden:</i> <code>size, containsKey, containsValue, isEmpty, remove, clear</code>