

Verwendung von Listen in Java

Die Liste ist eine dynamische Datenstruktur zur Speicherung einer beliebigen Menge von Objekten gleichen Objekttyps.

Voraussetzung: Import des ArrayList-Paktes **vor** der Klassendefinition durch

```
import java.util.ArrayList;
```

Erzeugung: Listen werden über den Konstruktor `ArrayList<Typ>()` erzeugt.

Beispiel: `ArrayList<Person> personenliste = new ArrayList<Person>();`

Methoden (Auszug):

Methodenname	Beschreibung
<code>add(objekt)</code>	Fügt das Objekt zur Liste hinzu. Die Objekte werden intern mit einem Index nummeriert. Die Nummerierung beginnt mit 0.
<code>clear()</code>	Löscht den Inhalt der gesamten Liste.
<code>clone()</code>	Gibt eine Kopie der Liste als neues Objekt zurück.
<code>contains(objekt)</code>	Gibt <code>true</code> zurück, falls das Element in der Liste enthalten ist.
<code>get(index)</code>	Liefert das Objekt mit dem Index zurück.
<code>indexOf(objekt)</code>	Gibt den Index des ersten Auftretens des Elementes in der Liste zurück.
<code>isEmpty()</code>	Gibt <code>true</code> zurück, falls die Liste leer ist.
<code>remove(index)</code>	Entfernt das Listenelement mit dem angegebenen Index.
<code>remove(objekt)</code>	Entfernt das Listenelement Objekt
<code>set(index, objekt)</code>	Ersetzt das Element an der Indexstelle durch das Objekt.
<code>size()</code>	Gibt die Anzahl der Listenelemente zurück.
<code>subList(anfang, ende)</code>	Gibt eine Liste bestehend aus den Elementen ab Anfang bis Ende-1 zurück.

Listen über primitive Datentypen können nur mit Hilfe der Wrapper-Klassen für primitive Typen erzeugt werden.

Beispiel für eine Integer-Zahlenliste:

```
ArrayList<Integer> zahlenliste = new ArrayList<Integer>();
```

Durchlauf:

Um Listenelemente einzeln zu untersuchen, durchläuft man die Liste. Dafür gibt es mehrere Varianten. Die Verwendung einer speziellen Zählschleife vereinfacht das Durchlaufen (siehe auch Algorithmen-Übersicht).