

ArrayList ist eine dynamische Datenstruktur - eine Klasse zum Speichern einer **offenen Anzahl** von Elementen (item) als Liste.

Eigenschaften der ArrayList:

- Die Anzahl der Elemente der Liste ändert sich nach Bedarf. **Die Liste kann wachsen oder schrumpfen.**
- Sie speichert die Größe der Liste.
- Die Elemente der Liste werden in der gleichen Reihenfolge gespeichert, wie sie eingegeben wurden. --> **ein Index wird automatisch vergeben.**
- Der Index beginnt bei 0.

Definition und Konstruktor eines Objektes ArrayList

```
import java.util.ArrayList           vor dem Programmkopf das Objekt-Package einbinden
ArrayList<ObjektTyp> liste          Festlegen des Datenfeldes und des Typs der ArrayList-Objekte
liste = new ArrayList<ObjektTyp> ( );  Angabe im Konstruktor
```

Das `java.util.ArrayList` bietet auch die **Werkzeuge zum Verwalten einer ArrayList**:

<code>liste.add (item);</code>	Element an das Ende der Liste anfügen, Ergebnistyp: boolean
<code>liste.add (index, item);</code>	Objekt in der angegebenen Position einfügen, rechts stehende Elemente werden nach rechts verschoben.
<code>liste.set (index, item);</code>	Das Element an Position <code>index</code> wird durch neues Objekt ersetzt
<code>System.out.println (liste);</code>	Ausgabe der gesamten Liste
<code>liste.remove (index);</code>	Löschen eines Elements
<code>liste.contains (item)</code>	Prüft, ob ein Element enthalten ist
<code>liste.get (index)</code>	Ein Element mit einem bestimmten <code>index</code> ausgeben.
<code>liste.size ()</code>	Gibt die Anzahl der Elemente der Liste an
<code>liste.indexOf (item)</code>	Liefert den Index eines Elements zurück
<code>liste.clear ()</code>	Die gesamte Liste wird gelöscht.
<code>liste.isEmpty ()</code>	Prüft, ob die Liste leer ist

Fehlermeldung: Wird versucht mit einem Index, der außerhalb der zulässigen Grenzen liegt, auf die Datensammlung zuzugreifen, wird das laufende Programm mit einer **IndexOutOfBoundsException** - Fehlermeldung abgebrochen.

Durchsuchen einer ArrayList

Oft ist es notwendig, die gesamte Liste zu durchsuchen. Dazu kann man mit der bereits bekannten while - Schleife arbeiten.

1. while-Schleife

```
while (index < liste.size()) {
Anweisungen
index++;
}
```

Beispiel:

```
public void AusgabeElemente(String element) {
int index = 0;
while (index < umzugsListe.size()) {
System.out.println (umzugsListe.get(index));
index++; } }
```

Es gibt allerdings in der ArrayList- Klasse von Java besondere Strukturen zum Durchsuchen einer ArrayList- gleichwertig ist auch die

2. for - each- Schleife

```
for (ObjektTyp item: liste) {
Anweisung
}
```

Beispiel:

```
public void suchenNach(String element) {
for (String element: umzugsListe){
System.out.println(element); }
}
```