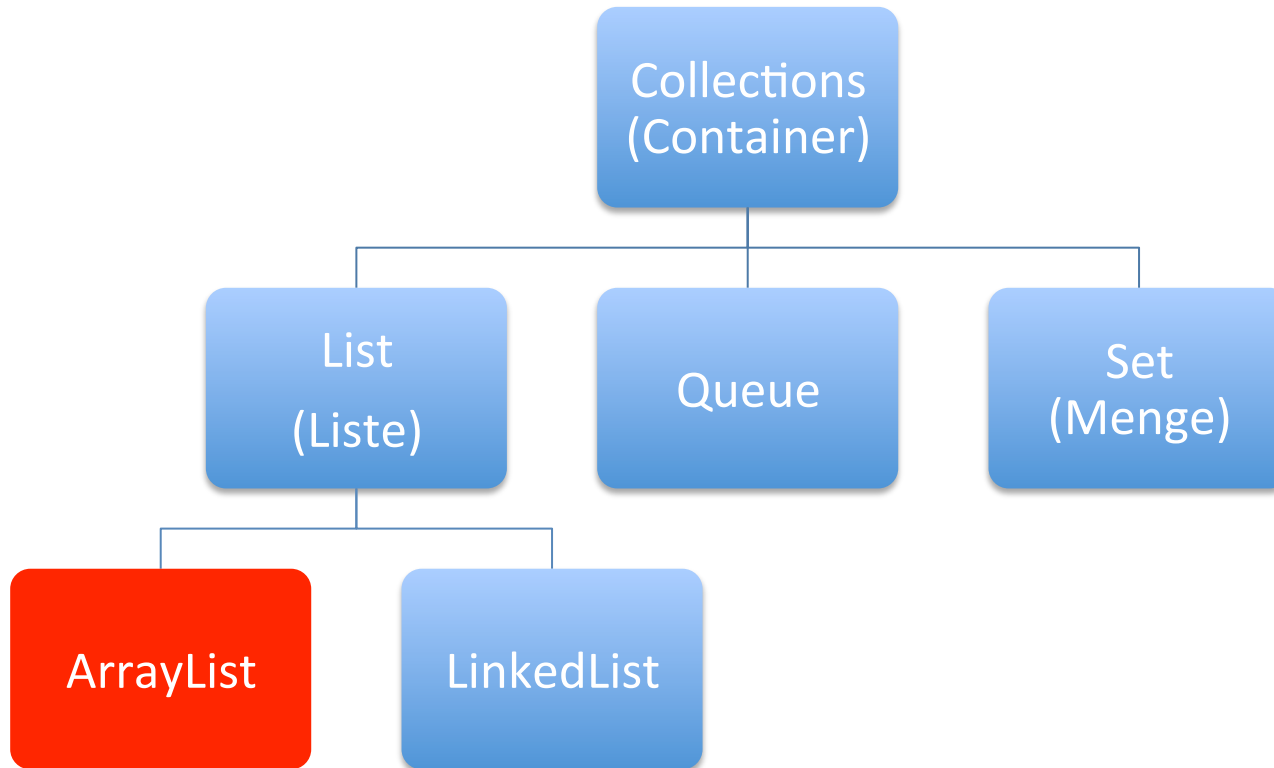


Java:

ArrayList

Hierarchie



ArrayList vs. LinkedList:

- ArrayList ist schneller bei Zugriff auf einzelne Elemente
- LinkedList ist schneller bei löschen/hinzufügen

Unterschied Array <-> ArrayList

wichtiger Unterschied:

ArrayList ist dynamisch = Größe veränderbar

Unterschied Array <-> ArrayList

ArrayList = Bibliotheksklasse aus dem Paket java.util (Array: Datentyp, der durch interne Java-Klasse repräsentiert wird)

→ beachten Sie den Import in Zeile 7

ArrayList: Benutzt Bibliothek

```
7 import java.util.ArrayList;
8
9 public class Main
10 {
11     public static void main(String[] args) {
12         ArrayList<Integer> z = new ArrayList<Integer>();
13     }
```

Array: Benötigt keinen Import

```
7
8 public class Main
9 {
0     public static void main(String[] args) {
1         int[] zahlArray = new int[13];
2     }
```

ArrayList: Syntax

```
ArrayList<String> eineListe = new ArrayList<String>();
```

In Spitzklammern: Datentyp

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

ArrayList kann keine primitiven Datentypen (int ...) verwalten, sondern nur Objekte
Deshalb Verwendung von Wrapperklassen.

ArrayList: Wichtige Methoden 1

```
ArrayList<String> liste = new ArrayList<String>();
```

Element ausgeben	<code>liste.get(13)</code>
Element einfügen (ohne Positionsangabe)	<code>liste.add("Heinrich")</code>
Element einfügen (mit Positionsangabe)	<code>liste.add(0, "Heinrich")</code>
Element ändern	<code>liste.set(0, "Heini")</code>
Element entfernen	<code>liste.remove("Heinrich")</code> <code>liste.remove(1) // Index</code>
Größe zurückgeben	<code>liste.size();</code>

Aufgabe: ArrayList verwenden - Lieblingsfächer

1. Erstellen Sie eine ArrayList, die Ihre zwei Lieblingsfächer enthält.
2. Oh, Entschuldigung: Es sollten drei Lieblingsfächer sein. Fügen Sie noch ein drittes ein.
3. Durchlaufen Sie die Liste mit einer for-Schleife (das geht eigentlich anders, lernen wir aber noch) und geben Sie die Lieblingsfächer aus im Format "Meine Lieblingsfächer sind: Deutsch Englisch Erdkunde"
4. Ändern Sie das erste Fach in "Leibesübung".

ArrayList: Wichtige Methoden 2

```
ArrayList<String> liste = new ArrayList<String>();
```

Prüfen, ob Objekt enthalten ist	<code>liste.contains("Heinri")</code>
Welchen Index hat ein Objekt?	<code>liste.indexOf("Heinri")</code>
Liste löschen	<code>liste.clear();</code>

Aufgabe: ArrayList verwenden - Lottozahlen

1. Die Lottozahlen vom Samstag waren 1,17,25,30,31,33. Speichern Sie sie in einer ArrayList "lottozahlen".
2. Prüfen Sie, ob sich die Zahl 12 in der Liste befindet. Prüfen Sie, ob sich die Zahl 31 in der Liste befindet.
3. Lassen Sie sich die Länge der Liste ausgeben (es müsste 6 herauskommen).
4. Angenommen, Sie würden diesen Befehl eingeben:

```
lottozahlen.remove(1);
```

Was glauben Sie: Welche Zahl würde entfernt werden?

→ Probieren Sie es aus, NACHDEM Sie Ihre Vermutung angestellt haben.

ArrayList: Listen ausgeben

a) toString()

```
ArrayList<Integer> liste = new ArrayList<Integer>();  
liste.add(12);  
liste.add(1000);
```

```
System.out.println(liste.toString());
```

ArrayList: Listen ausgeben

b) foreach-Schleife

```
ArrayList<Integer> liste = new ArrayList<Integer>();  
liste.add(12);  
liste.add(1000);
```

```
for(int ausgabe : liste)  
{  
    System.out.println(ausgabe);  
}
```

ArrayList: Listen ausgeben

c) ListIterator

```
ArrayList<Integer> liste = new ArrayList<Integer>();  
liste.add(12);  
liste.add(1000);
```

```
ListIterator<Integer> li =  
    liste.listIterator();
```

```
while (li.hasNext()) {  
    System.out.println(li.next());  
}
```

next() springt zum nächsten Element in der ArrayList!

Aufgabe: Listen sequenziell ausgeben

1. Erstellen Sie mit einer for-Schleife eine ArrayList, die die Zahlen 10, 20, 30 ... bis 1000 enthält.
2. Lassen Sie sich alle Elemente ausgeben,
 1. einmal mit einer foreach-Schleife,
 2. einmal mit dem ListIterator