

Listen

1. Nachteile von Arrays

Um viele Dinge vom gleichen Datentyp abzuspeichern, haben wir bisher Arrays benutzt. Arrays haben aber einige Nachteile:

- Arrays sind beschränkt, d. h. nachdem ein Array einmal mit einer festen Länge initialisiert wurde kann diese nachträglich nicht mehr geändert werden um beispielsweise weitere Elemente hinzufügen zu können.
- Als Index sind nur fortlaufende Zahlen möglich.
- Bei einer Suche müssen entweder die Einträge davor sortiert werden oder alle Einträge durchsucht werden.
- Ein Element zwischendrin einfügen oder löschen und damit alle nachfolgenden Elemente zu verschieben funktioniert nur manuell, ebenso das Einfügen oder Löschen am Anfang des Arrays.
- Sortiert einfügen ist somit sehr umständlich.

2. List

Um vor allem das erste Problem zu lösen, nimmt man in Java sogenannte Listen. Die einfachste hierbei ist die `ArrayList`:

```
1 ArrayList<Integer> liste = new ArrayList<Integer>();
2
3 liste.add(5);
4 liste.add(8);
5
6 for(int i=0 ; i<liste.size() ; i++) {
7     System.out.println( liste.get(i) );
8 }
```

Listing 1: ArrayList

Eine Besonderheit dabei ist, dass diese Listen prinzipiell mit allen Datentypen funktionieren. Man muss den Datentyp, den man allerdings verwenden will explizit in den geschweiften Klammern angeben!¹ Hierbei ist zu beachten, dass in den Spizen Klammern keine *primitiven Datentypen* (wie `int`, `float`, ...) stehen dürfen, sondern die entsprechenden „vollwertigen“ Klassen (`Integer`, `Float`, ...)

Zur Liste kann man dann mit der Methode `add()` ein weiteres Element hinzufügen. Die Methode `size()` gibt die Länge der Liste an, mit `get(i)` kann man das Element am Index `i` abrufen (äquivalent beim Array: `[i]`)

¹Anmerkung: das nennt sich *Generics*

3. Aufgabe: Schule

Erweitere deine Schule von letztem Mal um eine Klasse **Schule**. Diese soll eine Liste beinhalten mit allen Personen. Dazu benötigt man auch eine Methode **addPerson**, um neue Personen hinzuzufügen. Außerdem eine Methode **Ausgabe()**, die alle Personen auf der Konsole ausgibt.

- a) Zeichne zuerst das Klassendiagramm (inklusive der Klassen aus letzter Stunde!)
- b) Erstelle in der **Main**-Klasse² in der **main**-Methode eine Instanz der Schule und füge anschließend neue Personen über die Methode der Schule hinzu.
- c) Rufe anschließend die **Ausgabe**-Methode der Schule auf.

```
1 Schule dhg = new Schule("Droste-Hülshoff-Gymnasium");
2
3 Schüler as = new Schüler("Aaron","Sommer","KS1");
4 dhg.addPerson( as );
5
6 Person mc = new Schüler("Milena","Cordes","KS1");
7 dhg.addPerson( mc );
8
9 dhg.addPerson( new Lehrer("Alexander","Kimmig","10b,KS1,KS2") );
10 dhg.addPerson( new Hausmeister("Bruno","del_Core") );
11 dhg.addPerson( new Schulleiter("Stefan","Maier","KS2") );
12
13 dhg.Ausgabe();
```

Listing 2: Beispielcode

- d) Im Beispiel siehst du 3 Möglichkeiten, Personen hinzuzufügen. Nenne deren Unterschiede und beschreibe, welche Vor- und Nachteile diese haben.

²oder **Test**-Klasse, je nachdem wie du die genannt hast