Listen in Java

Array

- ein Array ist eine Liste von Werten/Objekten mit einer festen Anzahl.
- alle Werte und Objekte liegen im Speicher beieinander -> kann schnell durchlaufen werden.
- es ist nur ein Datentyp erlaubt.

Codebeispiel:

```
...
//erstellen des Arrays
int[] lotto = new int[6];

//Ziehen der Lottozahlen
...
//Speichern von Werten in einem Array
lotto[0]=5;
lotto[1]=15;v
lotto[2]=25;
lotto[3]=26;
lotto[4]=30;
lotto[5]=31;

//Werte ändern / löschen des 3.Wertes
lotto[2]=0;

//ausgabe
for(int i = 0; i<lotto.length; i++){
    System.out.println(lotto[i]);
}
```

Pseudocodebeispiel ForEach-Schleife

```
for (int i : primzahlen) {
  if(i==5){
      System.out.println("5 ist in primzahlen enthalten")
  }
}
```

```
FÜR alle i in primzahlen
  WENN i=5
    AUSGABE "5 ist in primzahlen enthalten"
  ENDE WENN
ENDE FÜR
```

ArrayList

- eine ArrayList ist eine dynamische Liste von Objekten.
- Intern wird die ArrayList mit einem Array abgebildet, welches immer genug freie Plätze vorhält. D.h. Es wird intern eine neue Array-Struktur aufgebaut, wenn die bestehende Liste bis zu einem bestimmten Schwellenwert gefüllt ist. -> eine ArrayList ist ähnlich schnell im zugriff wie ein Array.
- eine ArrayList ist keine Queue und kein Stack.

Codebeispiel:

```
 // ArrayListe erstellen
ArrayList<Integer> primzahlen = new ArrayList<>();

// Objekte hinzufügen
primzahlen.add(2);
primzahlen.add(3);
primzahlen.add(5);

// Liste ausgeben
System.out.println(primzahlen);

// Objekt entfernen
primzahlen.remove(2);

// Über Liste iterieren
for (Integer i : primzahlen) {
           System.out.println(i);
 }

 //prüfen ob Werte in Liste sind
 System.out.println(primzahlen.contains(3));
```

Linked List

Die LinkedList ist eine dopplet verkettete Liste. D.h. jedes Element kennt seinen Vorgänger und Nachfolger. Über der Namen der LinkedList greift man mit einem Zeiger auf das Erste Element der Liste zu. Das letzte Element der Liste enthält den Wert `null`.

Vorteile

- es können beliebig viele ELemente hinzugefügt und wieder entfernt werden.
- eine LinkedList kann auch als Stapel verwendet werden (Sie ist es nicht)

Nachteile

- langsame Iteration (z.B. beim sortieren)

Codebeispiel:

```
// LinkedList erzeugen
LinkedList<String> einkaufliste = new LinkedList<>();

// Werte hinzufügen
einkaufliste.add("Milch");
einkaufliste.add("Bier");
einkaufliste.add("Butter");
einkaufliste.add("Brot");
einkaufliste.add("Wein");
einkaufliste.add("Chips");

// Ausgabe
System.out.println("Ausgabe mit Syso");
System.out.println(einkaufliste);
System.out.println("---------------------");

// entfernen von Objekten
System.out.println("Entferne Element mit index 2 ");
einkaufliste.remove(2);
System.out.println(einkaufliste);
System.out.println("---------------------");

// Iterieren über Liste
System.out.println("Iteriere ueber ganze Liste");
for (String s : einkaufliste) {
    System.out.println(s);
}
System.out.println("---------------------");

// Liste als Queue
System.out.println("Stapeloperationen ");
einkaufliste.removeFirst();
einkaufliste.removeLast();
einkaufliste.addFirst("Wasser");
einkaufliste.addLast("Salami");

// Stapelfunktionen
System.out.println(einkaufliste);
System.out.println(einkaufliste.pop());
System.out.println(einkaufliste);
System.out.println(einkaufliste.pollLast());
System.out.println(einkaufliste);
```

```
Ausgabe mit Syso
[Milch, Bier, Butter, Brot, Wein, Chips]
---------------------
Entferne Element mit index 2
[Milch, Bier, Brot, Wein, Chips]
---------------------
Iteriere ueber ganze Liste
Milch
Bier
Brot
Wein
Chips
---------------------
Stapeloperationen
[Wasser, Bier, Brot, Wein, Salami]
Wasser
[Bier, Brot, Wein, Salami]
Salami
[Bier, Brot, Wein]
```