

# Vermischte Aufgaben Arrays

Das BlueJ-Projekt für Aufgabe 1 und 2 findest du hier.



## (A1)

Es soll ein Sparbuch modelliert werden, das eine Abfolge von Kontoständen speichern kann. Dazu wird ein Array kontostand verwendet.

- Das Array enthält die folgenden Werte: 998.12, 1122.54, 786.67, 1329.97.
  - Wie viele Elemente enthält das Array?
  - Welchen Index hat das erste Element?
  - Welchen Index hat das letzte Element?
- Deklare und initialisiere ein Array kontostand der Länge 4 in der Klasse Konto, das die angegebenen Werte enthält:
  - Mit `new ... + Initialisierung` im Konstruktor
  - Durch direkte Initialisierung mit Hilfe von `{...}` bei der Deklaration.



## (A2)

Verwende die Klasse Lotto um ein einfaches Lottospiel zu implementieren.

- Lege ein Attribut `gewinnzahlen` an und initialisiere das Attribut im Konstruktor mit zufälligen Zahlen zwischen 1 und 49. Jede Zahl darf dabei nur einmal vorkommen. Du kannst in den Gewichtsbeispielen aus den vorigen Aufgaben nachschauen, wie man Zufallszahlen erzeugen kann.
- Implementiere eine Methode `spielen`, die vom Spieler 6 Zahlen erfragt und diese im Attribut `getippt` der Klasse speichert. Die Methode `auswerten` soll anschließend zurückgeben, wieviele korrekte Zahlen getippt wurden. Die Ausgabe sollte z.B. so aussehen. "Sie haben 4 Richtige Zahlen getippt: 7, 12, 32 und 44".



## (A3)

1)

Löse die folgenden Aufgaben mit Stift und Papier.

Hier gibt es eine PDF-Version

Gegeben ist eine Klasse DenkArray, die die folgenden Attribute a und grenze besitzt:

```
7 public class DenkArray
8 {
9     // Objektvariable
10    private int[] a;
11    private int grenze = 19;
```

(I) Notiere im Schema unten, wie das Array a nach Ausführen von wasMacheIch() aussieht:

```
13 /** Geben Sie an, was passieren wird.
14  */
15 public void wasMacheIch() {
16     a = new int[grenze];
17     for (int i=0; i<grenze; i++) {
18         a[i] = grenze-1-i;
19     }
20 }
21
```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	

(II) Notiere im Schema, wie das Array a nach Ausführen von wasMacheIch2() aussieht:

```
23 /** Geben Sie an, was passieren wird.
24  * Wie sieht die Reihung a danach aus? */
25 public void wasMacheIch2() {
26     a = new int[grenze];
27     for (int i=0; i<grenze; i++) {
28         a[i] = grenze-1-i;
29     }
30     for (int i=0; i<grenze; i++) {
31         a[i] = a[a[i]];
32     }
33 }
34
```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	

**(III)** Notiere im Schema, wie das Array a nach Ausführen von wasMacheIch3() aussieht:

```

36  /** Geben Sie an, was passieren wird.
37  * Wie sieht die Reihung a danach aus? */
38  public void wasMacheIch3() {
39      a = new int[grenze];
40      for (int i=0; i<grenze; i++) {
41          a[i] = i;
42      }
43      for (int i=0; i<grenze; i++) {
44          a[i] = a[a[i]];
45      }
46  }
47  }

```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	

**(IV)** Ergänze die entstehende Reihe sinnvoll. Implementiere dann handschriftlich die Methode wasMacheIch4(), die die Reihe a mit untenstehenden Werten belegt. Gehe dabei sparsam mit Quelltextzeilen um.

1	1	2	3	5	8	13	21	34	55	89												
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	

```

38  public void wasMacheIch () {
39      a = new int[grenze];
40
41
42
43
44
45
46
47  }

```

**(V)** Ergänze die entstehende Reihe sinnvoll. Implementiere dann handschriftlich die Methode wasMacheIch5(), die die Reihe a mit untenstehenden Werten belegt. Gehe dabei sparsam mit Quelltextzeilen um.

1	5	9	13	17	21	25	29	33	37	41												
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	

```
38 public void wasMacheIch () {
39     a = new int[grenze];
40
41
42
43
44
45
46
47 }
```

1)

Die Aufgaben in A3 sind CC-BY-SA Heußler/Zechnall

From: <https://info-bw.de/> -

Permanent link: [https://info-bw.de/faecher:informatik:oberstufe:java:algorithmen:arrays:vermischte\\_aufgaben:start?rev=1698160748](https://info-bw.de/faecher:informatik:oberstufe:java:algorithmen:arrays:vermischte_aufgaben:start?rev=1698160748)

Last update: 24.10.2023 15:19

