

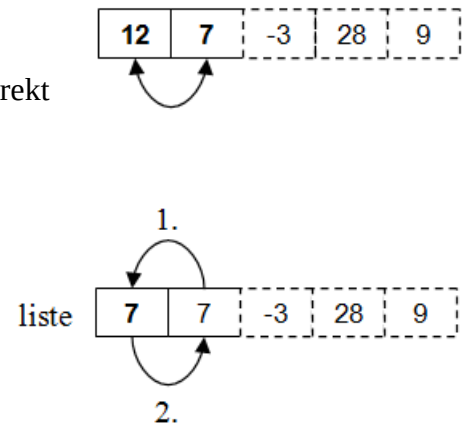
Bevor wir die relativ komplexe Aufgabe angehen, ein Array zu sortieren, erarbeiten wir einige Algorithmen, bei denen Elemente des Arrays verschoben werden.

Grundlage: Vertauschen von zwei Elementen

Als einfaches Beispiel betrachten wir den Fall, dass die ersten beiden Elemente eines Arrays vertauscht werden sollen. Leider gibt es keinen Befehl, um zwei Elemente eines Arrays direkt zu vertauschen – wir müssen es von Hand programmieren.

Versuch 1: Zuweisung

```
public void tauscheDieErsten()
{
    liste[0] = liste[1];
    liste[1] = liste[0];
}
```



Die erste Zuweisung `liste[0] = liste[1]` kopiert den Wert 7 an die erste Stelle des Arrays. Der Wert 12, der vorher dort stand, wird dabei allerdings überschrieben. Die zweite Zuweisung `liste[1] = liste[0]` kann also nur die 7 kopieren. Am Ende haben also beide Elemente den Wert 7, und die 12 ist verschwunden.

Diese Lösung funktioniert also nicht.

Versuch 2: Tauschen mit Zwischenspeicher

```
public void tauscheDieErsten()
{
    int temp;
    temp = liste[0];
    liste[0] = liste[1];
    liste[1] = temp;
}
```

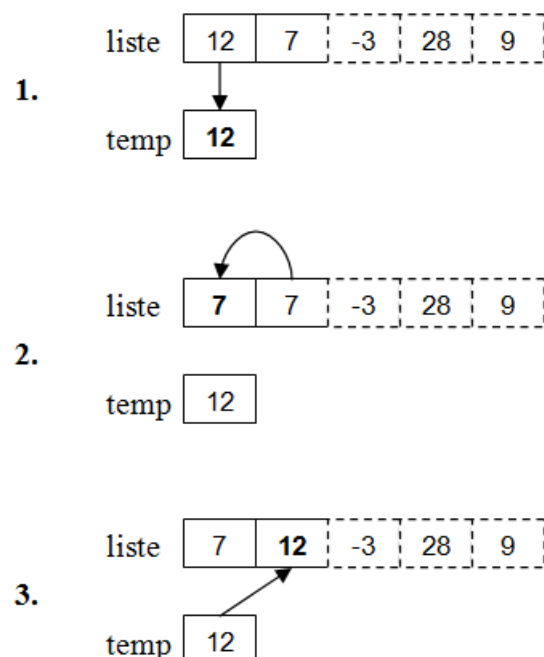
Als Zwischenspeicher wird eine Variable deklariert.

1. Schritt: In der Variable wird der Wert am Anfang der Liste gespeichert (`temp = liste[0]`).

2. Schritt: Jetzt kann `liste[1]` nach `liste[0]` kopiert werden – die 12 wird dabei zwar überschrieben, aber sie ist ja noch in der Variable `temp` gespeichert.

3. Schritt: Die gespeicherte 12 wird von der Variable nach `liste[1]` kopiert.

Danach sind die beiden Elemente der Liste vertauscht.



Aufgabe 1

Formuliere für die folgenden Aufgabenstellungen jeweils einen **Algorithmus**.

Tipp: Zeichne zum Ausprobieren ein Array mit Beispielwerten, sowie ggf. Variablen und spiele deine Idee mit den Beispielwerten durch.

- a) Das erste und das letzte Element des Arrays werden vertauscht.
- b) Die Reihenfolge aller Elemente des Arrays wird umgedreht.
D.h. tausche das erste mit dem letzten, das zweite mit dem vorletzten, usw.
- c) Alle Elemente werden eine Position nach links geschoben.
D.h. das zweite Element wird auf die erste Position geschoben, das dritte auf die zweite usw.
Das erste Element kommt ganz nach rechts (ans Ende).
- d) Alle Elemente werden eine Position nach rechts geschoben, das letzte Element kommt ganz nach links.

Aufgabe 2

Verwende die ausgeteilte BlueJ-Vorlage.

Implementiere die Algorithmen aus Aufgabe 1 (je eine Methode pro Teilaufgabe).

Zusatzaufgaben

- e) Das Array wird „gemischt“:
Mehrere Male (z.B. 50x), werden zwei zufällige Elemente des Arrays vertauscht.
- f) Etwas schwieriger: Alle negativen Zahlen in der Liste werden „gelöscht“.
Dazu rücken die restlichen Elemente jeweils so weit nach vorn, dass die leeren Plätze aufgefüllt werden. Die übrig bleibenden Elemente am Ende des Arrays werden auf 0 gesetzt.
Probiere deine Implementierung mit verschiedenen Beispielwerten aus!