

Arrays und Sortieren

Lernziele:

- Sie verstehen die Begriffe Algorithmen und Datenstrukturen und können diese erklären
- Sie können die elementare Datenstruktur „Array“ anwenden
- Sie kennen grundlegende Sortier-Algorithmen

1 Was haben Algorithmen und Datenstrukturen miteinander zu tun?

Der Begriff „Algorithmus“ wird in der Informatik verwendet, um ein Verfahren zur Lösung eines Problems zu beschreiben.

Vielleicht haben Sie schon in einem Programm einen Algorithmus angewendet, z.Bsp. wenn Sie eine Liste sortieren oder durchsuchen mussten. Algorithmen sind der „Stoff“ der Informatik. Sie stellen sicher, dass ein Programm einen verlässlichen Ablauf hat. Sie definieren, wie man vorgehen soll, um ein Problem zu lösen.

Eine Datenstruktur ist wichtig für den Algorithmus. Die Datenstruktur organisiert die Daten auf eine bestimmte Weise, damit sie effizient und sicher von einem Algorithmus bearbeitet werden können. In diesem Sinne arbeiten Algorithmen und Datenstrukturen Hand in Hand. Je nach Datenstruktur macht es Sinn, einen jeweiligen Algorithmus anzuwenden. Wir werden in diesem Modul das immer wieder anhand von Algorithmen und passenden Datenstrukturen feststellen.

Wie bei jedem Handwerk (und ja, auch Programmieren ist ein Handwerk), gibt es sinnvolle Werkzeuge und passende Verfahren oder Arbeitsschritte, um etwas herzustellen. Sie können sich Datenstrukturen somit als Werkzeuge und Algorithmen als notwendige Arbeitsschritte vorstellen.

2 Eine elementare Datenstruktur: Array

Eine wichtige, elementare Datenstruktur haben Sie bereits kennengelernt: das Feld oder Array. Ein Array ist eine feste Anzahl von einzelnen Daten, welche zusammenhängend gespeichert werden und über einen Index zugänglich sind. Als Programmierer müssen Sie darauf achten, dass die Elemente in einem Array korrekt gespeichert werden, bevor darauf Bezug genommen wird (übrigens ein klassischer Programmierfehler, der sehr häufig vorkommt).

Die Notation eines Arrays in Java sieht folgendermassen aus:

```
long form
double[] a;           ← declaration
a = new double[N];   ← creation
for (int i = 0; i < N; i++)
    a[i] = 0.0;      ← initialization

short form
double[] a = new double[N];

initializing declaration
int[] a = { 1, 1, 2, 3, 5, 8 };
```

3 Arrays anwenden

Nachdem Sie ein Array deklariert und erzeugt haben, können Sie jeden einzelnen Wert ansprechen, indem Sie nach dem Arraynamen einen Integerindex in eckigen Klammern angeben. Die Anzahl Elemente in einem Array ist $N - 1$ (Java bietet ihnen dabei die Variable *length* um die Anzahl Elemente zu erhalten).

Kopieren vs Aliasing

Denken Sie daran, dass sich ein Arrayname immer auf das GANZE Array bezieht. Wenn wir also einen Arraynamen einem anderen Array zuweisen, verweisen beide auf dasselbe Array:

```
int[] a = new int[10];
...
a[i] = 1234;
...
int[] b = a;
...
b[i] = 5678;    //a[i] ist jetzt 5678
```

Eine solche Zuweisung wird „Aliasing“ genannt. Beide verweisen auf dieselbe Adresse (für Leute aus der C-Welt: genau, es handelt sich um einen *Pointer* auf dieselbe Adresse).

Probieren Sie es aus:

Initialisieren Sie einen Array wie oben gezeigt und weisen Sie einen weiteren Array auf den ersten.

Vergleichen Sie die Objekt-Referenzen indem Sie folgendes machen:

```
System.out.println((Object)a.toString());
```

```
System.out.println((Object)b.toString());
```

Beide haben dieselbe Objekt-Referenz. Sie werden also intern als dasselbe Objekt behandelt.

Wenn Sie eine Kopie eines Arrays erzeugen wollen, müssen Sie ein neues Array deklarieren, erzeugen und initialisieren – und dann alle Einträge im ursprünglichen Array in das neue Array kopieren.

Ungefähr so:

```
int N = a.length;
double[] b = new double[N];
for (int i = 0; i < N; i++){
    b[i] = a[i];
}
```

Array Boundaries

Java führt eine automatische Bereichsprüfung durch. Wenn Sie also z.Bsp. zu weit in einem Array suchen (kleiner als 0 oder grösser N-1) dann wird das Programm mit der Exception *ArrayIndexOutOfBoundsException* abgebrochen.

Arbeit mit elementaren Arrays

Wir wollen in den folgenden Aufgaben mit primitiven Arrays arbeiten. Verwenden Sie dazu *statische* Methoden, damit wir uns auf Funktionen beschränken (Funktionen, wie im mathematischen Sinn) und uns nicht mit dem Initialisieren von Objekten befassen müssen (wir können das dann zu einem späteren Zeitpunkt anwenden).

Beispiel:

```
public static void printArrays(int[] numbers){
    for (int i = 0; i < numbers.length; i++){
        System.out.println("current number is " + numbers[i]);
    }
}
```

Natürlich hat Java auch Standardbibliotheken, welche vieles abdecken, welches wir behandeln werden (*Arrays, LinkedList, Sortieren, etc.*). Wir wollen aber zuerst stets versuchen, diese Datenstrukturen oder Algorithmen selber zu implementieren, bevor wir auf vorgefertigte Klassen zurückgreifen.

Benutzereingabe über Konsole oder GUI

Die Wahl der Benutzereingabe (und Ausgabe) werden wir nicht festlegen. Falls Sie mit der Java-Swing Bibliothek oder JavaFX vertraut sind, können sie auch kleinere GUIs erstellen, um die Ein- und Ausgabe zu implementieren. Sie können aber auch die Ein- und Ausgabe über Konsole verwenden.

Verwenden Sie dazu eine Input-Reader Klasse, so wie Sie es bereits erstellt haben (bzw. fragen Sie die Lehrperson).

3.1 Aufgabe: Elemente hinzufügen

Erstellen Sie ein Programm mit zwei Methoden, welche Elemente an das Ende der Liste bzw. Elemente an den Anfang der Liste hinzufügen.

3.2 Aufgabe: Minimum und Maximum in einem Array suchen

Erstellen Sie ein Programm, um das Minimum und Maximum in einem Array zu suchen. Verwenden Sie zehn Elemente mit fix-definierten oder zufällig gewählten Werten.

3.3 Aufgabe: Elemente nachrutschen

Erstellen Sie ein Programm, welches zehn Zahlen einliest (mit Benutzer-Eingabe) und diese entsprechend in einem Array mit fixer Grösse ablegt. Dabei soll jeweils das neue Element an der ersten Stelle im Array eingefügt werden. Überlegen Sie sich, ob ein Array für das Verwalten genügt.

3.4 Aufgabe: Buchstaben umgekehrt ausgeben

Schreiben Sie ein Programm, welches ein eingelesenes Wort in einem Character-Array abspeichert und umgekehrt wieder ausgibt. Also ungefähr so:

```
Ihre Eingabe: hello  
Ausgabe: olleh
```

4 Elementarer Sortier-Algorithmus: der Bubble-Sort

Das Sortieren gehört zu den Standard-Abläufen in der Informatik und reicht von simplen Datensätzen zu grossen Datenvolumen. Es gibt sehr viele Sortier-Algorithmen. Einen Klassiker werden wir hier betrachten, den „Bubble-Sort“.

4.1 Bubble-Sort manuell durchspielen

Bevor wir den Algorithmus in Code umsetzen, werden wir diesen Algorithmus manuell durchspielen. Siehe dazu separates **Arbeitsblatt**.

4.2 Bubble-Sort implementieren

Prinzip: Durchlaufe immer wieder die Datei (oder Liste) und vertausche jedesmal, wenn es notwendig ist, benachbarte Elemente. Wenn bei einem Durchlauf kein Austausch mehr erforderlich ist, ist die Datei sortiert.

In Pseudo-Code:

```
begin
  for i = N to 1 do
    for j = 2 to i do
      if array[j-1] > array[j] then
        e = array[j-1];
        array[j-1] = array[j];
        array[j] = e;
      end;
    end;
  end;
end;
```

Dieser Algorithmus funktioniert, auch wenn er nicht sehr praktisch ist bei grösseren Datenmengen. Jedesmal, wenn während des ersten Durchlaufs das maximale Element vorgefunden wird, wird dieses mit jedem der rechts von ihm befindlichen Elemente vertauscht, und zwar so lange, bis es die Position am rechten Ende des Arrays erreicht hat. Während des zweiten Durchlaufs wird dann das zweitgrösste Element an die richtige Position bewegt usw.

Somit bewegen sich die Elemente wie Blasen („Bubbles“) in einem Gefäss, bis alle Blasen ihren Platz gefunden haben.

6 Optional: Weiterführende Aufgabe mit Bubble-Sort

Der Bubble-Sort ist für numerische wie auch alphanumerische Sortieraufgaben geeignet. Dies darum, weil sich die Problematik in den beiden Fällen nicht unterscheidet : $5 > 2$ aber auch $'D' > 'B'$ (vergleichen Sie dazu die ASCII-Tabelle).

Sollen nun ganze Sätze alphabetisch geordnet werden, müssen mehrere Charakter miteinander verglichen werden. ZBsp. liegt "Muheim" vor "Mutzer".

- ➔ Lesen Sie ein File mit 10 Strings ein und sortieren Sie diese in alphabetischer Reihenfolge.
- ➔ Danach geben Sie die 10 sortierten Strings in der Konsole aus.