Allgemeines

Das *Sortieren* von Daten nach bestimmten Kriterien gehört zu den wichtigsten Problemen der Informatik. Bei großen Datenmengen ist es ungemein wichtig, dass effiziente Algorithmen zur Verfügung stehen. Die verschiedenen in der Geschichte der Informatik entwickelten Algorithmen haben alle ihre Vor- und Nachteile, je nach dem, auf welche Daten man sie loslässt. So kann bei einem bestimmten Datenbestand der „schlechteste“ Algorithmus am schnellsten mit der Arbeit fertig sein.
Zu Beginn werden wir einfachere Algorithmen behandeln und daran studieren, was „Effizienz eines Sortieralgorithmus“ bedeutet.

Praktische Übung

Lege einige Karten eines Kartenspiels gemischt und verdeckt nebeneinander auf den Tisch. Man darf jetzt immer nur gleichzeitig zwei Karten zusammen aufdecken und vergleichen, d.h. nur genau so, wie es ein Computer machen kann. Durch eine bestimmte Strategie sollen die Karten in eine aufsteigend sortierte Reihenfolge gebracht werden.

Programmierung

Mit Hilfe der immer gleichen Vorlage sollen die angegebenen Algorithmen getestet werden.

Vorlage

Entwickle ein Java-Programm, das ein Array mit Integer-Elementen anlegt, mit der Methode `feldFuellen()` das Feld irgendwie mit Daten füllt und mit einer Methode `feldAusgeben()` auf dem Bildschirm ausgeben kann. Dieses Programm soll uns als Vorlage zum Testen der verschiedenen Sortieralgorithmen dienen.
Tipp zum Feldfüllen: Eine zufällige Integerzahl zwischen 0 und 100 erhält man mit

```
feld[i] = (int)Math.round(Math.random()*100.0);
```

Erklärung: `Math.random()` liefert einen `double`-Wert zwischen 0.0 und 1.0. Die Multiplikation mit 100 macht daraus einen Wert zwischen 0.0 und 100.0. `Math.round()` macht daraus einen `long`(Integer)-Wert, der durch Voranstellen von `(int)` als `int`-Wert weiter gegeben wird.

Sortieren durch Minimumsuche: Minsort

- Minimum im Feld suchen und an erste Stelle setzen.
- Minimum im Restfeld suchen und an zweite Stelle setzen.
- u.s.w.

Sortieren durch Vertauschen: Bubblesort

- 1. und 2. Element vergleichen und Maximum an 2. Stelle setzen.
- 2. und 3. Element vergleichen und Maximum an 3. Stelle setzen.
- u.s.w., bis zum Ende des Feldes. Damit ist das Maximum des Feldes ans Ende gewandert.
- Das Verfahren für das Restfeld wiederholen.
- u.s.w., bis kein Restfeld mehr bleibt.

Verbessertes Sortieren durch Vertauschen: „Super“-Bubblesort

- Sortieren wie bei Bubblesort.
- Bei jedem Durchlauf merkt man sich, ob überhaupt zwei Zahlen vertauscht werden mussten. Wenn nicht, ist man schon fertig und kann sich weitere Vergleiche sparen.

Testphase

Teste die Algorithmen nicht nur an Feldern mit zufälligen Inhalten, sondern auch an „fast schon“ sortierten Feldern.