

## Aufgabe 5a: Algorithmen und Datenstrukturen

### Lernziele

Anwenden der modularen Programmierung, Vertiefen der Kenntnisse über die Gültigkeit und Sichtbarkeit von Variablen sowie Erfahrungen sammeln im gemeinsamen Erstellen von Software.

### Allgemeines

Die Aufteilung, wer aus der Zweiergruppe welches Modul implementiert, bleibt Ihnen überlassen. Am Ende des Praktikums müssen alle Module zusammen kompilierbar sein und das Programm die gewünschte Funktionalität aufweisen.

Um Daten innerhalb eines großen Datenbestands schnell finden zu können, muss der Datenbestand nach bestimmten Kriterien geordnet sein. So ist es in einem Telefonbuch zwar möglich, sehr schnell zu einem Teilnehmer die Rufnummer zu ermitteln. Um aber zu einer gegebenen Rufnummer den Teilnehmer zu finden, muss evtl. das gesamte Telefonbuch durchsucht werden.

### Aufgabe

Schreiben Sie ein Programm, das aus einer Datei Auftragsdaten ausliest, die Daten sortiert und eine Statistik mit der Anzahl der durchgeführten Vergleichs- und Vertausche-Operationen ausgibt.

**Teil 1:** Ein Auftrag besteht aus den Attributen `nr`, `kunde`, `artikel`, `menge` und `preis`. Implementieren Sie ein Modul `auftrag` mit folgender Schnittstelle:

```
int compFkt(void *auftr1, void *auftr2);
int toFile(auftrag_t auftrag, FILE *outfile);
auftrag_t fromFile(FILE *infile);
```

**Teil 2:** Implementieren Sie ein Modul `sort`, das beliebige Daten mittels des in der Vorlesung besprochenen Quicksort-Algorithmus sortiert. Bei jedem Sortiervorgang soll die Anzahl der durchgeführten Vergleichs- und Vertausche-Operationen gezählt werden. Verwenden Sie folgende Schnittstelle:

```
void sort(void *data, int n, int size, int (*cmp)(void *, void *),
          int *moves, int *comps);
```

Das Hauptprogramm bekommen Sie von uns gestellt. In dem Hauptprogramm werden die Auftragsdaten aus einer Datei gelesen, der Sortierfunktion zum Sortieren übergeben und anschließend werden die Daten sortiert wieder in einer Datei abgespeichert.

### Testat

Voraussetzung ist jeweils ein fehlerfreies, korrekt formatiertes Programm. Der korrekte Programmlauf muss anhand einer Beispieleingabe nachgewiesen werden. Sie müssen in der Lage sein, Ihr Programm zu erklären.