Universität Ulm - Abteilung Angewandte Informationsverarbeitung
Prof. Dr. Franz Schweiggert und Hans Braxmeier, Sommersemester 2003

3. Übungsblatt zur Vorlesung Allgemeine Informatik II

Abgabetermin: Donnerstag, 22.05.2003

Aufgabe 1: Unternehmensberater *(2 Punkte)*

Ihr Kumpel aus dem Sandkasten (der ohne Abitur!) leitet inzwischen die größte Unternehmensberaterfirma in Ihrer Heimatstadt, während Sie mühsam ihr Bafög (wird wieder nicht erhöht!) in den Kneipen durchbringen („Studieren“). Nur eines haben Sie Ihrem Kumpel voraus: Intelligenz!

Und nachdem er gehört hat, dass alle zur Zeit auf Customer Relationship Management schwören, beauftragt er Sie, seine Karteikartensammlung mal ein bißchen auf Vordermann zu bringen. Denn merke: Oft macht man 20% des Umsatzes mit 3% der Kunden... Nur wer sind diese Kunden bzw. Firmen („Cash Cows“)?

Da Sie sich bestens mit Sortieralgorithmen auskennen (www.sortieralgorithmen.de) schlagen Sie Ihrem Kumpel (Tutor!?) vor, die Sammlung mit Hilfe von Quicksort zu sortieren und erklären Ihm die Funktionsweise sowie die Vor- und Nachteile dieses Algorithmus!

Teil 2: *(8 Punkte)*

Leider ist Ihr Kumpel ein begeisterter Anhänger von Shakersort und bittet Sie daher ein Oberon-Programm zu schreiben, mit dem die Firmendaten von der Tastatur eingelesen und mit Hilfe von SHAKERSORT nach Name, Branche und Umsatz sortiert werden können.

Helfen Sie Ihrem Kumpel, indem Sie eine entsprechende Datenstruktur implementieren (Array of Records zur Verwaltung der Firmendaten) und das Sortierproblem mit Hilfe von Prozedurentypen lösen.

Tips:

- Verwenden Sie zum Sortieren eine Prozedur Sort, der zwei Parameter übergeben werden: Der erste Parameter ist ein ARRAY von Firmen, der zweite Parameter ist eine Vergleichsprozedur (Prozedurtyp), die angibt, nach welchem Feld sortiert wird.
- Zu diesem Prozedurtyp sollen also 3 Vergleichsprozeduren implementiert werden, mit denen die Firmen nach Name, Branche und Umsatz sortiert werden können.

Funktionsweise von Shakersort:

Ein Feld wird einmal vollständig von links nach rechts durchlaufen. Dabei werden die jeweiligen Nachbarelemente miteinander verglichen und ggf. ausgetauscht. Am Ende befindet sich das größte Element am Ende des Feldes. Nun wird das kleinere Teilfeld (ohne das letzte Element) wieder vollständig durchlaufen, aber diesmal von rechts nach links. Dabei werden die jeweiligen Nachbarelemente wieder miteinander verglichen und ggf. ausgetauscht. Am Ende befindet sich das kleinste Element am Anfang des Feldes. Dieser Schritt wird nun mit dem kleineren Teilfeld (Feld ohne das erste und letzte Element) wiederholt und wiederholt und ... Und irgendwann sind wir fertig und die Elemente sind sortiert.

Beispiel:

```
05   37   13   21   02   13       Feld komplett durchlaufen
05   13   21   02   13 | 37       Feld außer letztes Element durchlaufen
02 | 05   13   21   13 | 37       Feld außer erstes und letztes Element...
02 | 05   13   13 | 21   37
02   05 | 13   13 | 21   37
02   05 | 13 | 13   21   37
02   05  13 || 13   21   37       Feld ist sortiert
```

Viel Erfolg!