Universität Ulm - Abteilung Angewandte Informationsverarbeitung

Prof. Dr. Franz Schweiggert
Hans Braxmeier

SS 2002
23. Mai 2002

4. Übungsblatt zur Vorlesung Allgemeine Informatik II

Abgabetermin: 06. Juni 2002

Aufgabe 1: Sortierverfahren - Vierter und letzter Teil *(20 Punkte)*

Der Studentenrecord soll um das Feld `Semester` (Datentyp `INTEGER`) erweitert werden. Hierzu müßt Ihr eine Funktionsprozedur `CompareSemester` implementieren, um die Semester miteinander vergleichen zu können (im Prinzip copy und paste!) und zusaetzlich ein weiteres Unixargument einführen:

- *-v* für Sortieren nach dem Vornamen
- *-n* für Sortieren nach dem Nachnamen
- *-f* für Sortieren nach dem StudienFach
- *-m* für Sortieren nach der Martrikelnummer
- *-s* für Sortieren nach dem Semester

Der Schwerpunkt des neuen Übungsblattes liegt bei der Unixargument- und Unixdateiverarbeitung. Eure Aufgabe ist es, die Studentendaten aus einer Datei einzulesen, die Daten zu Sortieren und abschließend die sortierten Daten wieder in Dateien abzuspeichern!

Dabei sollen die Datendatei und – wie aus dem letzten Übungsblatt bekannt – die Sortierkriterien über Unixargumente angegeben werden können:

BubbleSortFiles studenten.txt -vsm

Die Studentendaten werden in diesem Beispiel aus der Datei *studenten.txt* gelesen und der Reihe nach nach Vorname, Semester und Matrikelnummer sortiert. Die 3 Eregebnisdateien enthalten die Studentendaten sortiert nach Vorname, Student und Matrikelnummer. Der Name der Ergebnisdateien kann im Programm „fest verdrahtet“ werden.

Die Studentendaten sind nach folgendem Muster abgespeichert:

```
Frank:Zeder:Informatik:98538539:1
Andrea:Honold:Biologie:23345321:3
Michaela:Abt:Medizin:12234534:9
Ralf:Maier:Medizin:43565643:12
Julia:Hofer:Wirtschaftsmathematik:87898767:3
Julia:Hof:Medizin:12345679:4
usw.
```

In *Oberon* gibt es eine Funktion `Read.Field` bzw. `Read.FieldS` (für Streams) mit der es möglich ist, Daten getrennt nach Spalten einzulesen. Als Spaltentrenner dient in unserem Fall der Doppelpunkt. In den Übungen werde ich die hierzu benötigten Module und Prozeduren vorstellen und die entsprechenden Beispielprogramme ins Netz stellen.

Um dem Anwender die Benutzung Eures Programms zu erleichtern, soll eine Fehlermeldung ausgegeben werden, falls die Datendatei nicht existiert oder falls keine oder falsche Sortierargumente übergeben werden.

Abschließend soll errechnet werden, welche Studenten (Informatiker, Biologen, WiMas usw...) im Durchschnitt am längsten studieren! Das Ergebnis soll in einer Datei – z.B. studienzeit.txt – abgespeichert werden!

Viel Erfolg!