Universität Ulm - Abteilung Angewandte Informationsverarbeitung

3. Übungsblatt (17.05.01 bis 31.05.01)
zur Vorlesung Allgemeine Informatik II für Wirtschaftswissenschaftler

SS 2001

Wer verträgt am meisten?? (15+5 Punkte)

Ein immer wieder aufkeimender Konflikt zwischen den einzelnen Fakultäten der Uni Ulm ist die Frage nach der Trinkfestigkeit. Jede Gruppe (WiMa's, WiWi's, Biologen...) behauptet von sich, deutlich mehr zu vertragen als die anderen. Diese Frage ("Wer trinkt am meisten Bier auf den Partys?") soll nun mit Ihrer Hilfe genauer untersucht werden!

Zuerst einmal ist es natürlich wichtig, eine zuverlässige Datenquelle zu finden ... zum Beispiel würde sich da ein Uni-Fescht sehr gut eignen. Nehmen wir also mal an, Sie haben auf irgendeiner Party eine größere Gruppe von Leuten gefragt, wieviele Flaschen Bier (in diesem Zusammenhang gelte Cocktail=Longdrink=Bier) sie über den Abend verteilt getrunken haben. Danach müssen die Daten nur noch ausgewertet werden (am besten mit der Berechnung des "durchschnittlichen Bierkonsums pro Person") ... und fertig ist die Statistik!

Ihre Aufgabe in diesem Übungsblatt soll es also sein, ein entsprechendes Oberon-Programm zu schreiben. Hier sind ein paar wichtige Detais dazu:

- Das Programm soll die Daten aus einer Datei einlesen, und zwar mittels File I/O und **nicht** über die Standard-Eingabe. In dieser Datei können die Daten beliebig angeordnet sein, d.h. es spielt keine Rolle, wieviele Zahlen in jeder Zeile stehen und wieviele Zeilen die Datei hat.
- Nach dem Einlesen der Daten sollen die Zahlen in einem Array gespeichert werden. Am besten ist, sie legen über eine Konstante die maximale Anzahl der Datensätze fest.
- Zur Steuerung der Parameter schreiben Sie bitte eine kleine Argumentverarbeitung. Die zulässigen Eingaben sollen der Form **`"-(M|W|B|O) infile outfile"`** genügen. Hierbei drückt der Flag am Anfang das Studienfach aus (M=WiMa, W=WiWi, B=Bio, O=Sonstige), und die beiden anderen Parameter stehen für die Eingabe- und Ausgabedateien (mehr dazu später).
- Die Ausgabe der Ergebnisse soll doppelt erfolgen; einerseits in die angegebene Datei **und** dazu noch auf die Standard-Ausgabe. Ich schlage vor, daß sie auch die gelesenen Daten nochmal mit ausgeben, dann haben Sie eine zusätzliche Kontrolle, ob auch alles richtig verarbeitet wurde (vgl. Beispiele).
- Das zentrale Resultat ist die Berechnung des Durchschnitts über die eingelesenen Daten - also sollten Sie auch dran denken, daß Sie dieses Feature auch nicht am Ende vergessen!

Bei der Auswahl der Daten lasse ich Ihnen freie Hand - Sie können entweder meine Beipieldateien verwenden (die übrigens frei erfunden sind) oder sich selbst welche basteln ... je realistischer, desto besser! An dieser Stelle kann ich aber dann auch gleich betonen, daß Ihr Ergebnis erst ab einer relativ großen Anzahl von Datensätzen statistisch aussagekräftig ist.

Diesmal werde ich Ihnen keine ausdrücklichen DEFINITION Files vorgeben und Ihnen auch keine Vorschriften über den Aufbau des Programms machen. Damit Sie aber trotzdem einen Anreiz haben, gibt es bei diesem Blatt 15 Punkte auf das Programm an sich (d.h. Funktionalität und Erfüllung der geforderten Features) und 5 Punkte auf guten Programmierstil. Zu einem "guten" Stil gehören brauchbare DEFINITION Files, sinnvolle Variablen- und Prozedurbezeichnungen und, nicht zu vergessen, ein **übersichtlicher** und **verständlicher** Aufbau des Programms. Auch gute Kommentare an den richtigen Stellen machen sich nicht schlecht.

Nützliche Hinweise:

- Diesmal empfiehlt es sich auf jeden Fall, mal einen Blick in die Oberon-Dokumentation zu werfen (siehe Link auf der Homepage). Besonders die Module **Streams**, **UnixArguments**, **UnixFiles** und **Strings** sind in diesem Zusammenhang wichtig. Ich empfehle Ihnen, sich vor dem Programmieren auf diesen Seiten ein bißchen umzuschauen.
- Wenn Sie bei der Datei-Steuerung (z.B. mit der Funktion **UnixFiles.Open**()) einen Fehler bekommen, dann sollten Sie das Programm auch direkt terminieren lassen. Eine nützliche Funktion hierfür ist **Process.Exit**(), die als Argument eine (selbst einstellbare) Fehlernummer mitbekommt und dann das Programm sofort beendet.
  Vergessen Sie auch nicht, daß **alle** Fehlermeldungen auf `Streams.stderr` ausgegeben werden und nicht mehr auf `Streams.stdout` (Stichwort "guter Programmierstil").
- Noch ein Wort zum Thema "Daten einlesen": Diesmal müssen Sie wieder mit `eof` arbeiten. Da die Struktur der Daten willkürlich sein soll (s.o.), empfehle ich Ihnen, erst zeilenweise in einen String einzulesen, diesen String dann als Stream zu öffnen und darauf dann **Read.IntS**() anzuwenden.
- Bei der Ausgabe des Durchschnitts brauchen Sie ein Kommando, um REAL-Zahlen auszugeben. Hierfür empfehle ich das Kommando **Print.S1**, das Sie beispielsweise zur Ausgabe einer REAL-Zahl `average` auf einen Stream `outfile` wie folgt benutzen sollten:
  `Print.S1(outfile,"%.2f",average)`

Viel Glück ... und bis zur nächsten Party!!! :-)