Universität Ulm - Abteilung Angewandte Informationsverarbeitung

6. Übungsblatt (21.06.01 bis 28.06.01)
zur Vorlesung Allgemeine Informatik II für Wirtschaftswissenschaftler

SS 2001

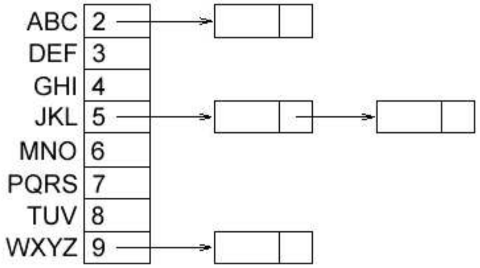
Mein kleines Handy-Telefonbuch (15 Punkte)

Zur Zeit besitzen ca. 50 Millionen Deutsche ein Handy - Tendenz steigend. Unter Studenten ist es ebenfalls sehr beliebt; die meisten von Ihnen werden selbst eins haben. Die häufigste Anwendung fuer das Handy ist das Schreiben von SMS (kurz: *sms-en*), pro Monat werden zig Millionen Kurznachrichten durch die vier deutschen Handynetze verschickt.

Egal ob Sie eine SMS verschicken oder einfach nur einen Freund anrufen wollen, in jedem Fall benutzen Sie garantiert die Telefonbuch-Funktion Ihres Handys. Darin sind die Nummmern all Ihrer Freunde und Bekannten immer griffbereit und abrufbar, außerdem halten sie es natürlich auch immer up-to-date. Ihre Aufgabe in diesem Übungsblatt wird es nun sein, solch ein Handy-Telefonbuch in Oberon zu implementieren.

Die Datenstruktur, die sich für dieses Problem anbietet, ist eine sogenannte *Hash*-Organisation. Was ist das? Ein *Hash Table* ist eine Mischung aus einem klassischen ARRAY und einer Liste. Die Elemente werden mit Hilfe eines Schlüssels (dem *Hash Key*) klassifiziert und dann dem entsprechenden ARRAY-Element zugeordnet. In jedem ARRAY-Element ist nun wiederum der Kopfzeiger für eine Liste abgelegt.
In unserem Fall wird der *Hash Key* diejenige Taste sein, unter der der Name der betreffenden Person im Handy abgelegt ist. Ein Beispiel: Ihr Übungsleiter (**D**avid) würde unter dem Hash Key 3 (steht für DEF als Anfangsbuchstaben) stehen.

Grafisch sieht ein entsprechendes *Hash Table* für ein Handy folgendermaßen aus:



Ich empfehle Ihnen die folgende Typ-Deklaration für Ihr *Hash Table*:

```
TYPE BookEntry = RECORD
                    name : ARRAY 80 OF CHAR;
                    number : ARRAY 13 OF CHAR;
                 END;
     PhoneList = POINTER TO PhoneListRec;
     PhoneListRec = RECORD
                       entry : BookEntry;
                       next : PhoneList;
                    END;
     PhoneBook = ARRAY 10 OF PhoneList;
```

Die Steuerung des Telefonbuchs wird über ein Menü erfolgen. Sie haben in einem anderen Beispielprogramm ja bereits solch eine Programmierung gesehen. Die wesentlichen Funktionen sollten sein:

- Namen aus einer Datei einlesen
- Einzelne Namen hinzufügen
- Einzelne Namen ausgeben
- Einzelne Namen löschen
- Das gesamte Telefonbuch (sortiert nach den Hash Keys) ausgeben

Alles in allem brauchen Sie also (mindestens) die folgenden Prozeduren:

```
PROCEDURE CreateBook(VAR book : PhoneBook) : BOOLEAN;
  (* Creates an empty hash table for the phonebook
   * returns FALSE if any errors occur
   *)

PROCEDURE GetKey(name : ARRAY OF CHAR) : INTEGER;
  (* returns the phonebook key for a given name *)

PROCEDURE IsEmpty(key : INTEGER; book : PhoneBook) : BOOLEAN;
  (* returns TRUE if the list entry in the hash table is empty *)

PROCEDURE Find(name : ARRAY OF CHAR; book : PhoneBook) : INTEGER;
  (* searches the phonebook for the given name
   * returns the hash index on success,
   * 0 if the name could not be found and 1 if there was an error
```

```
    *)

 PROCEDURE Insert(entry : BookEntry; VAR book : PhoneBook);
   (* inserts a name into the phonebook
    * does not insert if the name already exists or if any errors occur
    *)

 PROCEDURE Delete(name : ARRAY OF CHAR; VAR book : PhoneBook);
   (* deletes the given name from the phonebook (if it exists)
    * use Find() first to check if the entry exists!
    *)

 PROCEDURE PrintName(name : ARRAY OF CHAR; book : PhoneBook);
   (* looks for the given name in the phonebook and prints this entry
    * use Find() first to check if the entry exists!
    *)

 PROCEDURE PrintBook(book : PhoneBook);
   (* prints the whole phonebook sorted by hash keys *)

 PROCEDURE ReadNames(fileIN : ARRAY OF CHAR; VAR book : PhoneBook);
   (* reads names and phone numbers from the file specified by 'fileIN'
    * The line format in the infile is 'NAME:PHONE NUMBER'
    * All names are stored in the phonebook using Insert()
    *)
```

Sie finden das komplette DEFINITION File hier. Außerdem gibt es wie immer eine kleine Demonstration des Programms.

Nützliche Hinweise:

- Sie sollten sich bei diesem Übungsblatt durchaus an Blatt 5 sowie dem Beispielprogramm **QueueDemo.om** (vgl. Homepage) orientieren. Vielleicht können Sie die eine oder andere Idee für Ihr Programm verwenden.
- Für Ihr *Hash Table* brauchen Sie vom definierten ARRAY-Typ nur die Felder 2 (ABC) bis 9 (WXYZ). Es ist aber guter Programmierstil, in der Prozedur `CreateBook()` die Listen an den Indizes 0 und 1 mit NIL zu initialisieren.
- Die Prozedur `GetKey()` soll auch Namen verarbeiten können, die mit einem Kleinbuchstaben beginnen. Berücksichtigen Sie dies bei Ihrer Programmierung.
- Die Prozedur `Insert()` soll neue Elemente diesmal am **Anfang** der Liste einfügen. Alternativ können Sie auch versuchen, das Einfügen alphabetisch zu organisieren.
- Es darf (wie bei Ihrem Handy auch) **keine doppelten Einträge** im Telefonbuch geben, d.h. Sie sollten vor dem Einfügen testen, ob der jeweilige Name bereits vorhanden ist.
- Sowohl bei `Insert()` als auch bei `Delete()` müssen Sie 2 Fälle unterscheiden. Eine "Skizze" des Problems hilft Ihnen an diesem Punkt mit Sicherheit weiter!
- `ReadNames()` sollte so programmiert werden, daß Sie bei einem falschen Dateinamen nicht sofort das Programm beenden. Fordern Sie den Benutzer einfach dazu auf, die Eingabe zu wiederholen.
- Da dieses Programm durchaus etwas umfangreicher ist, empfiehlt es sich vielleicht, innerhalb Ihrer Gruppe ein bißchen Arbeitsteilung zu praktizieren. Sie haben dieses Thema ja bereits in der Vorlesung diskutiert.

Viel Erfolg!!!