Lehrstuhl für Programmiersprachen und Übersetzerkonstruktion
CAU Kiel, Informatik und Prakt. Mathematik
Prof. Dr. M. Hanus, Dr. F. Huch

24118 Kiel
Christian-Albrechts-Platz 4
Tel.: 0431 / 880-7277

Programmierpraktikum zur Informatik I
Testat am 18.01., 10-13 Uhr, CAP4, Raum 709

In diesem Testat sollen Sie eine Menüsteuerung und Ein-/Ausgabeprozeduren für die zu entwickelnde Adressdatenbank implementieren.

Beachten Sie aber, dass Sie auch noch **alle** fehlenden Teile des zweiten Aufgabenblattes fetigstellen beziehungsweise korrigieren müssen, da wir diese für die Implementierung der Adressdatenbank benötigen werden.

Aufgabe 6

Entwerfen Sie einen ADT `Address` für die Einträge im Adressbuch. Achten Sie darauf, dass dieser Datentyp auch als Eintrag in dem ADT `Search-Tree` verwendet werden kann. Als Schlüssel sollen Name und Vorname verwendet werden. Geben Sie hierzu Implementierungen der Vergleichsoperationen `less` und `equal` für Listen von Strings an. Strings, welche sich weiter hinten in der Liste befinden, müssen nur verglichen werden, wenn alle vorherigen Strings identisch waren. Mit diesem Ansatz können Sie dann Namen und Vornamen als Schlüssel verwenden. Informationen zu Strings und Vergleichsoperationen für Strings finden Sie in der Hilfe des DrScheme-Systems.
Als weitere Einträge soll die Adresse noch Straße, Hausnummer, Postleitzahl, Ort, Telefonnummer, EMail-Adresse und ein Feld für sonstige Bemerkungen enthalten.

Aufgabe 7

Implementieren Sie eine generische Prozedur `menu`, welche ein Menü auf dem Bildschirm ausgibt, vom Benutzer die Wahl eines Menüpunktes erfragt und diese Wahl als Ergebnis zurückliefert. Als Parameter erhält diese Prozedur eine Liste der Menüpunkte (als Strings). Diese sollen auf dem Bildschirm zusammen mit einer laufenden Nummer ausgegeben werden. Danach wird der Benutzer aufgefordert eine Eingabe zu machen. Ist die Eingabe eine der ausgegebenen Nummern, so wird diese Eingabe zurückgegeben. Sonst soll ein Fehler ausgegeben und eine neue Eingabe erfragt werden.
Hinweis: In Scheme können Werte beliebigen Typs mit Hilfe der Prozedur `(read)` eingelesen werden.

Aufgabe 8

Auch die Eingabe bzw. Ausgabe von Adressen wollen wir generisch implementieren. Dafür verwenden wir eine einfache Form von Masken. Eine Maske ist eine Ein-/Ausgabebeschreibung eines Datensatzes einer Datenbank, indem insbesondere die Komponenten des Datensatzes mit Beschreibungen versehen werden. In dieser einfachen Implementierung einer Datenbank bestehen die Masken einfach nur aus Strings, welche die einzelnen Komponenten benennen. In Anlehnung an die Paarstruktur eines Adressdatensatzes verwenden wir für die Implementierung einer Maske ebenfalls ein Paar von Listen von Strings. Für den ADT aus Aufgabe 4 kann z.B. folgende Maske definiert werden:

```
(define addressMask (make-entry (list "Name:" "Vorname:")
                                (list "Strasse:" "Hausnummer:" ...)))
```

a) Implementieren Sie eine Prozedur `printData`, welche einen Datensatz mit Hilfe einer Maske ausgibt, wie z.B.:

```
Name         : Mustermann
Vornamme     : Karl

Strasse      : Musterallee
Hausnummer   : 0
Postleitzahl: 12345
Ort          : Musterhausen
Telefon      : 01234/56789
EMail        : muster@man.de
Sonstiges    :
```

Beachten Sie, dass zwischen den Maskeneinträgen und den Datensatzeinträgen in der Ausgabe zusätzliche Leerzeichen und zwischen den Schlüsseln und den restlichen Datensätzen eine Leerzeile einzufügen sind. Dies soll in der Funktionalität der Prozedur `printData` enthalten sein und nicht durch Auffüllen der Maskeneinträge mit Leerzeichen implementiert werden.

b) Implementieren Sie eine Prozedur `inputData`, welche einen Datensatz mit Hilfe einer Maske einliest. Nach Eingabe der einzelnen Einträge soll exakt die gleiche Darstellung auf dem Bildschirm sichtbar sein, wie bei der Ausgabe (also gleiche Formatierung verwenden).
Hinweis: Die Prozedur `readline` aus dem Teachpack `io.scm` (siehe Web-Seite zum Praktikum) liest eine Zeile (bis zum Return) von der Tastatur ein und liefert diese als String zurück.