Übungen zu WS 95/96

Algorithmen und Programmierung 3

- Blatt 3 -

H. Schweppe / M. Walter

Abgabe: Do. 9.11.95 bis 12.00

*** Aufgabe 1 (15 P)**

Ein Hochregellager bestehe aus n x m gleichartigen Boxen, in denen je ein zu lagerndes Teil abgelegt werden kann. Das Lager wird von einem Roboter bedient, der sich in x- und y-Richtung bewegen und damit gezielt Boxen anfahren kann. Es soll ein abstrakter Datentyp `Lager` entwickelt werden, der über Operationen wie `einlagern`, `ausgeben`, oder `freierPlatz` verfügt und zur Verwaltung eines Lagers benutzt wird. Ein Codefragment der Lagerverwaltung könnte so aussehen:

```
VAR r: Lager; x,y: CARDINAL;
...
BEGIN
     neuesRegal(r);
     ...
     ok := freierPlatz (r,x1,y1,x,y);
     (* finde Position x,y , die Pos. x1,y1 am naechsten liegt *)
     ...
     ok := einlagern(r,x,y);
     ...
     ok := ausgeben(r,x,y)
     ...
```

Die Beziehung zwischen einer durch ihre Koordinaten gegebenen Box und dem Inhalt (etwa: Teilenummer) wird außerhalb des Moduls festgehalten. Aus Optimierungsgründen, soll ein „möglichst nahe“ an der Position x1, y1 liegender Platz x, y freier Platz gefunden werden (Operation freierPlatz). „Möglichst nah“ soll heißen: die Anzahl Schritte, die der Roboter insgesamt in x- und y-Richtung gehen soll, wenn er vom Punkt(x1,y1) startet, soll minimal sein („Manhatten block distance“).

a) Es soll ein geeigneter Satz von Operationen entworfen und in Form eines Definitionsmoduls mit natürlichsprachlicher Spezifikation präzisiert werden.

b) Entwickeln Sie ein zugehöriges Implementierungsmodul.

c) Spezifizieren und implementieren Sie den abstrakten Datentyp als Miranda - abstype .

***Aufgabe 2 (15 P)**

Es soll ein Taschenrechner implementiert werden, der vollständig geklammerte arithmetische Ausdrücke der folgenden Art auswerten kann:

```
term   ::= num | klammer
klammer::= (term op term)
op     ::= + | - | * | /
num    ::= ziffer | ziffer num
ziffer ::= 0| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Beispiel:
            ((((3 * 4) / 2) - 5) + (99 - 98))
```

a) Gesucht ist eine Implementierung in Modula, die wie oben definierte arithmetische Ausdrücke auswertet. Das Programm soll beendet werden, wenn ein leerer arithmetischer Ausdruck eingegeben wird. Eingabefehler (keine Zahl, fehlerhafter Ausdruck) sollen erkannt werden. Werten Sie nach dem in der Vorlesung besprochenen Verfahren aus. Es duerfen zwar nur positive Zahlen eingegeben werden, das Ergebnis kann negativ sein. Zur weiteren Vereinfachung: / bedeutet ganzzahlige Division (DIV).

Verwenden Sie einen abstrakten Datentyp `Stapel` (Keller, stack) mit geeignetem Basistyp. Dessen Implementierung soll mit Hilfe eines Feldes erfolgen.

b) Ersetzen Sie die Implementierung des ADT `Stapel` durch eine Implementierung mit einer verketteten Liste.