



6. Übungsblatt

Modula-2: Strukturierte Datentypen - Zeiger

Aufgabe 26:

Übertragen Sie den Algorithmus *M* „Minimum einer Zahlenreihe“ vom Beginn dieser Vorlesung (Ende Kapitel 2.1) in ein Modula-2-Programm.

Aufgabe 27:

Schreiben Sie ein Modula-2-Programm, das die ersten 100 Fibonaccizahlen berechnet, speichert und ausgibt. Überlegen Sie ob es sinnvoll ist, die Fibonaccizahlen iterativ oder rekursiv zu berechnen und begründen Sie Ihre Antwort.

Aufgabe 28:

- a) Schreiben Sie ein Modula-2-Programm, das zwei 2×2 -Matrizen *A* und *B* einliest, miteinander multipliziert, das Ergebnis in einer neuen Matrix *C* speichert und formatiert auf dem Bildschirm ausgibt. Verwenden Sie - so weit es möglich ist - Prozeduren, um Ihr Programm übersichtlich zu halten.
Die Rechenvorschrift für die Multiplikation zweier Matrizen *A* und *B* ist wie folgt:

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \quad \Rightarrow \quad AB = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{pmatrix}$$

- b) Ändern Sie Ihr Programm so ab, dass es (ausschließlich) 3×3 -Matrizen verarbeitet.

Aufgabe 29:

Entwerfen Sie eine Datenstruktur in Modula-2 zur Speicherung der Vereine der Fußball-Bundesliga mit all ihren Spielern und den zugehörigen Eigenschaften. Überlegen Sie sich selbständig sinnvolle Attribute und Wertebereiche.

Aufgabe 30: Siehe umseitig

Aufgabe 30:

Gegeben sei das folgende Programm:

```
MODULE Zeigerwirrwarr;  
...  
TYPE ZeigerTyp = POINTER TO StationTyp;  
   StationTyp = RECORD  
       inhalt      : CARDINAL;  
       links,rechts : Zeigertyp;  
   END ;  
VAR Anfang,Ende: Zeigertyp;  
BEGIN  
   NEW(Anfang);  
   Anfang^.inhalt:=7;  
   NEW(Anfang^.links);  
   Anfang^.links^.inhalt:=23;  
   NEW(Anfang^.rechts);  
   Anfang^.rechts^.inhalt:=42;  
   NEW(Ende);  
   Ende^.inhalt:=4;  
   Anfang^.links^.rechts:=Ende;  
   Anfang^.rechts^.links:=Anfang^.links;  
   Anfang^.links^.links:=Anfang;  
   Anfang:=Anfang^.rechts;  
   Anfang^.links^.links^.rechts:=Ende;  
   Anfang^.rechts:=Anfang^.links^.links^.rechts;  
   Ende:=Anfang^.rechts;  
   Ende^.links:=Anfang^.links^.links;  
   Ende^.rechts:=Ende^.links^.links^.rechts;  
END Zeigerwirrwarr.
```

Stellen Sie die durch das Programm `zeigerwirrwarr` erzeugte Datenstruktur graphisch dar. Zeichnen Sie hierzu die erzeugten Objekte vom Typ `StationTyp` als Rechtecke und die Referenzen (Pointer) als Pfeile (vergleiche Skizze).

