Programmierung	WS 2000 / 2001	Prof. H. Lichter A. Nowack

Übung 8

Musterlösung

Aufgabe 8.1:

```
MODULE Zykl_Liste EXPORTS Main;

(* Autor: Antje Nowack
   Umgebung: PM3, Windows95
   Erstellt: 16. 12. 2000
   Letzte Aenderung: 18. 12. 2000
*)

IMPORT SIO;

(*-------------------------Aufgabenteil a-------------------------*)

(* Definition der Datenstruktur. Der Datensatz besteht aus einer
   Zahl vom Typ Integer. Jedes Element besitzt einen Nachfolger und
   einen Vorgaenger. *)

TYPE Link = REF Element;
     Element = RECORD
                 nr : INTEGER;
                 vor, nach : Link;
               END;

(* Die benutzten globalen Variablen werden definiert. kette ist der
   Anker zur aktuellen Liste.
   wahl enthaelt die aktuelle Wahl im Hauptprogramm.
   neu soll beim Einfuegen die einzufuegende Zahl enthalten.
   dummy dient zum Abfangen der Probleme mit SIO.GetInt().
   a,b steuern die Zerstoerung/Beschaedigung der Verkettung einer
   Liste. *)

VAR kette : Link;
    wahl, neu : INTEGER;
    dummy : TEXT;
    a,b : CARDINAL;



PROCEDURE SucheLetzten(kette : Link) : Link =
(* Diese Funktion liefert zu einer Liste den Vorgaenger des Ankers.
*)

(* aktuell enthaelt das aktuell betrachtete Listenelement. *)
VAR aktuell : Link;
  BEGIN
    (* Der Fall der leeren Liste wird nicht betrachtet, da diese
       Funktion nicht mit der leeren Liste aufgerufen wird. *)
    (* Wenn die Liste aus mehr aus einem Element besteht ... *)
```

```
    IF kette.nach # kette THEN
      aktuell := kette^.nach;
      (* Die Schleife wird so lange ausgefuehrt, bis der Vorgaenger
         des Ankers erreicht ist ... *)
      WHILE aktuell^.nach # kette DO
      (* jeweils das naechste Listenelement wird betrachtet ... *)
        aktuell := aktuell^.nach;
      END;
      (* ... und der Vorgaenger des Ankers zurueckgegeben. *)
      RETURN aktuell;
    ELSE
     RETURN kette;
  END;
END SucheLetzten;


PROCEDURE Einfuegen(VAR kette : Link;  neu : INTEGER)=
(* Diese Prozedur fuegt in eine Liste eine Zahl ein. Hierbei wird
   das einzufuegende Element Vorgaenger des Ankers (und Nachfolger
   des urspruenglichen Vorgaengers des Ankers). Der urspruengliche
   Anker bleibt der Anker. *)

VAR elem : Link;
BEGIN
  elem := NEW(Link);
  elem^.nr := neu;
  IF kette = NIL THEN
    elem^.nach := elem;
    elem^.vor := elem;
    kette := elem;
  ELSE
    elem^.nach := kette;
    elem^.vor := SucheLetzten(kette);
    SucheLetzten(kette)^.nach := elem;
    kette^.vor := elem;
  END;
END Einfuegen;


PROCEDURE AusgebenVorwaerts (kette : Link) =
(* Diese Prozedur gibt eine Liste auf dem Bildschirm aus, indem
   zuerst der Inhalt des Ankers ausgegeben und dann jeweils der des
   Nachfolgers des aktuellen Elementes, bis der Vorgaenger des
   Ankers erreicht ist. *)

VAR aktuell : Link;
BEGIN
  IF kette # NIL THEN
    SIO.PutInt(kette^.nr);
    SIO.Nl();
    dummy := SIO.GetLine();
    IF kette.nach # kette THEN
      aktuell := kette^.nach;
      WHILE aktuell # kette DO
        SIO.PutInt(aktuell^.nr);
        SIO.Nl();
        dummy := SIO.GetLine();
        aktuell := aktuell^.nach;
      END;
    END;
```

```
  END;
END AusgebenVorwaerts;


PROCEDURE AusgebenRueckwaerts (kette : Link) =
(* Diese Prozedur gibt eine Liste auf dem Bildschirm aus, indem
   zuerst der Inhalt des Ankers ausgegeben und dann jeweils der des
   Vorgaengers des aktuellen Elementes bis der Nachfolger des Ankers
   erreicht ist. *)

VAR aktuell : Link;
BEGIN
  IF kette # NIL THEN
    SIO.PutInt(kette^.nr);
    SIO.Nl();
    dummy := SIO.GetLine();
    IF kette.vor # kette THEN
      aktuell := kette^.vor;
      WHILE aktuell # kette DO
        SIO.PutInt(aktuell^.nr);
        SIO.Nl();
        dummy := SIO.GetLine();
        aktuell := aktuell^.vor;
      END;
    END;
  END;
END AusgebenRueckwaerts;


PROCEDURE Zerstoeren(VAR kette : Link; a, b : CARDINAL) =
(* Mit Hilfe dieser Prozedur kann die Verkettung einer Liste
   beschaedigt werden. Uebergeben wird die Kette (mit Hilfe eines
   Referenzparameters) und zwei natuerliche Zahlen a und b.
   Der b-te Nachfolger des Ankers wird zum Nachfolgers des a-ten
   Nachfolgers des Ankers. *)

VAR element_a, element_b : Link;
BEGIN
  element_a := kette;
  element_b := kette;
  FOR i := 0 TO a DO
    element_a := element_a^.nach;
  END;
  FOR i := 0 TO b DO
    element_b := element_b^.nach;
  END;
  element_a^.nach := element_b;
END Zerstoeren;

(*-------------------------Aufgabenteil b-------------------------*)


PROCEDURE Test(kette: Link): BOOLEAN=
(* Diese Funktion liefert zu einer doppelt verketteten zyklischen
   Liste einen booleschen Wert. Dieser ist TRUE gdw. die Verkettung
   nicht beschaedigt ist. *)

VAR vorElement, nachElement: Link;
    verkettungOK           : BOOLEAN;
BEGIN
```

```
  (* Zunaechst den Sonderfall der leeren Liste abfangen. *)
  IF kette = NIL THEN
    RETURN TRUE;
  ELSE
    (* Der Test laesst sich auf eine lokale Bedingung reduzieren.
       Wenn NIL auftritt als Nachfolger oder wenn der Vorgaenger des
       Nachfolgers eines Elementes nicht das Element selber ist, so
       ist die Verkettung beschaedigt. Andere Faelle koennen nicht
       auftreten. *)
    nachElement := kette;
    vorElement  := kette^.vor;  (* Nicht leere Kette! *)
    verkettungOK := TRUE;
    (* Ueberpruefung der obigen lokalen Bedingung in einer Schleife
       ... *)
    REPEAT
      IF (vorElement = NIL) OR
         (nachElement # vorElement^.nach) THEN
         verkettungOK := FALSE;
      ELSE
        nachElement := vorElement;
        vorElement := vorElement^.vor;
      END;
    (* ... bis eine Stelle gefunden wurde, die die Beschaedigung der
       Verkettung belegt oder die gesamte Liste durchlaufen wurde
       (ohne eine solche Stelle zu finden). *)
    UNTIL (vorElement = kette^.vor) OR NOT verkettungOK;
    RETURN verkettungOK;
  END;
END Test;

(*-------------------------Hauptprogramm-------------------------*)

(* Steuerung ueber ein Menue *)

BEGIN
  kette := NIL;
  WHILE wahl # 6 DO
    SIO.Nl();
    SIO.PutLine("******************** Menue ********************");
    SIO.Nl();
    SIO.PutLine("Waehlen Sie einen Menuepunkt:");
    SIO.Nl();
    SIO.PutLine("Zahl zur Liste hinzufuegen (1)");
    SIO.Nl();
    SIO.PutLine("Liste vorwaerts ausgeben (2)");
    SIO.Nl();
    SIO.PutLine("Liste rueckwaerts ausgeben (3)");
    SIO.Nl();
    SIO.PutLine("Zerstoeren der Verkettung (4)");
    SIO.Nl();
    SIO.PutLine("Ueberpruefung der Verkettung (5)");
    SIO.Nl();
    SIO.PutLine("Programm beenden (6)");
    SIO.Nl();
    SIO.PutLine("****************** Ende Menue ******************");
    SIO.Nl();
    SIO.PutText("Wahl: ");
    wahl := SIO.GetInt();
    dummy := SIO.GetLine();
```

```
    IF wahl = 1 THEN
      SIO.PutText("Bitte geben Sie die einzufuegende Zahl ein: ");
      neu := SIO.GetInt();
      dummy := SIO.GetLine();
      Einfuegen(kette, neu);

    ELSIF wahl = 2 THEN
      SIO.PutLine("Die Liste sieht vorwaerts wie folgt aus: ");
      AusgebenVorwaerts(kette);
      SIO.PutLine("Hier schliesst sich der Zykel. ");

    ELSIF wahl = 3 THEN
      SIO.PutLine("Die Liste sieht rueckwaerts wie folgt aus: ");
      AusgebenRueckwaerts(kette);
      SIO.PutLine("Hier schliesst sich der Zykel. ");

    ELSIF wahl = 4 THEN
      SIO.PutText("Bitte geben Sie die erste Position an: ");
      a := SIO.GetInt();
      dummy := SIO.GetLine();
      SIO.PutText("Bitte geben Sie die zweite Position an: ");
      b := SIO.GetInt();
      dummy := SIO.GetLine();
      Zerstoeren(kette, a, b);

   ELSIF wahl = 5 THEN
      IF Test(kette) THEN
        SIO.PutLine("Die Verkettung ist NICHT BESCHAEDIGT.");
      ELSE
        SIO.PutLine("Die Verkettung ist BESCHAEDIGT.");
      END;

    ELSIF wahl # 6 THEN
      SIO.PutLine("Keine korrekte Eingabe. Bitte wiederholen.");
    END;
  END;
  SIO.PutLine("Programmende!");
END Zykl_Liste.
```

Probelauf des Programms:

```
************************* Menue *************************

Waehlen Sie einen Menuepunkt:

Zahl zur Liste hinzufuegen (1)

Liste vorwaerts ausgeben (2)

Liste rueckwaerts ausgeben (3)

Zerstoeren der Verkettung (4)

Ueberpruefung der Verkettung (5)

Programm beenden (6)

************************* Ende Menue *************************
```

```
Wahl:
1
Bitte geben Sie die einzufuegende Zahl ein: 1

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 1
Bitte geben Sie die einzufuegende Zahl ein: 2

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 1
Bitte geben Sie die einzufuegende Zahl ein: 3

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 1
Bitte geben Sie die einzufuegende Zahl ein: 4

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 1
Bitte geben Sie die einzufuegende Zahl ein: 5

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 2
Die Liste sieht vorwaerts wie folgt aus:
1

2

3

4

5


Hier schliesst sich der Zykel.

************************* Menue ***************
...
************************* Ende Menue ***********

Wahl:3
Die Liste sieht rueckwaerts wie folgt aus:
1

5
```

```
4

3

2

Hier schliesst sich der Zykel.

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 5
Die Verkettung ist NICHT BESCHAEDIGT.

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 4
Bitte geben Sie die erste Position an: 2
Bitte geben Sie die zweite Position an: 4

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 5
Die Verkettung ist BESCHAEDIGT.

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 4
Bitte geben Sie die erste Position an: 2
Bitte geben Sie die zweite Position an: 4

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 5
Die Verkettung ist BESCHAEDIGT.

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 2
Die Liste sieht vorwaerts wie folgt aus:
1

2

3


4
```

```
Hier schliesst sich der Zykel.

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 3
Die Liste sieht rueckwaerts wie folgt aus:
1

5

4

3


2

Hier schliesst sich der Zykel.

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 6
Programmende!
```

Aufgabe 8.2:

```
MODULE Schuelerliste EXPORTS Main;

(* Autor: Antje Nowack
   Umgebung: PM3, Windows95
   Erstellt: 16. 12. 2000
   Letzte Aenderung: 21. 12. 2000
*)

IMPORT SIO, Math;

(*----------------------Aufgabenteil a --------------------------*)

(* Ein Schueler wird durch einen Verbund/Record mit den
   entsprechenden Datentypen realisiert. *)
TYPE Schueler = RECORD
                  Name : TEXT;
                  Vorname : TEXT;
                  Alter : REAL;
                  Note : REAL;
                END;

     (* Mit Hilfe der Datenstruktur SchuelerRef laesst sich eine
        einfach verkettete Liste, in der Schueler abgelegt sind,
        realisieren. *)
     SchuelerRef = REF Element;
     Element = RECORD
                  schueler : Schueler;
```

```
                naechster : SchuelerRef;
            END;

     (* Die folgenden beiden Typen sind Prozedurtypen.
        Operation enthaelt alle zweistelligen Funktionen auf REAL,
        welche einen REAL-Wert zurueckliefern. Dies sind z.B.
        Addition und Multiplikation. *)
     Operation = PROCEDURE (a,b : REAL) : REAL;

     (* Projektion enthaelt alle einstelligen Funktionen, die
        Schueler auf REAL abbilden. Dies sind z.B. Die Funktionen,
        die einen Schueler auf seine Note oder sein Alter abbilden.
     *)
     Projektion = PROCEDURE (schueler : Schueler) : REAL;

     (* Vergleich enthaelt alle zweistelligen Relationen, d.h.
        booleschen Funktionen, auf reellen Zahlen, z.B. die <- und
        die >-Relation. *)
     Vergleich = PROCEDURE (a,b : REAL) : BOOLEAN;

(* ... einige Hilfsvariablen ... *)
VAR schuelerListe : SchuelerRef := NIL;
    schueler : Schueler;
    wahl: INTEGER;
    dummy : TEXT;
    sortiert : SchuelerRef;
    limitAlter, limitNote : REAL;


PROCEDURE Einfuegen(VAR liste : SchuelerRef; neu : Schueler)=
(* Diese Prozedur fuegt einen Schueler in eine Liste ein. Dieser
   wird an den Beginn der Liste gesetzt. Hierbei wird nicht
   ueberpruft, ob der Schueler bereits in der Liste vorhanden ist.
*)

VAR elem : SchuelerRef;
BEGIN
  elem := NEW(SchuelerRef);
  elem^.schueler := neu;
  elem^.naechster := liste;
  liste := elem;
END Einfuegen;


PROCEDURE Ausgeben(liste : SchuelerRef) =
(* Diese Prozedur gibt eine Liste auf dem Bildschirm aus, indem die
   gesamte Liste durchlaufen wird. *)

BEGIN
  IF liste # NIL THEN
    SIO.PutText("Name: ");
    SIO.PutText(liste^.schueler.Name);
    SIO.Nl();
    SIO.PutText("Vorname: ");
    SIO.PutText(liste^.schueler.Vorname);
    SIO.Nl();
    SIO.PutText("Alter: ");
    SIO.PutReal(liste^.schueler.Alter);
    SIO.Nl();
    SIO.PutText("Note: ");
```

```
    SIO.PutReal(liste^.schueler.Note);
    SIO.Nl();
    SIO.Nl();
    SIO.PutText("-----------------------------------------------");
    SIO.Nl();
    dummy := SIO.GetLine();
    SIO.Nl();
    Ausgeben(liste^.naechster);
    SIO.Nl();
  END;
END Ausgeben;


(*----------------------Aufgabenteil b ---------------------------*)


PROCEDURE Alter (schueler : Schueler) : REAL =
(* Diese Funktion bildet einen Schueler auf sein Alter ab. *)

BEGIN
  RETURN schueler.Alter;
END Alter;


PROCEDURE Note (schueler : Schueler) : REAL =
(* Die folgende Funktion bildet einen Schueler auf seine Note ab. *)

BEGIN
  RETURN schueler.Note;
END Note;

PROCEDURE A_groesser_B ( a, b : REAL) : BOOLEAN =
BEGIN
 RETURN a > b;
END A_groesser_B;

PROCEDURE A_kleiner_B ( a, b : REAL) : BOOLEAN =
BEGIN
 RETURN a < b;
END A_kleiner_B;

PROCEDURE StelleSuchen(liste : SchuelerRef; person : Schueler; proj
: Projektion; rel : Vergleich) : SchuelerRef =
(* Mit Hilfe dieser Prozedur wird zu einem Schueler und einer
  Schuelerliste die Stelle gesucht, wo der Schueler eingefuegt
  werden muss. Hierbei wird von einer sortierten Liste gemaess rel
  ausgegangen. *)

VAR stelle, vergleich : SchuelerRef;
BEGIN
  vergleich := liste;
  stelle := liste;
  WHILE (vergleich # NIL) AND NOT rel(proj ( vergleich^.schueler),
proj(person)) DO
    stelle := vergleich;
    vergleich := vergleich^.naechster;
  END;
  RETURN stelle;
END StelleSuchen;
```

```
PROCEDURE Filter( liste : SchuelerRef; proj : Projektion; rel :
Vergleich; limit : REAL; VAR filterListe : SchuelerRef) =
(* Diese Prozedur liefert zu einer einfach verketteten Liste, einer
   Funktion, die einen Schueler auf ein REAL-Zahl abbildet und einer
   REAL-Zahl die Liste, die Liste der Schueler, deren Bild unter der
   Funktion ueber der Zahl liegen. Die Datensaetze der produzierten
   Liste sind Kopien der Original-Liste und diese ist nach Werten
   der uebergebenen Funktion sortiert gemaess rel. *)

VAR kopie : SchuelerRef;
BEGIN
  IF liste # NIL THEN
    (* Speicherplatz reservieren *)
    kopie := NEW(SchuelerRef);
    (* Datensatz des aktuell betrachteten Listenelementes kopieren
     *)
    kopie^.schueler := liste^.schueler;
    (* nur wenn die Grenze ueberschritten wurde, wird der schueler
in die
       resultierende Liste aufgnommen. *)
    IF rel ( proj(kopie^.schueler), limit) THEN
        (* Die erste Operation zum Einfuegen kann nur durchgefuehrt
           werden, wenn die resultierende Liste bereits mindestens
           ein Element enthaelt und der Funktionswert muss kleiner
           sein als der einzufuegende. Dann wird das neue Element an
           der berechneten Stelle eingefuegt. *)
        IF filterListe # NIL AND
          rel (proj(kopie^.schueler),proj(filterListe^.schueler))
        THEN
          kopie^.naechster :=
                       StelleSuchen(filterListe, kopie^.schueler,
                                    proj, rel)^.naechster;
          StelleSuchen(filterListe, kopie^.schueler, proj,
                                         rel)^.naechster := kopie;
        (* Sonst wird das neue Element an den Anfang der bis jetzt
           berechneten resultierenden Liste gesetzt. *)
        ELSE
          kopie^.naechster := filterListe;
          filterListe := kopie;
        END;
    END;
    (* rekursiver Aufruf der Prozedur *)
    Filter(liste^.naechster, proj, rel, limit, filterListe);
  END;
END Filter;


PROCEDURE FilterAlter(liste: SchuelerRef; limit : REAL; VAR ergebnis
: SchuelerRef) =
(* Diese Prozedur wendet die Prozedur Filter an, wobei durch die
   uebergebene Funktion jeder Schueler auf sein Alter projiziert
   wird und > als Vergleichsrelation uebergeben wird. *)

BEGIN
  Filter( liste, Alter, A_groesser_B,limit, ergebnis);
END FilterAlter;
```

```
PROCEDURE FilterNote_schlechter_spaeter(liste: SchuelerRef; limit :
REAL; VAR ergebnis : SchuelerRef) =
(* Diese Prozedur wendet die Prozedur Filter an, wobei durch die
   uebergebene Funktion jeder Schueler auf seine Note projiziert
   wird und < als Vergleichsrelation uebergeben wird. *)

BEGIN
  Filter( liste, Note, A_groesser_B, limit, ergebnis);
END FilterNote_schlechter_spaeter;

PROCEDURE FilterNote_besser_spaeter(liste: SchuelerRef; limit :
REAL; VAR ergebnis : SchuelerRef) =
(* Diese Prozedur wendet die Prozedur Filter an, wobei durch die
   uebergebene Funktion jeder Schueler auf seine Note projiziert
   wird und < als Vergleichsrelation uebergeben wird. *)

BEGIN
  Filter( liste, Note, A_kleiner_B, limit, ergebnis);
END FilterNote_besser_spaeter;


PROCEDURE Filtern_schlechter_spaeter(liste: SchuelerRef; limitAlter,
limitNote : REAL; VAR ergebnis : SchuelerRef) =
(* Diese Prozedur wendet die beiden vorhergehenden Prozeduren an.
   Zuerst diejenige fuer die Filterung nach Alter. Auf die hieraus
   resultierende Liste wird der Filter nach Noten angewandt, welcher
   die Schueler liefert, die schlechter als eine einzugebende Grenze
   enthaelt.*)

VAR zwischenErgebnis : SchuelerRef;
BEGIN
  FilterAlter(liste, limitAlter, zwischenErgebnis);
  FilterNote_schlechter_spaeter(zwischenErgebnis, limitNote,
ergebnis);
END Filtern_schlechter_spaeter;

PROCEDURE Filtern_besser_spaeter(liste: SchuelerRef; limitAlter,
limitNote : REAL; VAR ergebnis : SchuelerRef) =
(* Diese Prozedur wendet die beiden vorhergehenden Prozeduren an.
   Zuerst diejenige fuer die Filterung nach Alter. Auf hieraus
   resultierende Liste wird der Filter nach Noten angewandt, welcher
   die Schueler liefert, die besser als eine einzugebende Grenze
   enthaelt.*)

VAR zwischenErgebnis : SchuelerRef;
BEGIN
  FilterAlter(liste, limitAlter, zwischenErgebnis);
  FilterNote_besser_spaeter(zwischenErgebnis, limitNote, ergebnis);
END Filtern_besser_spaeter;


(*----------------------Aufgabenteil c -------------------------*)


PROCEDURE Anwenden(op : Operation; liste : SchuelerRef; init : REAL)
: REAL =
(* Dieser Prozedur werden eine zweistellige Funktion auf REAL-Werten
   uebergeben, eine Schuelerliste und ein REAL-Wert uebergeben. Die
   Funktion wird sukzessive auf die Noten der Liste angewandt. Der
   resultierende Wert wird zurueckgegeben. Der uebergebene REAL-Wert
```

```
   dient hierbei als Startwert.
   Die sukzessive Anwendung wird durch rekusiven Aufruf der Prozedur
   durchgefuehrt. *)

BEGIN
  IF liste = NIL THEN
    RETURN init;
  ELSE
    init := op(init, liste^.schueler.Note);
    RETURN Anwenden(op, liste^.naechster, init);
  END;
END Anwenden;


PROCEDURE Add(a,b : REAL) : REAL =
(* Addition *)

BEGIN
  RETURN (a+b);
END Add;


PROCEDURE Mult(a,b : REAL) : REAL =
(* Multiplikation *)

BEGIN
  RETURN (a*b);
END Mult;


PROCEDURE Inc(a, b : REAL) : REAL =
(* Inkrementierung des ersten Wertes *)

BEGIN
  RETURN (a + 1.0)
END Inc;


PROCEDURE Arithm_Mittel(liste : SchuelerRef) : REAL =
(* Diese Funktion liefert zu einer Schuelerliste das arithmetische
   Mittel der Noten. *)

BEGIN
  IF liste # NIL THEN
    (* Nach Definition des arithmetischen Mittels wird die Summe der
       Noten durch die Anzahl der Elemente in der Liste dividiert.
       Die Summe wird berechnet durch sukzessives Anwenden der
       Addition. Startwert ist hierbei 0.
       Die Anzahl der Elemente kann hierbei durch sukzessives
       Anwenden der Funktion Inc berechnet werden. Pro Listenelement
       wird um 1 erhoeht. *)
    RETURN (Anwenden(Add, liste, 0.0) / Anwenden(Inc, liste, 0.0));
  ELSE
    (* Bei der leeren Liste wird 0 zurueckgeliefert. *)
    RETURN 0.0;
  END;
END Arithm_Mittel;


PROCEDURE Geom_Mittel(liste : SchuelerRef) : REAL =
```

```
(* Diese Funktion liefert zu einer Schuelerliste das geometrische
   Mittel der Noten. *)

BEGIN
  IF liste # NIL THEN
  (* Nach Definition des geometrischen Mittels wird aus dem Produkt
     der Noten die n-te Wurzel gezogen. n ist hierbei die Anzahl der
     Elemente in der Liste.
     Das Produkt wird berechnet durch sukzessives Anwenden der
     Multiplikation. Startwert ist hierbei 1.
     Die Anzahl der Elemente kann hierbei durch sukzessives Anwenden
     der Funktion Inc berechnet werden. Pro Listenelement wird um 1
     erhoeht.
     Die Verwendung von FLOAT ist notwendig, da die
     Potenzierungfunktion pow aus Math fuer LONGREAL-Werte definiert
     ist und Noten REAL-Werte sind. *)
     RETURN FLOAT( Math.pow ( FLOAT ( Anwenden(Mult, liste, 1.0),
                                      LONGREAL),
                              FLOAT(1.0 / Anwenden(Inc, liste, 0.0),
                                      LONGREAL)), REAL);
  ELSE
    RETURN 1.0 ;
  END;
END Geom_Mittel;


(*--------------------- Hauptprogramm ----------------------------*)

PROCEDURE SchuelerEingabe() : Schueler =
(* Eingabe eines Schuelers durch Eingabe der einzelnen Werte *)

VAR schueler : Schueler;
  BEGIN
  SIO.PutLine("Bitte den Namen des Schuelers eingeben: ");
  schueler.Name := SIO.GetLine();
  SIO.PutLine("Bitte den Vornamen des Schuelers eingeben: ");
  schueler.Vorname := SIO.GetLine();
  SIO.PutLine("Bitte das Alter des Schuelers eingeben: ");
  schueler.Alter := SIO.GetReal();
  SIO.PutLine("Bitte die Note des Schuelers eingeben: ");
  schueler.Note := SIO.GetReal();
  RETURN schueler;
END SchuelerEingabe;

(* Menue *)

BEGIN
  wahl := 0;
  WHILE wahl # 10 DO
    SIO.PutLine("******************** Menue ********************");
    SIO.PutLine("Waehlen Sie einen Menuepunkt:");
    SIO.Nl();
    SIO.PutLine("Schueler zur Liste hinzufuegen (1)");
    SIO.Nl();
    SIO.PutLine("Schuelerliste ausgeben (2)");
    SIO.Nl();
    SIO.PutLine("Schueler ueber einem einzugebenden Alter ausgeben
                 (3)");
    SIO.Nl();
    SIO.PutLine("Schueler schlechter als eine einzugebende Note
```

```
            ausgeben (4)");
SIO.Nl();
SIO.PutLine("Schueler besser als eine einzugebende Note ausgeben
            (5)," );
SIO.Nl();
SIO.PutLine("Schueler ueber einem einzugebenden Alter und
            schlechter");
SIO.PutLine("schlechter als eine einzugebenden Note ausgeben
            (6)");
SIO.Nl();
SIO.PutLine("Schueler ueber einem einzugebenden Alter und
            besser");
SIO.PutLine("als eine einzugebende Note ausgeben (7)");
SIO.Nl();
SIO.PutLine("Arithmetisches Mittel der Noten berechnen (8)" );
SIO.Nl();
SIO.PutLine("Geometrisches Mittel der Noten berechnen (9)" );
SIO.Nl();
SIO.PutLine("Programm beenden (10)");
SIO.PutLine("****************** Ende Menue ******************");
SIO.Nl();
SIO.PutText("Wahl: ");
wahl := SIO.GetInt();
dummy := SIO.GetLine();
IF wahl = 1 THEN
  schueler := SchuelerEingabe();
  Einfuegen(schuelerListe, schueler);
ELSIF wahl = 2 THEN
  Ausgeben(schuelerListe);
ELSIF wahl = 3 THEN
  sortiert := NIL;
  SIO.PutLine("Bitte die Altersgrenze eingeben: ");
  limitAlter := SIO.GetReal();
  dummy := SIO.GetLine();
  SIO.Nl();
  SIO.PutLine("Schueler ueber ");
  SIO.PutReal(limitAlter);
  SIO.PutLine(" (aufsteigend sortiert nach Alter):");
  SIO.Nl();
  FilterAlter(schuelerListe, limitAlter, sortiert);
  Ausgeben(sortiert);

 ELSIF wahl = 4 THEN
  sortiert := NIL;
  SIO.PutLine("Bitte die Notengrenze eingeben: ");
  limitNote := SIO.GetReal();
  dummy := SIO.GetLine();
  SIO.Nl();
  SIO.PutText("Schueler mit Note ueber ");
  SIO.PutReal(limitNote);
  SIO.PutLine(" (sortiert nach Note):");
  SIO.Nl();
  FilterNote_schlechter_spaeter(schuelerListe, limitNote,
                                sortiert);
  Ausgeben(sortiert);

ELSIF wahl = 5 THEN
  sortiert := NIL;
  SIO.PutLine("Bitte die Notengrenze eingeben: ");
  limitNote := SIO.GetReal();
```

```
    dummy := SIO.GetLine();
    SIO.Nl();
    SIO.PutText("Schueler mit Note unter ");
    SIO.PutReal(limitNote);
    SIO.PutLine(" (sortiert nach Note):");
    SIO.Nl();
    FilterNote_besser_spaeter(schuelerListe, limitNote, sortiert);
    Ausgeben(sortiert);

  ELSIF wahl = 6 THEN
    sortiert := NIL;
    SIO.PutLine("Bitte die Altersgrenze eingeben: ");
    limitAlter := SIO.GetReal();
    dummy := SIO.GetLine();
    SIO.PutLine("Bitte die Notengrenze eingeben: ");
    limitNote := SIO.GetReal();
    dummy := SIO.GetLine();
    SIO.Nl();
    SIO.PutText("Schueler ueber ");
    SIO.PutReal(limitAlter);
    SIO.PutLine(" und Note ueber ");
    SIO.PutReal(limitNote);
    SIO.PutLine(" (sortiert nach Note):");
    SIO.Nl();
    Filtern_schlechter_spaeter(schuelerListe, limitAlter,
                               limitNote, sortiert);
    Ausgeben(sortiert);

  ELSIF wahl = 7 THEN
    sortiert := NIL;
    SIO.PutLine("Bitte die Altersgrenze eingeben: ");
    limitAlter := SIO.GetReal();
    dummy := SIO.GetLine();
    SIO.PutLine("Bitte die Notengrenze eingeben: ");
    limitNote := SIO.GetReal();
    dummy := SIO.GetLine();
    SIO.Nl();
    SIO.PutText("Schueler ueber ");
    SIO.PutReal(limitAlter);
    SIO.PutLine(" und Note unter ");
    SIO.PutReal(limitNote);
    SIO.PutLine(" (sortiert nach Note):");
    SIO.Nl();
    Filtern_besser_spaeter(schuelerListe, limitAlter, limitNote,
                           sortiert);
    Ausgeben(sortiert);

   ELSIF wahl = 8 THEN
     SIO.Nl();
     SIO.Nl();
     IF schuelerListe = NIL THEN
       SIO.PutLine("Die Liste ist leer!");
     ELSE
       SIO.PutText("Arithmetisches Mittel der Noten: ");
       SIO.PutReal(Arithm_Mittel(schuelerListe));
     END;
     SIO.Nl();
    dummy := SIO.GetLine();
   ELSIF wahl = 9 THEN
     SIO.Nl();
```

```
      SIO.Nl();
      IF schuelerListe = NIL THEN
        SIO.PutLine("Die Liste ist leer!");
      ELSE
        SIO.PutText("Geometrisches Mittel der Noten: ");
        SIO.PutReal(Geom_Mittel(schuelerListe));
      END;
      SIO.Nl();
      dummy := SIO.GetLine();
   ELSIF wahl # 10 THEN
      SIO.PutLine("Unzulaessige Eingabe! Bitte neu waehlen!");
      SIO.Nl();
    END;
  END;
SIO.PutLine("Programmende!");
END Schuelerliste.
```

Probelauf des Programms:

```
************************* Menue *************************
Waehlen Sie einen Menuepunkt:

Schueler zur Liste hinzufuegen (1)

Schuelerliste ausgeben (2)

Schueler ueber einem einzugebenden Alter ausgeben (3)

Schueler schlechter als eine einzugebende Note ausgeben (4)

Schueler besser als eine einzugebende Note ausgeben (5),

Schueler ueber einem einzugebenden Alter und schlechter
schlechter als eine einzugebenden Note ausgeben (6)

Schueler ueber einem einzugebenden Alter und besser
als eine einzugebende Note ausgeben (7)

Arithmetisches Mittel der Noten berechnen (8)

Geometrisches Mittel der Noten berechnen (9)

Programm beenden (10)
************************* Ende Menue *************************

Wahl: 1
Bitte den Namen des Schuelers eingeben:
Schlechterschueler
Bitte den Vornamen des Schuelers eingeben:
Emil
Bitte das Alter des Schuelers eingeben:
20
Bitte die Note des Schuelers eingeben:
5.0

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 1
```

```
Bitte den Namen des Schuelers eingeben:
Jungerschueler
Bitte den Vornamen des Schuelers eingeben:
Daniel
Bitte das Alter des Schuelers eingeben:
15
Bitte die Note des Schuelers eingeben:
2.0

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 1
Bitte den Namen des Schuelers eingeben:
Alterschueler
Bitte den Vornamen des Schuelers eingeben:
Christina
Bitte das Alter des Schuelers eingeben:
25
Bitte die Note des Schuelers eingeben:
4.0

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 1

Bitte den Namen des Schuelers eingeben:
Musterschueler
Bitte den Vornamen des Schuelers eingeben:
Berta
Bitte das Alter des Schuelers eingeben:
17
Bitte die Note des Schuelers eingeben:
1.0

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 1

Bitte den Namen des Schuelers eingeben:
Schueler
Bitte den Vornamen des Schuelers eingeben:
Anton
Bitte das Alter des Schuelers eingeben:
18
Bitte die Note des Schuelers eingeben:
3.0

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 2
Name: Schueler
Vorname: Anton
```

```
Alter: 18
Note: 3

-----------------------------------------------

Name: Musterschueler
Vorname: Berta
Alter: 17
Note: 1

-----------------------------------------------

Name: Alterschueler
Vorname: Christina
Alter: 25
Note: 4

-----------------------------------------------

Name: Jungerschueler
Vorname: Daniel
Alter: 15
Note: 2

-----------------------------------------------

Name: Schlechterschueler
Vorname: Emil
Alter: 20
Note: 5

-----------------------------------------------

************************** Menue **************************
...
************************** Ende Menue **************************

Wahl: 3
Bitte die Altersgrenze eingeben:
17

Schueler ueber
17 (aufsteigend sortiert nach Alter):

Name: Schueler
Vorname: Anton
Alter: 18
Note: 3

-----------------------------------------------

Name: Schlechterschueler
Vorname: Emil
Alter: 20
Note: 5

-----------------------------------------------

Name: Alterschueler
Vorname: Christina
```

```
Alter: 25
Note: 4

-----------------------------------------------

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 4
Bitte die Notengrenze eingeben:
2.5

Schueler mit Note ueber 2.5 (sortiert nach Note):

Name: Schueler
Vorname: Anton
Alter: 18
Note: 3

-----------------------------------------------

Name: Alterschueler
Vorname: Christina
Alter: 25
Note: 4

-----------------------------------------------

Name: Schlechterschueler
Vorname: Emil
Alter: 20
Note: 5

-----------------------------------------------

************************* Menue *************************
...
************************* Ende Menue *************************
Wahl: 5
Bitte die Notengrenze eingeben:
4

Schueler mit Note unter 4 (sortiert nach Note):

Name: Schueler
Vorname: Anton
Alter: 18
Note: 3

-----------------------------------------------

Name: Jungerschueler
Vorname: Daniel
Alter: 15
Note: 2

-----------------------------------------------

Name: Musterschueler
```

```
Vorname: Berta
Alter: 17
Note: 1

------------------------------------------------

************************* Menue **************************
...
************************* Ende Menue **************************

Wahl: 6
Bitte die Altersgrenze eingeben:
16
Bitte die Notengrenze eingeben:
3

Schueler ueber 16 und Note ueber
3 (sortiert nach Note):

Name: Alterschueler
Vorname: Christina
Alter: 25
Note: 4

------------------------------------------------

Name: Schlechterschueler
Vorname: Emil
Alter: 20
Note: 5

------------------------------------------------

************************* Menue **************************
...
************************* Ende Menue **************************

Wahl: 7
Bitte die Altersgrenze eingeben:
15
Bitte die Notengrenze eingeben:
5

Schueler ueber 15 und Note unter
5 (sortiert nach Note):

Name: Alterschueler
Vorname: Christina
Alter: 25
Note: 4

------------------------------------------------

Name: Schueler
Vorname: Anton
Alter: 18
Note: 3

------------------------------------------------
```

```
Name: Musterschueler
Vorname: Berta
Alter: 17
Note: 1

-----------------------------------------------

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 8

Arithmetisches Mittel der Noten: 3

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 9

Geometrisches Mittel der Noten: 2.6051712

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 2
Name: Schueler
Vorname: Anton
Alter: 18
Note: 3

-----------------------------------------------

Name: Musterschueler
Vorname: Berta
Alter: 17
Note: 1

-----------------------------------------------

Name: Alterschueler
Vorname: Christina
Alter: 25
Note: 4

-----------------------------------------------

Name: Jungerschueler
Vorname: Daniel
Alter: 15
Note: 2

-----------------------------------------------

Name: Schlechterschueler
Vorname: Emil
Alter: 20
Note: 5
```

```
------------------------------------------------

************************* Menue *************************
...
************************* Ende Menue *************************

Wahl: 10
Programmende!
```