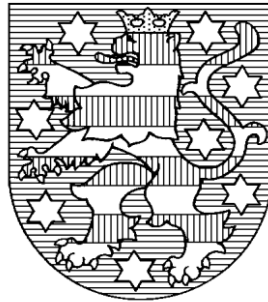


Thüringer Kultusministerium



Abiturprüfung 1997

Informatik

als Leistungsfach
(Haupttermin)

Arbeitszeit: 240 Minuten

Einlesezeit: 30 Minuten

Hilfsmittel: Formeln und Tabellen für die Sekundarstufen I und II:
Gymnasien, Gesamtschulen, Berufsschulen.
paetec Ges. für Bildung und Technik Berlin;
PC mit Oberon- und Prolog-System;
Taschenrechner

Der Prüfungsteilnehmer löst **alle** Aufgaben.

Rechts unten neben jeder Teilaufgabe steht die für diese Teilaufgabe maximal erreichbare Anzahl von Bewertungseinheiten (BE).

Der Prüfungsteilnehmer sichert bei der praktischen Arbeit am Computer mindestens alle 10 Minuten die von ihm erarbeiteten Oberon-Programme und Prolog-Programme.

Der Prüfungsteilnehmer hat die von ihm erarbeiteten Quelltexte abzugeben.

1 Abstrakte Datentypen

1.1 Erläutern Sie im Zusammenhang mit abstrakten Datentypen die folgenden Begriffe:

- Modularität,
- Geheimnisprinzip,
- Geschütztheit.

6 BE

1.2 Sie sollen ein Team leiten, das aus mehreren Mitgliedern besteht und das sich das Ziel gesetzt hat, ein größeres Softwareprodukt zu entwickeln. Man einigt sich, hierzu abstrakte Datentypen zu schaffen und zu verwenden.

Beschreiben Sie, welche Arbeitsschritte auszuführen sind!
Welche Absprachen sind dazu zu treffen?

4 BE

1.3 Es stehen zu Ihrer Verfügung:

- im Anhang: die verbindliche Spezifikation des abstrakten Datentyps "Liste" (ADT Liste) und
- auf dem PC: zwei Oberon-Quelltexte (`adtliste.Mod` und `test.Mod`)

Die Datei `adtliste.Mod` stellt eine Realisation des ADT Liste dar (bis auf die Operationen `AnhaengenElement` und `GeheLetztes`). Das Modul `test` demonstriert die Nutzung des Moduls `adtliste`.

- a) Gegeben ist die Prozedur `Struktur`, die wie die Prozedur `LoeschenElement` aus dem Modul `adtliste` aufgebaut ist:

```

PROCEDURE Struktur(VAR liste: pliste);
BEGIN
  IF ~ ListeLeer(liste) THEN
    IF liste = letztes THEN
      (* Dies ist Fall 1. *)
    ELSIF ListenAnfang(liste) THEN
      (* Dies ist Fall 2. *)
    ELSIF ListenEnde(liste) THEN
      (* Dies ist Fall 3. *)
    ELSE
      (* Dies ist Fall 4. *)
    END
  END
END Struktur;

```

Geben Sie für jeden der in der Prozedur angegebenen vier Fälle an:

- Aus wie vielen Elementen kann die Liste bestehen?
- Welche Elemente können aktuelles Element sein?

4 BE

- b) Realisieren Sie die Operationen `AnhaengenElement` und `GeheLetztes` !

Entwerfen und implementieren Sie dazu jeweils eine **Oberon-Prozedur!**

Kommentieren Sie die Prozeduren im Quelltext!

Fügen Sie die zwei Prozeduren in das Modul `adtliste` ein!

8 BE

1.4 Geben Sie eine Spezifikation für den ADT Stapel an!

6 BE

1.5 Realisieren Sie den ADT Stapel!

Beachten Sie die folgenden Festlegungen:

- Als Realisation ist ein Oberon-Modul `adtstapel` anzugeben.
- Es ist das Modul `adtliste` zu verwenden.
- Für die Realisation ist Ihre Spezifikation maßgebend.
- Kommentieren Sie Ihr Modul im Quelltext!

6 BE

1.6 Entwerfen und implementieren Sie ein **Oberon-Modul**, das ein gegebenes Wort zeichenweise umdreht und ausgibt.

Beispiel: Aus `INFORMATION` wird `NOITAMROFNI` .

Beachten Sie die folgenden Festlegungen:

- Das gegebene Wort ist zeichenweise einzulesen.
- Die Ausgabe des umgedrehten Wortes erfolgt in einem Textviewer.
- Es ist das Modul `adtstapel` zu verwenden.
- Ihr Modul erhält den Namen `wort` (Datei `wort.Mod`)!
- Kommentieren Sie Ihr Modul im Quelltext!

6 BE

2 Symbolisches Differenzieren

- 2.1 Entwerfen und implementieren Sie ein **Prolog-Programm**, das die 1. Ableitung einer gegebenen Funktion ermittelt und ausgibt!

Gegeben sind die folgenden Funktionen:

$$y = a \quad (\text{Konstante})$$

$$y = x$$

$$y = x^n \quad (n \text{ ist eine positive ganze Zahl})$$

$$y = \sin x$$

$$y = \tan x$$

$$y = e^x$$

$$y = \ln x$$

Diese Funktionen können durch die Operatoren + und * verknüpft werden. Es können auch verkettete Funktionen gebildet werden.

Von Ihnen sind die folgenden Differentiationsregeln in dem Prolog-Programm umzusetzen:

die Faktorregel,
die Summenregel,
die Produktregel und
die Kettenregel für die oben genannten Funktionen.

Geben Sie zu Ihrem Prolog-Programm für vier verschiedene Funktionen je eine Abfrage zum Ermitteln der 1. Ableitung an!

Beachten Sie die folgenden Festlegungen:

- Ihr Prolog-Programm erhält den Namen diff.pro !
- Kommentieren Sie Ihr Prolog-Programm im Quelltext!

15 BE

- 2.2 Formulieren Sie zu dem Prolog-Programm eine Abfrage zum Ermitteln der 1. Ableitung der Funktion $y = x^2 \sin e^x$!

Geben Sie die Antwort an, die das Prolog-System auf Ihre Abfrage liefert!

Begründen Sie, warum das Prolog-System die von Ihnen genannte Antwort liefert!

5 BE

Anhang:**Spezifikation des abstrakten Datentyps "Liste" (ADT Liste)****(A) Wertebereich:**

Eine (einfach verkettete) Liste besteht aus Elementen, die in linearer Folge angeordnet sind. Eine solche Liste kann nur von vorn nach hinten durchlaufen werden.

Ist die Liste nicht leer, dann gibt es das **erste Element**, das **letzte Element** und ein **aktuelles Element**. Enthält eine Liste genau ein Element, dann ist dieses Element sowohl aktuelles, erstes als auch letztes Element.

Für jedes Element einer Liste, mit Ausnahme des letzten Elementes, gibt es das **nächste Element**.

(B) Operationen:**1. Operation: Erzeugen**

Eine leere Liste wird angelegt.

Diese Operation ist Voraussetzung für alle anderen Operationen.

2. Operation: EinfuegenElement

Ein Element wird vor dem aktuellen Element eingefügt.

Wird diese Operation auf eine leere Liste angewandt, dann enthält die Liste anschließend genau ein Element.

Das neue Element wird zum aktuellen Element.

3. Operation: AnhaengenElement

Ein Element wird nach dem letzten Element angefügt.

Wird diese Operation auf eine leere Liste angewandt, dann enthält die Liste anschließend genau ein Element.

Das neue Element wird zum aktuellen Element.

4. Operation: LoeschenElement

Das aktuelle Element wird gelöscht.

Das erste Element wird dann zum aktuellen Element.

Ist die Liste leer, dann passiert nichts.

Enthält die Liste genau ein Element, dann wird dieses gelöscht; dadurch entsteht eine leere Liste.

5. Operation: GeheErstes

Das erste Element wird zum aktuellen Element.

Ist die Liste leer, dann passiert nichts.

6. Operation: GeheLetztes

Das letzte Element wird zum aktuellen Element.

Ist die Liste leer, dann passiert nichts.

7. Operation: GeheNaechstes

Das nächste Element wird zum aktuellen Element.

Diese Operation darf nicht angewandt werden, wenn das letzte Element aktuelles Element ist.

Ist die Liste leer, dann passiert nichts.

8. Operation: HoleEintrag

Der Inhalt des aktuellen Elementes wird bereitgestellt.

Das aktuelle Element bleibt aktuelles Element.

Diese Operation darf nicht angewandt werden, wenn die Liste leer ist.

9. Operation: SchreibeEintrag

Der Inhalt des aktuellen Elementes wird überschrieben.

Das aktuelle Element bleibt aktuelles Element.

Diese Operation darf nicht angewandt werden, wenn die Liste leer ist.

10. Operation: ListeLeer

Diese Operation liefert den Wert TRUE, wenn die Liste leer ist.

Ansonsten liefert sie den Wert FALSE.

11. Operation: ListenEnde

Diese Operation liefert den Wert TRUE, wenn das letzte Element aktuelles Element ist.

TRUE wird auch geliefert, wenn die Liste leer ist.

Ansonsten liefert die Operation den Wert FALSE.