

# Testklausur

Ausgabe: Di, 30.01.2001

Lösung in der  
Globalübung: Di, 13.02.2001

## Aufgabe 1 : Syntaxdiagramm und EBNF

Es sollen Bezeichner definiert werden, die folgenden Vorschriften genügen:

- Bezeichner bestehen aus Buchstaben, Ziffern und Punkten.
- Jeder Bezeichner beginnt mit einem Buchstaben.
- Die Bezeichner sind beliebig lang, aber nicht kürzer als zwei Zeichen
- Bezeichner dürfen durch Punkte gegliedert werden, jedoch dürfen diese weder am Ende noch mehrfach hintereinander stehen (d.h. „ABC.“ Und „AB..C“ sind falsch).

Die Produktionen für **Buchstabe** und **Ziffer** können als gegeben betrachtet werden.

**1.1** Geben Sie die Syntax für Bezeichner durch ein Syntaxdiagramm an!

Die Lösung muß knapp und eindeutig sein.

**1.2** Wie verändert sich die Lösung, wenn zusätzlich gelten soll, daß ein Bezeichner höchstens eine Ziffer enthalten darf?

Formulieren Sie die veränderte Syntaxbeschreibung in EBNF!

## Aufgabe 2: Kacheln

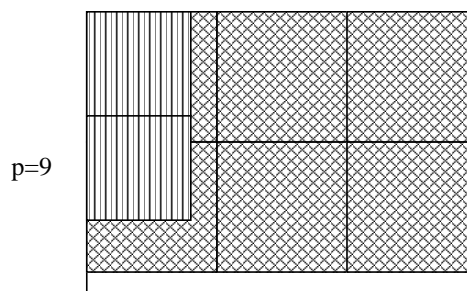
Bei der folgenden Aufgabe sind alle Längenmaße Vielfache der Grundeinheit  $E = 10$  cm. Die Gewichte (präzise: Massen) sind Vielfache der Masse einer kleinen Kachel (10 cm im Quadrat), die  $m = 100$  g wiegt. Die Kachel mit doppelter Kantenlänge ist entsprechend viermal so schwer usw.

Es stehen quadratische Kacheln aller Größenstufen bis zur Kantenlänge  $k_{\max} \cdot E$  zur Verfügung, also mit den Kantenlängen  $E, 2E, \dots, k_{\max} \cdot E$ .

Ein Rechteck der Länge  $p \cdot E$  und der Breite  $q \cdot E$  ( $p, q \in \mathbb{N}$ ) ist mit den größten verfügbaren und in das Rechteck passenden Kacheln zu belegen, wobei diese nicht über das Rechteck hinausragen dürfen. Sie bilden nun ein neues Rechteck, das mit den nächst kleineren Kacheln ausgelegt wird usw., bis schließlich in der obersten Schicht Kacheln der Größe  $E$  liegen.

Für ein gegebenes Tripel  $p, q, k_{\max}$  ist das Gesamtgewicht der Kacheln zu berechnen, die auf dem Rechteck liegen. Beispiel:  $p = 9, q = 12, k_{\max} = 4$ .

$q=12$



Die Abbildung zeigt, daß  $2 \cdot 3 = 6$  Kacheln der Größe 4 auf das Rechteck passen, darauf  $2 \cdot 4 = 8$  Kacheln der Größe 3 (von denen zwei links gezeigt sind). Auf diese wiederum kommen  $3 \cdot 6$  der Größe 2 und  $6 \cdot 12$  der Größe 1. Das ergibt ein Gesamtgewicht von  $(6 \cdot 16 + 8 \cdot 9 + 18 \cdot 4 + 72 \cdot 1) \cdot 100 \text{ g} = 31\,200 \text{ g}$ .

**2.1** Geben Sie eine Modula-3 Funktion an, die aus den Parametern p, q, und kmax das Gewicht (in g) der Kacheln nach obigem Berechnungsschema liefert.

### Aufgabe 3: Programmanalyse

Gegeben sei die folgende rekursive Prozedur:

```
PROCEDURE Unknown (p1 : INTEGER; p2 : BOOLEAN) =
BEGIN
  IF NOT p2 THEN
    SIO.PutInt (p1 DIV 7);
    SIO.Nl();
  END;
  IF p2 OR (p1 > 10) THEN
    Unknown ( (p1 + 33) MOD 41, p2 # (p1 < 20) )
  END;
END Unknown;
```

Die Prozedur erzeugt also mit jeder Inkarnation keine oder eine Zeile Ausgabe.

**3.1** Geben Sie nach dem Muster des Beispiels (Aufruf Unknown (10, TRUE)) die Inkarnationen von Unknown an, die aus dem Aufruf Unknown (30, FALSE) entstehen, und zwar bis zum Abbruch der Rekursion oder bis zur letzten Zeile in der dafür vorgesehenen Tabelle.

#	p1	p2	Ausgabe (falls erzeugt)
1	10	TRUE	-
2	2	FALSE	0 Abbruch

#	p1	p2	Ausgabe (falls erzeugt)
1			
2			
3			
4			
5			
6			
7			

## Aufgabe 4: Listen

Seien  $x = (x_1, x_2, \dots, x_m)$  und  $y = (y_1, y_2, \dots, y_n)$  zwei nicht-leere einfach verkettete Listen.

Aus diesen Listen soll eine neue Liste  $Z$  konstruiert werden, die folgenden Bedingungen genügt.

$$\begin{aligned} Z &= (x_1, y_1, x_2, y_2, \dots, x_m, y_m, y_{m+1}, \dots, y_n) && \text{falls } n > m \\ Z &= (x_1, y_1, x_2, y_2, \dots, x_n, y_n, x_{n+1}, \dots, x_m) && \text{falls } m > n \\ Z &= (x_1, y_1, x_2, y_2, \dots, x_m, y_n) && \text{falls } n = m \end{aligned}$$

- 4.1** Entwerfen Sie eine geeignete Datenstruktur und schreiben Sie eine Modula-3 Prozedur `ErzeugeZ`, die aus zwei nicht-leeren einfach verketteten Listen  $x$  und  $y$  eine neue Liste  $z$  gemäß obiger Spezifikation erzeugt (die beiden Eingabe-Listen werden dabei zerstört)

## Aufgabe 5: Abstrakter Datentyp

Die Realisierung von Mengen in Modula-3 ist beschränkt auf Ordinaltypen als Elementtyp der Menge. Sollen andere Elementtypen in Mengen verwendet werden, so muß dafür eine geeignete Implementierung erstellt werden.

Entwerfen Sie einen Abstrakten Datentyp `SetOfText`, der eine Menge realisiert, deren Elementtyp der vordefinierte Typ `TEXT` ist. Als Operationen sollen bereitgestellt werden: Vereinigung, Schnitt, Aufnahme und Entfernen eines Elements aus der Menge sowie der Test des Enthaltenseins.

- 5.1** Geben Sie das Interface-Modul für den ADT `SetOfText` an!

- 5.2** Geben Sie den Deklarationsteil des Implementierungs-Moduls des ADTs `SetOfText` an (d.h. alles bis zur ersten implementierten Funktion).  
Dabei sollen Mengen prinzipiell unendlich viele Elemente aufnehmen können.

- 5.3** Implementieren Sie die Funktion der Mengenvereinigung des ADTs `SetOfText` gemäß Ihrer Deklaration im Interface-Modul!

## Aufgabe 6: Klassenhierarchie und Objekttypen

Sie haben die Aufgabe, ein Programm zu entwickeln, mit dem Literaturreferenzen verwaltet werden sollen. Dabei stellen Sie fest, daß es die folgenden vier verschiedenen Arten von Literaturreferenzen gibt.

*Referenz auf ein Buch* : diese ist charakterisiert durch Angabe von Autor, Titel, Erscheinungsjahr und Verlag

*Referenz auf einen Technischen Bericht*: diese ist charakterisiert durch Angabe von Autor, Titel, Erscheinungsjahr, Name der Institution und Nummer des Berichts.

*Referenz auf einen Beitrag in einem Buch*: diese ist charakterisiert durch Angabe von Autor, Titel des Beitrags, Erscheinungsjahr, Buchtitel und Herausgeber.

*Referenz auf einen Beitrag in einer Zeitschrift*: diese ist charakterisiert durch Angabe von Autor, Titel des Beitrags, Erscheinungsjahr, Zeitschriftentitel und Ausgabennummer.

**6.1** Entwerfen Sie eine geeignete Klassenhierarchie, um die oben aufgelisteten Arten an Literaturreferenzen zu realisieren.

Achten Sie dabei darauf, daß gemeinsame Merkmale in abstrakten Klassen zusammengezogen werden. Notieren Sie Ihren Entwurf grafisch und verwenden Sie dazu die UML-Notation (geben Sie lediglich pro Klasse den Namen der Klassen und die Bezeichner der Exemplarvariablen an).

**6.2** Geben Sie die Deklaration der Objekttypen (keine geschützten Objekttypen) für die Klassen `ReferenzAufBuch` und `ReferenzAufBeitragInZeitschrift` an. Dabei soll berücksichtigt werden, daß alle Objekte der Klassenhierarchie die Nachrichten `initialisiere` (initialisiert alle Merkmale einer Literaturreferenz mit Standard-Werten) und `alsText` (liefert einen TEXT zurück, der die textuelle Repräsentation aller Merkmale einer Literaturreferenz enthält) kennen müssen. Geben Sie für die beiden Klassen (Objekttypen) auch die notwendigen Zugriffsmethoden für deren spezielle Exemplarvariablen an.

*Hinweis*: Verwenden Sie der Einfachheit halber ausschließlich TEXT als Typ für die Exemplarvariablen der Klassen.

**6.3** Implementieren Sie die zur Nachricht `alsText` gehörende Prozedur der Klasse `ReferenzAufBeitragInZeitschrift`.

## Aufgabe 7: Lisp

In dieser Aufgabe seien Mengen von Zahlen identisch mit aufsteigend geordneten Listen von Zahlen.

Z. B. entspricht hierbei die Menge  $\{8, 3, 6, 2\}$  der Liste  $(2, 3, 6, 8)$  und die Menge  $\{9, 7, 1, 4, 3\}$  der Liste  $(1, 3, 4, 7, 9)$ . Die Vereinigung der beiden Mengen entspricht der Liste  $(1, 2, 3, 4, 6, 7, 8, 9)$ .

**7.1** Implementieren Sie in LISP eine Funktion `vereinigungs-menge`, die die Vereinigung zweier solcher Mengen berechnet.

**7.2** Schätzen Sie den zeitlichen Aufwand Ihrer Funktion für die Berechnung der Vereinigungsmenge ab.

Ist der Aufwand Ihrer Funktion nicht in  $O(n)$ , so überlegen Sie, wie sich eine Funktion schreiben läßt, die die Vereinigung in der Zeit  $O(n)$  berechnet.