

Name:			
Vorname:			
Matrikelnummer:			
Punkte:	/24 =	%	Note:
FP erstmalig:			1./2. Wiederholung:

Klausur im SS 96
zur
Informatik II
am 14. Juni 1996

Hinweise:

- Bitte jedes Blatt einzeln mit Ihrem Namen und der Nummer der bearbeiteten Aufgabe bezeichnen.
- Es sind **beliebige schriftliche Unterlagen** zugelassen, jedoch **keine Rechner**, auch keine Taschenrechner. Klausurabgabetermin ist 12³⁰ Uhr.
- Sie können farbige Stifte verwenden, jedoch **kein Rot**.
- Maßgeblich für die Beantwortung der Fragen ist die Sprachbeschreibung von Oberon-2 auf der Basis der Vorlesung und des Buches: M. Reiser, N. Wirth *Programmieren in Oberon. Das neue Pascal*.

Aufgabe 1

(Punktzahl: 4/24)

- a) Zeichnen Sie den Vererbungsgraphen der Datenstruktur in dem nachfolgend abgedruckten Modul mit den üblichen Konventionen: speziellere Klassen nach unten, gerichtete Kante von der spezielleren zur allgemeineren Klasse. Verwenden Sie zur Bezeichnung der Klassen in dem Graphen die Namen der Zeigertypen (also `Obj0`, ...).
- b) Welches ist die Ausgabe nach einer Kommandoaktivierung `KlausurAufgabe.Start`?

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

```
MODULE KlausurAufgabe;  
  IMPORT Out;  
  TYPE  
    Obj0 = POINTER TO Obj0Desc;  
    Obj0Desc = RECORD END;  
  
    Obj1 = POINTER TO Obj1Desc;  
    Obj1Desc = RECORD ( Obj0Desc ) END;  
  
    Obj2 = POINTER TO Obj2Desc;  
    Obj2Desc = RECORD (Obj1Desc) END;  
  
    Obj3 = POINTER TO Obj3Desc;  
    Obj3Desc = RECORD (Obj2Desc) END;  
  
    Obj4 = POINTER TO Obj4Desc;  
    Obj4Desc = RECORD (Obj2Desc) END;  
  
    Obj5 = POINTER TO Obj5Desc;  
    Obj5Desc = RECORD (Obj1Desc) END;  
  
    Obj6 = POINTER TO Obj6Desc;  
    Obj6Desc = RECORD (Obj1Desc) END;  
  
  PROCEDURE (p: Obj0) P;  
  BEGIN Out.String("0") END P;
```

```
  PROCEDURE (p: Obj1) P;  
  BEGIN Out.String("1") END P;  
  
  PROCEDURE (p: Obj3) P;  
  BEGIN Out.String("3") END P;  
  
  PROCEDURE (p: Obj6) P;  
  BEGIN Out.String("6") END P;  
  
  PROCEDURE (p:Obj2) Q;  
  BEGIN p.P END Q;  
  
  PROCEDURE Start*;  
  VAR  
    p0: Obj0; p1: Obj1; p2: Obj2; p3: Obj3;  
    p4: Obj4; p5: Obj5; p6: Obj6;  
  BEGIN  
    NEW(p0); NEW(p1); NEW(p2); NEW(p3);  
    NEW(p4); NEW(p5); NEW(p6);  
    p0.P; p1.P; p2.P; p3.P; p4.P; p5.P; p6.P;  
    p2.Q;  
  END Start;  
  
END KlausurAufgabe.
```

Die Datenstruktur zu einem binären Suchbaum sei folgendermaßen gegeben:

```
TYPE
  Node = POINTER TO NodeDesc;
  NodeDesc = RECORD
    key: INTEGER;
    left, right: Node
  END;
  Tree = RECORD
    root: Node
  END;
```

Schreiben Sie eine Prozedur **ShowTree**, die diesen Baum traversiert und die Werte des Record-Feldes **key** aller Elemente der Größe nach geordnet ausdruckt. **ShowTree** soll eine an **Tree** typgebundene Prozedur sein.

Gegeben sei das folgende Modul mit einigen Typdefinitionen.

```
MODULE KlausurAufgabe;

TYPE
  P = POINTER TO Integer;
  Integer = RECORD Integer: INTEGER END;      (* 1 *)

  A1 = POINTER TO ARRAY OF REAL;              (* 2 *)
  A3 = POINTER TO ARRAY -3, 3 OF REAL;        (* 3 *)

  Q = POINTER TO RECORD
    A: ARRAY 5 OF REAL
  END;                                         (* 4 *)

  Record = RECORD c: CHAR END;
  R = POINTER TO RECORD (Record) END;         (* 5 *)

  ObjDesc = RECORD next: Obj END;
  Obj = POINTER TO ObjDesc;                  (* 6 *)

  IntObject = POINTER TO IntObjectDesc;
  IntObjectDesc = RECORD i: INTEGER END;
  Object = POINTER TO ObjectDesc;
  ObjectDesc = RECORD (IntObject) END;       (* 7 *)

  CharObject = RECORD c: CHAR END;
  Function = PROCEDURE(c: CHAR): CharObject; (* 8 *)

END KlausurAufgabe.
```

Überprüfen Sie, welche dieser Typdefinitionen richtig sind und welche nicht. Tragen Sie Ihr Ergebnis (**R**ichtig/**F**alsch) in die folgende Tabelle ein. Die Nummern beziehen sich jeweils auf die zusammenstehenden Zeilen.

1	2	3	4	5	6	7	8

Sie sollen im Auftrag eines Softwarehauses ein System zur Verschlüsselung von Zeichenketten erstellen. Die mit dem Kunden vereinbarten Bedingungen sind die folgenden:

- Der Kunde besitzt bereits ein Modul (**User**), dessen Schnittstelle unten abgedruckt ist. Die Prozedur **Apply** liest einen String ein, ruft die Prozedur **Encrypt** aus **User** auf und gibt das Resultat aus. Dieses Modul *müssen* Sie in der vorliegenden Form verwenden. Sowohl Ihnen als auch dem Kunden steht nur die **Obj**-Datei von **User** zur Verfügung, nicht der Quellcode; dieser kann daher von Ihnen auch nicht geändert werden.
- Sie liefern ein Modul dazu (Name: **Encryption1**), in welchem die Verschlüsselung ausgeführt wird.

- Um die Verschlüsselungsvorschrift vertraulich zu halten liefern Sie Ihren Quellcode (den von **Encryption1**) nicht aus.
 - Der Kunde stellt die Bedingung, daß er selbst zu einem späteren Zeitpunkt Ihr Modul mit Namen **Encryption1** durch ein Modul **Encryption2** ersetzen kann, obwohl ihm der Quellcode weder von **Encryption1** noch von **User** zur Verfügung steht.
- a) Legen Sie zunächst die Schnittstelle von **Encryption1** fest (kurze Erläuterung). Der Name der Prozedur, welche die Verschlüsselung konkret durchführt, sei **Encrypt**.
 - b) Schreiben Sie das Modul **Encryption1** vollständig auf mit ausgeführtem Quelltext von **Encrypt**. Die Prozedur **Encrypt** soll hier der Einfachheit halber nur jedes ASCII-Zeichen zwischen A und z durch das jeweils nächstfolgende ersetzen, also A durch B, B durch C, ..., z durch {.
 - c) Welche Oberon-Kommandos müssen vom Benutzer nacheinander aufgerufen werden, um die Verschlüsselung auszuführen. Welche müssen aufgerufen werden, nachdem der Kunde sein Modul **Encryption2** geschrieben hat und anwenden will?

Schnittstelle von User:

```
DEFINITION User;
  TYPE
    String = ARRAY 1024 OF CHAR;
  VAR
    Encrypt: PROCEDURE (VAR s: String);
  PROCEDURE Apply;
END User.
```

Hinweis: Der entscheidende Teil der Aufgabe ist das Auffinden der richtigen Modulstruktur. Gesucht ist ausdrücklich nur der Quellcode des Moduls **Encryption1**.

Aufgabe 5

(Punktzahl: 4/24)

Unter dem Mittelwert einer Funktion $f : \mathbb{Z} \rightarrow \mathbb{R}$ über dem vorgegebenen Intervall $[a, b]$ versteht man

$$M_{a,b}[f] := \frac{1}{n} \sum_{i=a}^b f(i),$$

wobei n die Gesamtzahl der Werte in $[a, b]$ ist mit $a, b \in \mathbb{Z}$ und $\mathbb{Z}$ bzw. $\mathbb{R}$ die ganzen bzw. die reellen Zahlen bedeuten.

- a) Ergänzen Sie das abgedruckte Modul um die fehlende Typ-Definition und die Funktionsprozedur **MeanVal**, die $M_{a,b}[f]$ berechnet.
- b) Welcher Wert wird ausgegeben?

```
MODULE KlausurAufgabe;
```

```
  IMPORT Out;
  TYPE Function = ...
```

```
  PROCEDURE f(x: INTEGER): REAL;
  BEGIN RETURN (2 + x)/2.5;
  END f;
```

```
  PROCEDURE MeanVal(f: Function; a, b: INTEGER): REAL;
  ...
  END MeanVal;
```

```
  PROCEDURE Start*;
  BEGIN Out.Real(MeanVal(f, 1, 10), 12); Out.Ln
  END Start;
```

```
END KlausurAufgabe.
```