

Physikalisch-Technische Lehranstalt Wedel

Staatsexamen für Technische Assistenten WS2003/4 schriftliche Prüfung im Fach Prozesstechnik gewählter Vorschlag

Dauer : 180 Minuten

keine externen Hilfsmittel

Aufgabe :

Nachteilig an der klassischen Parameterübergabe beispielsweise zwischen Haupt- und Unterprogramm ist die starre Festlegung auf Anzahl und Typen der Parameter. Entwickeln Sie daher auf der Basis des 8086-Prozessors einen simplen Prototypen für eine dynamische Parameterliste durch möglichst bedeutungstreue Übersetzung des Pascal-Programms *VorschlagZwei*.

Zur Realisierung der New-Funktion und der Dispose-Prozedur werden die DOS-Funktionen "Allocate Memory" (Funktionscode 48h) und "Free Memory" (Funktionscode 49h) verwendet. Die Funktion zum Reservieren von Speicherplatz erwartet im BX-Register die Anzahl der benötigten zusammenhängenden 16-Byte Blöcke und liefert als Ergebnis im AX-Register die Segmentadresse der reservierten Blöcke (als Offsetadresse innerhalb des Segments wird Null impliziert). Der Fall "nicht genügend zur Verfügung stehender Speicher" bleibt unberücksichtigt. Die Funktion zum Freigeben von Speicherplatz erwartet im ES-Register die Segmentadresse der freizugebenden Blöcke, also eine Adresse die von der Funktion "Allocate Memory" zurückgeliefert wurde.

Vergessen Sie nicht die hinreichende Kommentierung Ihres Assembler-Programms z.B. durch Zuordnung der Pascal-Befehle zu den Assembler-Befehlen.

Das PTL-Team wünscht viel Erfolg

Program VorschlagZwei;

```
Type pBoolean      = ^Boolean;
    pWord          = ^Word;
    pString        = ^String;
    tParameterArt  = (tBoolean,tWord,tString);
    tpParameter    = ^tParameter;
    tParameter     = Record
        Art        : tParameterArt;
        Inhalt     : Pointer;
        Naechster  : tpParameter
    End;
```

```
Var Parameterliste : tpParameter;
    Parameter       : tpParameter;
```

```
Procedure ListParameters(P:tpParameter);
```

```
Var B : Byte;
    W : Word;
    S : String;
```

```
Begin
    While P <> Nil Do Begin
        Case P^.Art Of
            tBoolean : If Boolean(P^.Inhalt^) Then
                S := 'Alaaf'
            Else
                S := 'Helau';
            tWord      : Begin
                W := Word(P^.Inhalt^);
                S := '';
                While W <> 0 Do Begin
                    S := Char(W Mod 10 + Byte('0')) + S;
                    W := W Div 10
                End
            End;
            tString    : S := String(P^.Inhalt^)
        End;
        For B := 1 To Byte(S[0]) Do
            Write(S[B]);
        Write(#13#10);
        P := P^.Naechster
    End
End;
```

```

Procedure DestroyParameters(P:tpParameter);

Var ltpParameter : tpParameter;
    lpBoolean    : pBoolean;
    lpWord       : pWord;
    lpString     : pString;

Begin
  While P <> Nil Do Begin
    ltpParameter := P;
    Case P^.Art Of
      tBoolean : Begin
        lpBoolean := P^.Inhalt;
        Dispose(lpBoolean)
      End;
      tWord    : Begin
        lpWord := P^.Inhalt;
        Dispose(lpWord)
      End;
      tString  : Begin
        lpString := P^.Inhalt;
        Dispose(lpString)
      End
    End;
    P := P^.Naechster;
    dispose(ltpParameter)
  End
End;

Begin
  New(Parameterliste);
  Parameter := Parameterliste;

  Parameter^.Art := tBoolean;
  Parameter^.Inhalt := New(pBoolean);
  Boolean(Parameter^.Inhalt^) := True;
  Parameter^.Naechster := New(tpParameter);
  Parameter := Parameter^.Naechster;

  Parameter^.Art := tWord;
  Parameter^.Inhalt := New(pWord);
  Word(Parameter^.Inhalt^) := 4711;
  Parameter^.Naechster := New(tpParameter);
  Parameter := Parameter^.Naechster;

  Parameter^.Art := tString;
  Parameter^.Inhalt := New(pString);
  String(Parameter^.Inhalt^) := 'Koeln am Rhein';
  Parameter^.Naechster := Nil;

  ListParameters(Parameterliste);

  DestroyParameters(Parameterliste)
End.

```