

7.2.1 Strukturen – Struct

Die folgenden Zeilen beschreiben die Syntax für die Deklaration einer Struktur:

```
PUBLIC STRUCT Identifier
  Field_1 [ Embedded array declaration ] AS [ Datatype ]
  Field_2 [ Embedded array declaration ] AS [ Datatype ]
  ...
  Field_n [ Embedded array declaration ] AS [ Datatype ]
END STRUCT
```

7.2.1.1 Deklaration Struct

Diese Deklaration definiert die Struktur *Schueler* mit 5 Feldern:

```
Public Struct Schueler
  Jahrgangsstufe As Integer
  GebDatum As Date
  Nachname As String
  DG1Kurs As String
  DG2Kurs As String
End Struct
```

Beachten Sie, dass Sie bei der Deklaration einer Struktur als Zugriffsrecht nur Public setzen können. Die Felder von *Jahrgangsstufe* bis *DG2Kurs* sind p.d. Variablen mit dem Zugriffsrecht Public.

7.2.1.2 Arrays von Strukturen

Die Besonderheit der folgenden Deklaration einer Struktur *Kurs* besteht darin, dass in der Struktur *Schueler* das Feld *Kursliste[8]* ein Array mit Elementen vom Typ *Struct* aus der Kurs-Deklaration ist:

```
Public Struct Kurs
  KursFach As String
  Kurslehrer As String
End Struct

Public Struct Schueler
  Jahrgangsstufe As Integer
  GebDatum As Date
  Nachname As String
  Kursliste[8] As Struct Kurs
End Struct
```

Arrays mit Elementen vom Typ *Struct* sind keine echten Arrays, denn sie verfügen nur über sehr wenige Methoden der originalen Array-Klasse:

- Auf die Elemente kann über einen Index zugegriffen werden.
- Es kann über alle Elemente iteriert werden (Aufzählung).
- Informationen über die Länge des Arrays und die Array-Dimensionen kann man abfragen.

Anschließend können Sie Variablen vom benutzerdefinierten (Pseudo-)Typ *Struct* deklarieren. Zuerst eine Variable *KursTeilnehmer* vom Typ *Struct* und anschließend ein Array *Kursliste[22]*, in dem die relevanten Daten von 22 Schülern gespeichert werden und dessen Elemente vom Typ *Struct* sind:

```
Public KursTeilnehmer As Struct Schueler
Public Schueler As New Schueler ' Das funktioniert alternativ auch ...

Public Kursliste[22] As Struct Schueler
Public Kursliste[22] As New Schueler ' → Fehlermeldung
```

7.2.1.3 Zugriff auf Strukturen und deren Felder

Nach der Deklaration von Strukturen erfahren Sie, wie man auf die im vorangegangenen Abschnitt deklarierte Struktur und deren Felder lesend und schreibend über Variablen vom Typ *Struct* zugreift. Für die Eingabe der Werte der einzelnen Felder und deren Anzeige wird die folgende Programmoberfläche genutzt:

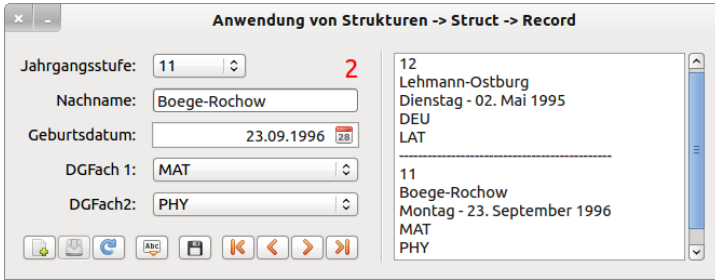


Abbildung 7.2.1.3.1: Programmoberfläche

- Zuerst wird der Zugriff auf die Eingabemaske freigegeben.
- Dann werden allen 5 Feldern der Variablen *KursTeilnehmer* vom Typ Struct (*Schueler*) Werte zugewiesen oder die Vorgabewerte überschrieben.
- Abschließend wird der aktuelle Inhalt der Variablen *KursTeilnehmer* in das Array *Kursliste[22]* eingefügt.

```
Public Sub SaveNewRecord()
    KursTeilnehmer.Jahrgangsstufe = cmbJGS.Text
    KursTeilnehmer.Nachname = txbNachname.Text
    KursTeilnehmer.GebDatum = dbGebDatum.Value
    KursTeilnehmer.DG1Kurs = cmbDGF1.Text
    KursTeilnehmer.DG2Kurs = cmbDGF2.Text
    ' Neuen Record im Array Kursliste[] speichern
    KursListe[iRecordNumber] = KursTeilnehmer
End ' SaveNewRecord()
```

Mit Hilfe der With-Anweisung können alternativ die Zuweisungen vereinfacht geschrieben werden. Der Quelltext ist besser lesbar:

```
Public Sub SaveNewRecord()

    With KursTeilnehmer
        .Jahrgangsstufe = cmbJGS.Text
        .Nachname = txbNachname.Text
        .GebDatum = dbGebDatum.Value
        .DG1Kurs = cmbDGF1.Text
        .DG2Kurs = cmbDGF2.Text
    End With

    KursListe[iRecordNumber] = KursTeilnehmer

End ' SaveNewRecord()
```

Eine kompakte Alternative bietet sich an, wenn Sie den Inhalt eines Records sofort in die das Array *Kursliste[22]* einfügen:

```
Public Sub SaveNewRecord()
    KursListe[iRecordNumber].Jahrgangsstufe = cmbJGS.Text
    KursListe[iRecordNumber].Nachname = txbNachname.Text
    KursListe[iRecordNumber].GebDatum = dbGebDatum.Value
    KursListe[iRecordNumber].DG1Kurs = cmbDGF1.Text
    KursListe[iRecordNumber].DG2Kurs = cmbDGF2.Text
End ' SaveNewRecord()
```

Die folgenden zwei Prozeduren demonstrieren, wie man auf die Inhalte der einzelnen Felder einer Struktur lesend zugreift:

```
Public Sub ShowCurrentRecord()
    cmbJGS.Text = KursListe[iCurrentRecord].Jahrgangsstufe
    txbNachname.Text = KursListe[iCurrentRecord].Nachname
    dbGebDatum.Value = KursListe[iCurrentRecord].GebDatum
    cmbDGF1.Text = KursListe[iCurrentRecord].DG1Kurs
    cmbDGF2.Text = KursListe[iCurrentRecord].DG2Kurs
End ' ShowCurrentRecord()
```

```
Public Sub ShowRecords()
    Dim iCount As Integer

    TextAreal.Clear
```

```
For iCount = 0 To iRecordNumber
    TextAreal.Insert(KursListe[iCount].Jahrgangsstufe & gb.NewLine)
    TextAreal.Insert(KursListe[iCount].Nachname & gb.NewLine)
    TextAreal.Insert(Format(KursListe[iCount].GebDatum, "dddd - dd. mmmmm yyyy") & gb.NewLine)
    TextAreal.Insert(KursListe[iCount].DG1Kurs & gb.NewLine)
    TextAreal.Insert(KursListe[iCount].DG2Kurs & gb.NewLine)
    If iCount < iRecordNumber Then TextAreal.Insert("-----" & gb.NewLine)
Next ' iCount
TextAreal.Pos = Len(TextAreal.Text) ' ---> Sprung in die letzte Zeile

End ' ShowRecords()
```