7.2.2 Projekt mit Strukturen

Im folgenden Abschnitt wird der vollständige Quelltext für ein Projekt angegeben, das eine Struktur und ein Array mit Elementen vom Typ *Struct* nutzt. Wenn Sie das Projekt an eine Datenbank-Anwendung erinnert, dann sind Sie auf der richtigen Spur.

7.2.2.1 Projektbeschreibung

Einige Besonderheiten des Projekts müssen erwähnt werden:

- Sie können nur maximal 22 Datensätze erfassen und deren *Anzahl* zur Laufzeit des Programms auch nicht ändern! Die Begriffe Struct, Record und Datensatz werden hier synonym verwendet.
- Durch die Verwendung von ComboBoxen und einer DateBox werden Eingabefehler minimiert, sofern die Listen für die drei ComboBoxen mit Sorgfalt gefüllt worden sind. Für den einzugebenden Nachnamen existiert im Projekt keine Datenprüfung!
- Sie können durch die einzelnen Datensätze – als Elemente des Arrays KursListe[22] – navigieren und sie anzeigen.
- Einzelne Datensätze können Sie überschreiben, jedoch nicht löschen.
- Wenn Sie das Programm beenden, werden alle Daten – die sich ja nur im Speicher befinden – gelöscht.
- Für das Projekt ist ein Speicher-Management implementiert worden, um den Inhalt des Arrays Kursliste[22] in einer Datei zu speichern (Daten-Export) oder mit den gespeicherten Daten das Array Kursliste[22] zu restaurieren.
- Die Anzeige aller Datensätze in einer *TextArea* diente nur zu Kontrollzwecken beim Projekt-Test.

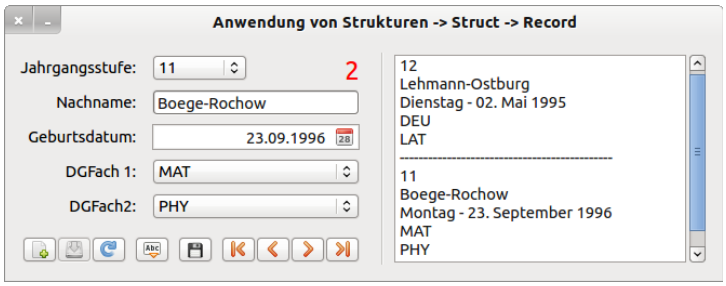

Abbildung 7.2.2.1.1: Programmoberfläche

7.2.2.2 Projekt-Quelltext:

```
' Gambas class file

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

Public KursTeilnehmer As New Schueler
Public Const MAX_SCHUELER As Integer = 22
Public KursListe[22] As Struct Schueler

Public iRecordNumber As Integer = -1
Public iCurrentRecord As Integer

Public Sub Form_Open()
  FMain.Center
  FMain.Resizable = False
  cmbDGF1.Text = cmbDGF1[0].Text
  cmbDGF2.Text = cmbDGF2[1].Text
  tlblRecords.Foreground = Color.Red
  tlblRecords.Visible = False
  Status(False)
```

```
  btnUpdate.Enabled = False
  btnSave.Enabled = False
  txbNachname.SetFocus
End ' Form_Open()

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()

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

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
  tlblRecords.Text = iCurrentRecord + 1
End ' ShowCurrentRecord()

'::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

Public Sub btnNew_Click()
  btnUpdate.Enabled = False
  cmbJGS.Text = cmbJGS[0].Text
  txbNachname.Clear
  dbGebDatum.Value = "6/1/1995"
  cmbDGF1.Text = cmbDGF1[0].Text
  cmbDGF2.Text = cmbDGF2[1].Text
  btnSave.Enabled = True
  Status(True)
  txbNachname.SetFocus
End ' btnNew_Click()

Public Sub btnSave_Click()
  If txbNachname.Text = Null Then
     Message.Warning("Der Nachname muss eingegeben werden!")
     txbNachname.SetFocus
     Return
  Endif ' txbNachname.Text = Null?
  Inc iRecordNumber
  iCurrentRecord = iRecordNumber
  SaveNewRecord()
  ShowRecords()
  tlblRecords.Visible = True
  tlblRecords.Text = iRecordNumber + 1
  If iRecordNumber = MAX_SCHUELER - 1 Then
     Message.Info("Achtung!\nDie KursTeilnehmer-Liste ist voll.")
     btnSave.Enabled = False
     btnNew.Enabled = False
     btnUpdate.Enabled = True
     Return
  Endif ' iRecordNumber = MAX_SCHUELER-1 ?
  btnSave.Enabled = False
  btnUpdate.Enabled = True
  btnExport.Enabled = True
End ' btnSave_Click()

Public Sub btnBack_Click()
  If iCurrentRecord > 0 Then
```

```
    Dec iCurrentRecord
    ShowCurrentRecord()
  Endif ' iCurrentRecord > 0 ?
End ' btnBack_Click()

Public Sub btnForward_Click()
  If iCurrentRecord < iRecordNumber Then
    Inc iCurrentRecord
    ShowCurrentRecord()
  Endif ' iCurrentRecord < iRecordNumber ?
End ' btnBack_Click()

Public Sub btnFirst_Click()
  If iRecordNumber > 0 Then
    iCurrentRecord = 0
    ShowCurrentRecord()
  Endif ' iRecordNumber > 0 ?
End ' btnFirst_Click()

Public Sub btnLast_Click()
  If iRecordNumber > 0 Then
    iCurrentRecord = iRecordNumber
    ShowCurrentRecord()
  Endif ' iRecordNumber > 0 ?
End ' btnLast_Click()

Public Sub btnUpdate_Click()
  UpdateCurrentRecord()
  ShowRecords()
  btnExport.Enabled = True
End ' btnUpdate_Click()

Public Sub Status(iStatus As Boolean)
  If iStatus = True Then
    cmbJGS.Enabled = True
    txbNachname.Enabled = True
    dbGebDatum.Enabled = True
    cmbDGF1.Enabled = True
    cmbDGF2.Enabled = True
  Else
    cmbJGS.Enabled = False
    txbNachname.Enabled = False
    dbGebDatum.Enabled = False
    cmbDGF1.Enabled = False
    cmbDGF2.Enabled = False
    btnExport.Enabled = False
  Endif
End ' Status(iStatus As Boolean)

Public Sub ShowRecords()
  Dim iCount As Integer

  TextArea1.Clear
  For iCount = 0 To iRecordNumber
   TextArea1.Insert(KursListe[iCount].Jahrgangsstufe & gb.NewLine)
   TextArea1.Insert(KursListe[iCount].Nachname & gb.NewLine)
   TextArea1.Insert(Format(KursListe[iCount].GebDatum, "dddd - dd. mmmm yyyy") & gb.NewLine)
   TextArea1.Insert(KursListe[iCount].DG1Kurs & gb.NewLine)
   TextArea1.Insert(KursListe[iCount].DG2Kurs & gb.NewLine)
   If iCount < iRecordNumber Then TextArea1.Insert("--------------------------" & gb.NewLine)
  Next ' iCount
  TextArea1.Pos = Len(TextArea1.Text) ' ---> Sprung in die letzte Zeile

End ' ShowRecords()

Public Sub btnExport_Click()
  Dim hFile As File
  Dim iCount As Integer

  Dialog.Title = "Exportieren Sie Datensätze aus einer StructDatenBasis-Datei!"
  Dialog.Filter = ["*.sdb", "StructDatenBasis-Dateien"]

  If Dialog.SaveFile() Then Return
  hFile = Open Dialog.Path For Write Create
```

```
  For iCount = 0 To iRecordNumber
      Write #hFile, KursListe[iCount] As Schueler
  Next ' iCount
  Close #hFile

  Catch
    Message.Error(Error.Text)
End ' btnExport_Click()

Public Sub btnImport_Click()
  Dim hFile As File
  Dim iCount As Integer

  Dialog.Title = "Importieren Sie Datensätze aus einer StructDatenBasis-Datei!"
  Dialog.Filter = ["*.sdb", "StructDatenBasis-Dateien"]

  If Dialog.OpenFile(False) Then Return
  hFile = Open Dialog.Path For Read
  iCount = 0
  While Not Eof(hFile)
    KursListe[iCount] = Read #hFile As Schueler
    Inc iCount
  Wend
  Close #hFile

  iRecordNumber = iCount - 1
  Print iRecordNumber
  If iCount Then
     Status(True)
     ShowRecords()
     tlblRecords.Visible = True
     btnUpdate.Enabled = True
     iCurrentRecord = 0
     ShowRecords()
     ShowCurrentRecord()
  Else
     Message.Info("Die Import-Datei leer!")
     Return
  Endif ' iIndexCount?

  Catch
    Message.Error("Der Daten-Import war fehlerhaft!" & gb.Newline & gb.NewLine & "Fehler:   "
                  & Error.Text)
End ' btnImport_Click()
```

7.2.2.3 Daten-Export und Daten-Import

Das Projekt verfügt über ein Speicher-Management, mit denen Sie eine Struktur in einer Datei abspeichern können (Daten-Export) oder den Inhalt einer Export-Datei in eine Struktur einlesen können (Daten-Import). Es werden alle eingetragenen Datensätze über WRITE serialisiert in den Datei-Stream geschrieben, denn Gambas unterstützt die "*Serialisierung*" von Variablen, wenn diese in einen Stream geschrieben werden. Gambas formatiert die Daten so, dass sie später aus der Datei wieder hergestellt werden können (Daten-Import). Die Daten in der Export-Datei sind jedoch nur für Gambas-Programme lesbar, weil sie gambas-spezifisch formatiert sind. Die Extension der Export-Datei ist frei wählbar; gut geeignet wären *.txt* oder *.dat* oder *.sdb* (StructDatenBasis). Die Reihenfolge der Daten in der Export-Datei entspricht der Deklaration der Struktur, wie der folgende Ausschnitt aus einer Export-Datei zeigt (Jahrgangsstufe, Geburtsdatum, Nachname, Kurs1, Kurs2):

```
Byte (hex)  0B 00 00 00 36 DF 25   00 00 00 00 00 05 41 61 64 6D 03 42 49 4F 03 43 48 45
Klartext:   11 .  .  .  01 06 1995 .  .  .  .  .  .  A  d  a  m  .  B  I  O  .  C  H  E
Konvertierung Datum:   CFloat(Date(1995, 06, 01, 0, 0, 0)) = 2881974 (dez) = 25DF36 (hex)
```