

17.5.2 Projekt 2

Das Projekt 2 zum Kapitel 17.5 ColumnView demonstriert die Verwendung von Eigenschaften, Methoden und Ereignissen des Steuerelementes ColumnView sowie das Speicher-Management für den Datenexport und Datenimport.

- Beim ersten Programmstart kommen die Daten der ColumnView aus einer DB-Tabelle. Interessant ist in diesem Zusammenhang die statische Klasse DBC (DataBaseConnection). Die Klasse DBC basiert auf einer Idee von Christof Thalhofer und wird im → Kapitel 22. Datenbanken beschrieben.
- Die Speicherung der Daten in der ColumnView erfolgt nach dem *ersten* Schließen des Programmfensters in einer Text-Datei.
- Als Daten-Format wurde das JSON-Format gewählt. Andere Möglichkeiten wären u.a. CSV-Dateien, Settings-Dateien oder XML-Dateien.

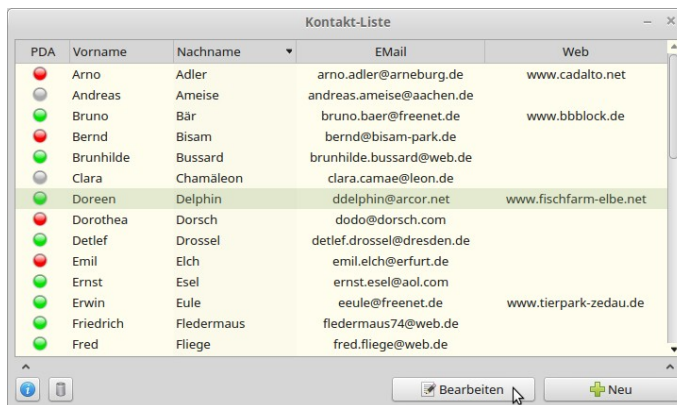


Abbildung 17.5.2.1: Hauptprogramm mit der GridView

- Eine Settings-Datei (/home/username/.config/gambas3/Contacts.conf) speichert in einem Wert Imported=1 im Slot [ImportDBData] das Umschalten von einmaligem Einlesen der Daten aus einer DB-Tabelle auf Daten-Import/Daten-Export über die Text-Datei DataBase/json.data im Projekt-Verzeichnis.
- Das Bild in der ersten Spalte symbolisiert den Status eines Kontakts (privat (grün), dienstlich (rot) und andere Kontakte (grau)).
- Die Kontakt-Daten werden beim Programmstart stets nach der 3. Spalte 'Nachname' sortiert. Alle weiteren Spalten können ebenso sortiert werden. Beachten Sie: Eine Sortierung nach Symbolen in der 1. Spalte hat keinen Effekt, da nur nach dem Spalten-Text sortiert wird und den Symbolen kein Text zugeordnet wurde.
- Sie können neue Kontakte anlegen, bestehende Kontakt-Daten ändern oder einen Kontakt löschen. Die Eingabe eines Nachnamens ist obligatorisch. Die Eingaben in den Eingabefeldern werden syntaktisch *nicht* geprüft.

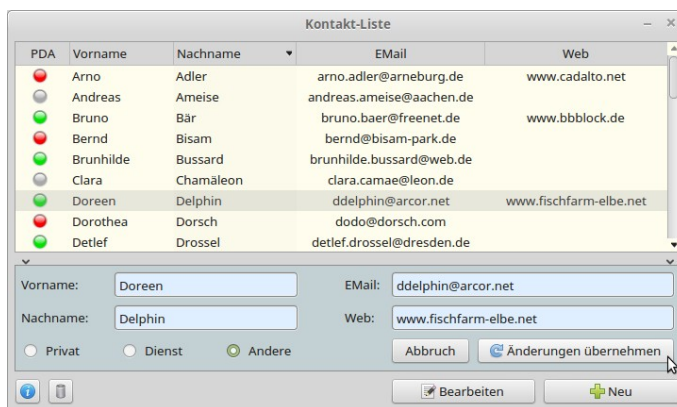


Abbildung 17.5.2.2: Einen Kontakt ändern

Hinweise zu Prüfungen auf valide Daten finden Sie in den → Kapiteln 16.6.1 TextBox und ihre Spezialisierungen, 16.6.2 Valide Daten und 19.6.5 Prüfung der Syntax von Zeichenketten.

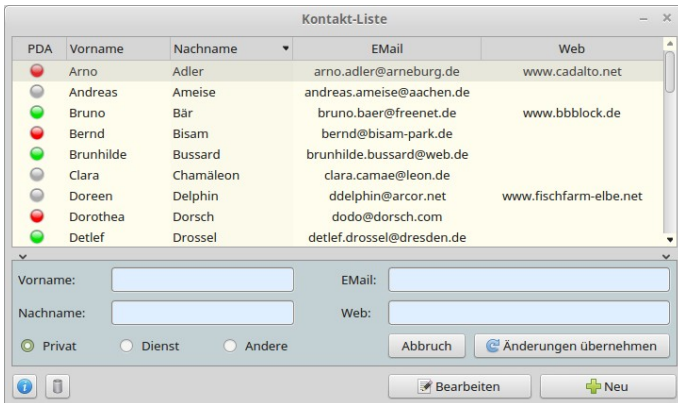


Abbildung 17.5.2.3: Einen neuen Kontakt einfügen

Der Quelltext wird in ausgewählten Auszügen angegeben, welche die Iteration über alle Elemente einer ColumnView zeigen, die durch unterschiedliche Kontrollstrukturen realisiert werden:

```
[1] Private Sub SetDBData()
[2]
[3]     Dim rDBResult As Result
[4]     Dim iKey As Integer
[5]     Dim sDBTableName, sSQL_Command, sKey As String
[6]     Dim hItem As _TreeView_Item
[7]
[8]     sDBTableName = "kontakte" ' Name of the DB
[9]     sSQL_Command = Subst("&1 &2&3", "SELECT vorname, nachname, email, web, pbo FROM", sDBTableName, ";")
[10]
[11]     rDBResult = DBC.DBConnection.Exec(sSQL_Command)
[12]
[13]     If rDBResult.Count = 0 Then
[14]         Message.Warning(Subst("&1 &2 &3", "<h1>", ("The number of selected records is zero!"), "</h1>"))
[15]         Return
[16]     Endif
[17]
[18]     If rDBResult.Available Then
[19]         For Each rDBResult
[20]             Inc iKey ' Der erste Key ist "1", weil eine Kopfzeile definiert ist
[21]             sKey = Str(iKey)
[22]             Select rDBResult["pbo"]
[23]                 Case "p"
[24]                     covData.Add(sKey, "", Picture["green16.png"]) ' 1. Spalte
[25]                 Case "b"
[26]                     covData.Add(sKey, "", Picture["red16.png"])
[27]                 Case "o"
[28]                     covData.Add(sKey, "", Picture["gray16.png"])
[29]             End Select
[30]             ' Weitere ColumnView-Spalten einfügen ...
[31]             covData[sKey][1] = rDBResult["vorname"]
[32]             covData[sKey][2] = rDBResult["nachname"]
[33]             covData[sKey][3] = rDBResult["email"]
[34]             covData[sKey][4] = rDBResult["web"]
[35]         Next
[36]     Endif
[37]
[38]     ' If rDBResult.Available Then
[39]     '     For Each rDBResult
[40]     '         Inc iKey
[41]     '         sKey = Str(iKey)
[42]     '         Select rDBResult["pbo"]
[43]     '             Case "p"
[44]     '                 hItem = covData.Add(sKey, "", Picture["green16.png"])
[45]     '             Case "b"
[46]     '                 hItem = covData.Add(sKey, "", Picture["red16.png"])
[47]     '             Case "o"
[48]     '                 hItem = covData.Add(sKey, "", Picture["gray16.png"])
[49]     '             End Select
[50]     '         ' Weitere ColumnView-Spalten einfügen ...
[51]     '         hItem[1] = rDBResult["vorname"]
[52]     '         hItem[2] = rDBResult["nachname"]
[53]     '         hItem[3] = rDBResult["email"]
[54]     '         hItem[4] = rDBResult["web"]
[55]     '     Next
[56]     ' Endif
[57]
[58]     CVSettings["ImportDBData/Imported"] = 1
```

```
[59] CVSettings.Save()
[60]
[61] Catch
[62] Message.Error(Error.Text)
[63]
[64] End
```

Hinweise:

- In der Zeile 11 wird die in der (statischen) Klasse DBC definierte DB-Verbindung eingesetzt und das Ergebnis der DB-Abfrage in der Variablen rDBResult gespeichert.
- Das Auslesen der ausgewählten Felder der DB-Tabelle 'kontakte' erfolgt in den Zeilen 18 bis 36.
- Ein Alternative finden Sie in den Zeilen 38 bis 56.
- In der Zeile 58 wird der Wert in der Settings-Datei im Slot 'ImportDBData' von 0 auf 1 geändert und in der Zeile 59 abgespeichert. Mit dem Ändern dieses Wertes werden alle exportierten Daten aus der ColumnView – gespeichert in einer Datei im JSON-Format – zukünftig wieder aus dieser Datei in die ColumnView eingelesen.

In den nächsten beiden Absätzen werden Ihnen die Export- und Import-Prozedur vorgestellt:

Daten-Export:

```
[1] Private Sub ExportData(sJSONPath As String)
[2]
[3] Dim i As Integer
[4] Dim cJSONCollection As JSONCollection
[5] Dim cJSON As JSONCollection
[6]
[7] cJSON = New JSONCollection ' Ein neues JSONCollection-Objekt erzeugen
[8] Repeat
[9] cJSONCollection = New JSONCollection
[10] cJSONCollection["pbo"] = covData.Item[0]
[11]
[12] Select covData.Item.Picture ' Bild in der 1. Spalte (Index=0)
[13] Case Picture["red16.png"]
[14] cJSONCollection["picture"] = "red16.png"
[15] Case Picture["green16.png"]
[16] cJSONCollection["picture"] = "green16.png"
[17] Case Picture["gray16.png"]
[18] cJSONCollection["picture"] = "gray16.png"
[19] End Select
[20]
[21] cJSONCollection["vorname"] = covData.Item[1] ' Text in der 2. Spalte (Index=1)
[22] cJSONCollection["nachname"] = covData.Item[2]
[23] cJSONCollection["email"] = covData.Item[3]
[24] cJSONCollection["web"] = covData.Item[4]
[25] Inc i
[26] ' In die JSONCollection 'cJSON' eine JSONCollection als Element einfügen
[27] cJSON["RECORD " & Str(i)] = cJSONCollection
[28] Until covData.MoveBelow() ' ... das letzte Element in der ColumnView erreicht ist
[29]
[30] File.Save(sJSONPath & "json.data", JSON.Encode(cJSON)) ' Den JSONText in einer Datei speichern
[31]
[32] End
```

Daten-Import:

```
[1] Private Sub ImportData(sJSONPath As String, hColumnView As ColumnView)
[2]
[3] Dim iElement As Integer = 1
[4] Dim cJSONCollection As JSONCollection
[5] Dim cCollection As JSONCollection
[6]
[7] If Not Exist(sJSONPath) Then Return
[8] hColumnView.Clear()
[9] cJSONCollection = JSON.Decode(File.Load(sJSONPath), True)
[10]
[11] For Each cCollection In cJSONCollection
[12] hColumnView.Add(Str(iElement), cCollection["pbo"], Picture[cCollection["picture"]])
[13] hColumnView[iElement][1] = cCollection["vorname"]
[14] hColumnView[iElement][2] = cCollection["nachname"]
[15] hColumnView[iElement][3] = cCollection["email"]
[16] hColumnView[iElement][4] = cCollection["web"]
[17] Inc iElement
[18] Next
[19] End
```