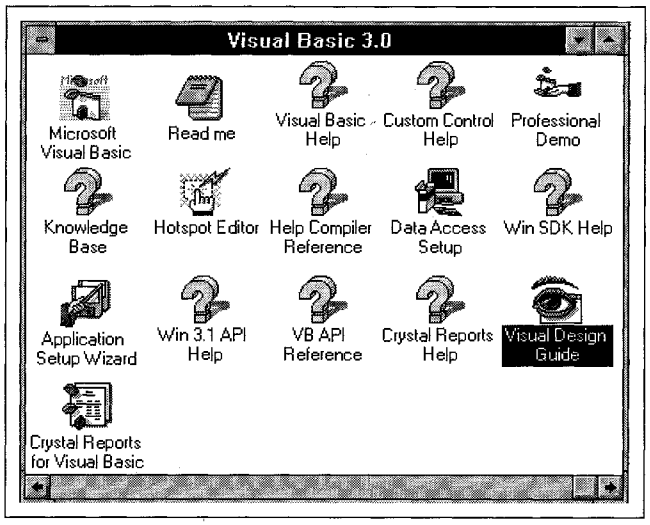


Programmieren für Juristen Visual Basic – Lektion IV

Maximilian Herberger

Arbeiten mit Index-Dateien

Rekapitulieren wir kurz: Nachdem in Folge II gezeigt wurde, wie man mit Hilfe eines SQL-Befehls einen Filter über eine Datenbank legen kann, war in Folge III wegen des Zeitaspekts dieses Verfahrens eine weniger zeitkritische Suchmethode vorgestellt worden. Indessen wird beim Erproben dieser Methode dem kritischen Leser nicht verborgen geblieben sein, daß es sich dabei um einen sequentiellen Zugriff handelt, der in großen Datenbeständen ebenfalls an seine Grenzen stößt. Man muß deswegen, will man hier zu einer Beschleunigung kommen, mit Index-Dateien arbeiten. Dies wird – immer noch bezogen auf dBASE-Dateien – Gegenstand des heutigen Kurses sein. Vorab muß man sich aber darüber klar sein, daß so zwar die Suche beschleunigt werden kann. Nichtsdestotrotz taucht das Zeitproblem an einer anderen Stelle auf, beim Einfügen neuer Datensätze nämlich. Wenn hier in großen Datenmengen alle Indizes aktualisiert werden müssen, dauert auch das seine Zeit. In der EDV (und vielleicht nicht nur dort) "bezahlt" man immer an irgendeiner Stelle für das, was man an anderer Stelle gewonnen hat.

Die neue Aufgabe: Anzeige in unterschiedlicher Reihenfolge

Dieses vorausgeschickt sei noch eine unangenehme Wahrheit ins Auge gefaßt: Beim Arbeiten mit Index-Dateien müssen wir uns von der Data-Control verabschieden. Das läßt sich aber verschmerzen, da sich der zusätzliche Programmieraufwand in Grenzen hält. Weil es sich dementsprechend um einen Neuanfang handelt, verlassen wir das bisherige Formular und konstruieren ein neues. Es soll im ersten Anlauf die Sätze unserer Leitsatzdatenbank nach Wahl des Benutzers in unterschiedlicher Index-Reihenfolge anzeigen.

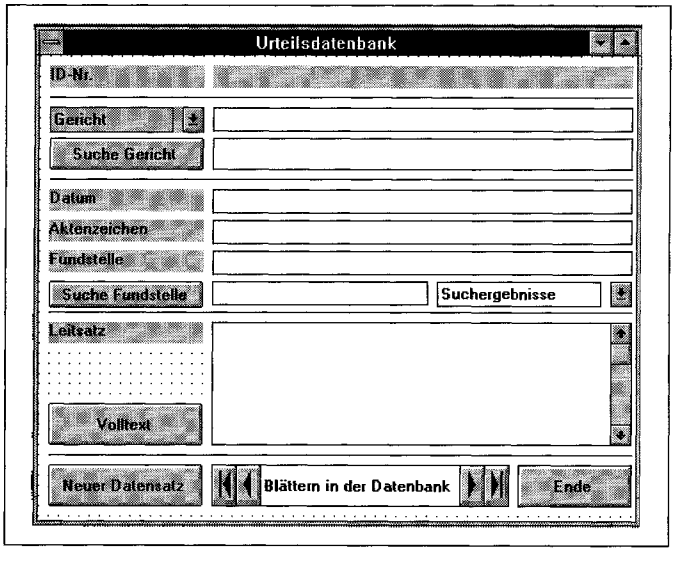


Abb. 1: Formular mit umsortierbarer Liste

JurPC-Archiv: http://www.jurpc.de

Unser neues Formular hat in der ersten Ausbaustufe das aus Abb. 1 ersichtliche Aussehen. Als Objekte im Formular wurden verwendet:
Die Liste, drei "Radio-Buttons" und ein Rahmen, der die Gruppe der "Radio-Buttons" zusammenhält. In der Liste soll eine Kurzinformation zu den Urteilen erscheinen, ein "Click" auf die Buttons soll die Reihenfolge nach dem jeweiligen Index umordnen.

Schritt Nr. 1: Die .INF-Datei
Um auf dBASE-Indizes zugreifen zu können, muß eine Datei angelegt werden, die wie die Datenbank-Datei heißt, allerdings statt .DBF die Extension .INF hat. Darin werden die zu verwendenden Indexdateien angegeben. In unserem Fall sieht der Inhalt dieser Datei wie folgt aus:

```
[dbase]
NDX1=lfdnr.ndx
NDX2=gericht.ndx
NDX3=datum.ndx
```

Damit stehen nun die Indizes für die laufende Nummer (unsere Identifizierung), das Gericht und das Datum zur Verfügung.

Schritt Nr. 2: Die Initialisierung der Datenbank
Da wir auf die Data-Control verzichten müssen, sind die folgenden Zeilen zur Initialisierung der Datenbank (beim Laden des Formulars in der Prozedur Form_Load) erforderlich:

```
pat$ = "C:\jur-pc\vb\04"
Set database = OpenDatabase(pat$, False, False, "Dbase III;")
Set table1 = database.OpenTable("urteile")
table1.Index = "lfdnr"
```

Hier wird erstens der Pfad für die Datenbank angegeben, danach wird zweitens die Datenbank-Variable mit Voreinstellungen gesetzt, drittens wird die Datenbank als Tabelle mit Namen *table1* geöffnet und schließlich wird viertens der Index nach laufender Nummer ausgewählt.

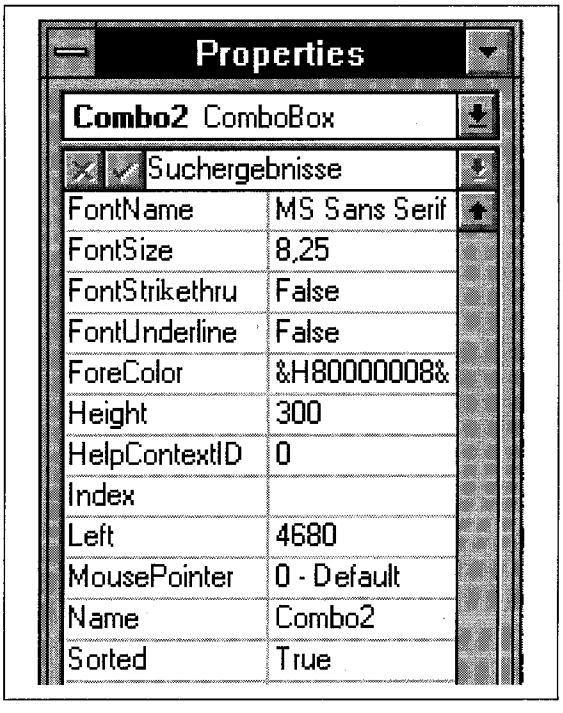
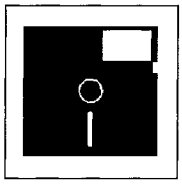


Abb. 2: "Definitionsfenster" für eine eigene Prozedur

Schritt Nr. 3: Das Auffüllen der Liste
Es gilt nun, die Liste gemäß dem eben eingestellten Index aufzufüllen. Da dieser Vorgang verschiedentlich anfallen wird (etwa bei jedem Umsortieren nach anderem Index), verlagern wir diesen Vorgang in eine eigene Prozedur.
Um eine eigene Prozedur schreiben zu können, muß man zunächst durch Doppelclick auf das Formular das Code-Fenster öffnen. Danach wählt man unter dem Menü-Punkt "View" die Zeile "New Procedure" aus. Es erscheint das Definitionsfenster für die neue Prozedur (*vgl. Abb. 2*).
Da wir eine Prozedur schreiben wollen, belassen wir es bei der Voreinstellung "Sub" und vergeben einen Namen, sagen wir "Liste_auffrischen".
In die dann angebotene leere Prozedur fügen wir folgenden Code ein:

```
Sub Liste_Auffrischen ()
list2.Clear 'Liste wird gelöscht
table1.MoveFirst 'Man geht zum ersten Datensatz
Do While Not table1.EOF
   eintrag = Str(table1("lfdnr")) + "= " \\weiter in derselben Zeile
   + table1("gericht") \\weiter in derselben Zeile
```


JurPC-Archiv: http://www.jurpc.de




```
   + " vom " + Str(tabel("datum")) + ", Az.: " + tabel("az")
   list2.AddItem eintrag
   tabel.MoveNext 'Man geht zum nächsten Datensatz
Loop
tabel.MoveFirst 'und wieder zurück zum ersten Datensatz
End Sub
```

Zentral ist dabei die "Do While"-Schleife. Sie wird solange durchlaufen, bis das Ende der Datenbank erreicht ist (EOF = End of File). Bei jedem einzelnen Datensatz wird dann ein Eintrag zusammengestellt, der anschließend mit dem Befehl "AddItem" in die Liste eingefügt wird. Da der Eintrag insgesamt ein String sein muß, gilt es, die laufende Nummer (die eine numerische Variable ist) und das Datum (das eine Datumsvariable ist) in einen String umzuwandeln. Das besorgt die Funktion Str, die einen String zurückliefert. Bei dieser langen Zeile ist zu beachten, daß alles in einer Zeile stehen muß. Darauf soll der Zusatz "\\weiter in derselben Zeile" hinweisen, der nicht zum Code gehört. Man sieht an dieser Stelle auch, wie man auf Felder der deklarierten Tabelle zugreifen kann: *table1("gericht")* etwa liefert für den aktuellen Datensatz den Inhalt des Feldes "Gericht" usw.
Wir müssen nun der Prozedur "Form_Load" nur noch die Zeile

```
Liste_Auffrischen
```

hinzufügen, um bei einem ersten Programmstart das Formular wie in Abb. 3 zu sehen.
Bleibt noch übrig, den "Radio-Buttons" den Befehl zum Index-Wechsel zu hinterlegen,

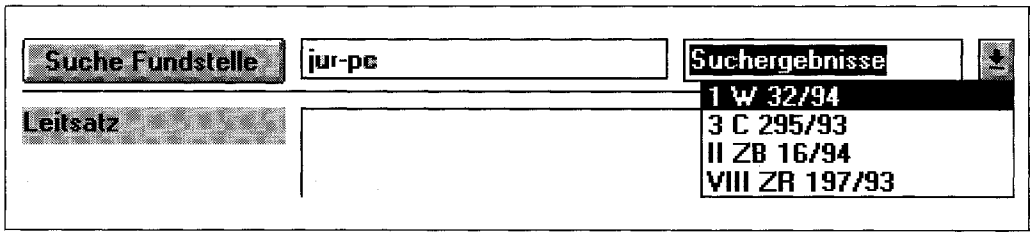


Abb. 3:
"Gefüllte" Liste nach Programmstart

was vergleichsweise einfach ist und für den Fall des Gerichtsindexes so aussieht:

```
tabel.Index = "Gericht"
Liste_Auffrischen
```

Beide Befehle kennen wir schon: Der erste benennt den neuen Index, der zweite frischt in der beschriebenen Weise die Liste auf.

Mit der Urteilsliste arbeiten
Die Urteilsliste dient der übersichtlichen Bewegung in der Datenbank durch "Scrollen". Beim gewünschten Datensatz angekommen will man dann natürlich zusätzliche Angaben sehen können. Wahrscheinlich wird der Benutzer erwarten, diese weiteren Informationen durch Click auf den aktuellen Datensatz aufrufen zu können. Um die diesbezügliche Programmierung zu zeigen, wird das Formular zunächst um ein Textfeld erweitert, in dem der Leitsatz erscheinen soll *(vgl. Abb. 4)*. Die Lösung benötigt schon die indexgestützte Suche, wie wir sehen werden.
Auf das Ereignis "Click" für die Liste wird folgender Code gelegt:

```
gleich = InStr(list2.Text, "=")
ziel = Val(Mid$(list2.Text, 1, gleich - 1))
tabel.Index = "lfdnr"
tabel.Seek "=", ziel
text1.Text = tabel("leitsatz")
```

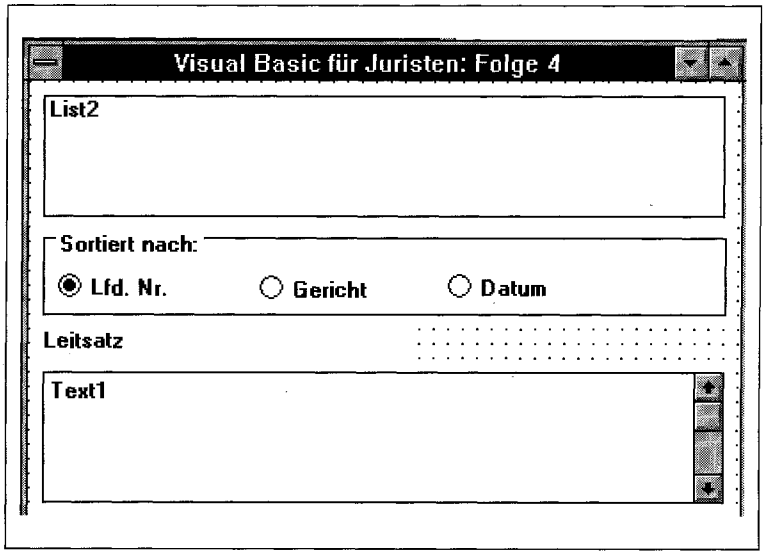


Abb. 4:
Formular mit Textfeld für den Leitsatz

JurPC-Archiv: http://www.jurpc.de

Die erste Zeile ermittelt die Position des Gleichheitszeichens in der Zeile, dies deswegen, weil vor dem Gleichheitszeichen die laufende Nummer (der Datensatz-Identifikator) steht. Unter Verwendung dieser Information wird dann in der Folgezeile in der Variablen *ziel* die laufende Nummer gespeichert. Dies geschieht, indem zunächst der Teil der aktuellen Zeile *list2.text* extrahiert wird, der vor dem Gleichheitszeichen steht. Das tut der Operator *mid$*, der von der Position 1 an bis zu einer Stelle vor dem Gleichheitszeichen den entsprechenden Stringbestandteil herauszieht. Da es sich dann um einen String handelt, wir aber eine Zahl brauchen (das Feld ist numerisch), besorgt der Operator *val* noch die Umwandlung in eine Zahl. Der Rest ist einfach: Es wird der Index auf "lfdnr" umgestellt und nach dem Datensatz mit der zugehörigen laufenden Nummer gesucht. Dies bewirkt die Zeile

```
table1.Seek "=", ziel
```

Erläuterungsbedürftig ist hier das in Anführungszeichen stehende Gleichheitszeichen. Es bedeutet, daß die Suchoperation nur bei exakter Übereinstimmung erfolgreich ist. An dieser Stelle könnten auch stehen:

< für "kleiner"
<= für "kleiner gleich"
>= für "größer gleich"
> für "größer"
<> für "verschieden.

dBASE-Programmierer müssen sich sicher erst daran gewöhnen, daß ein Treffer bei "=" nur erzielt wird, wenn der gesamte Feldinhalt 'gematcht' wird. Eine links beginnende teilweise Übereinstimmung zwischen Feld und Suchanfrage reicht nicht aus.
Auf diese Weise ist es jetzt möglich, beim Blättern durch die Liste Leitsätze aufzurufen *(vgl. Abb. 5)*.

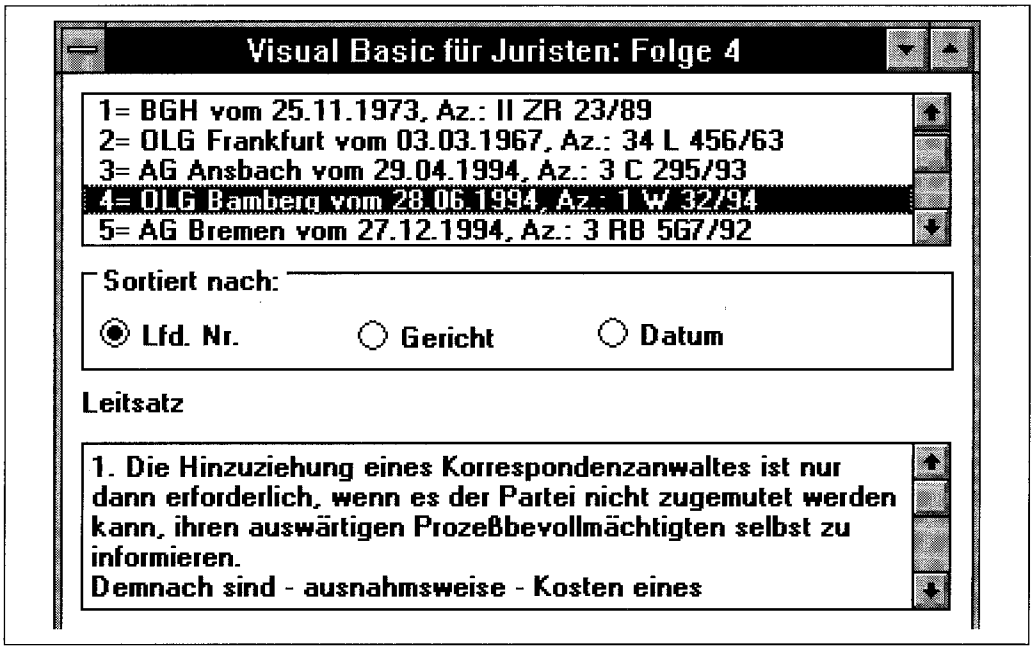
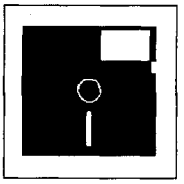


Abb. 5: Leitsatz-Aufrufaus der Liste

Aufbau einer indexgestützten Ergebnisliste
In großen Datenbeständen wird man nicht durch die lange Liste aller Datensätze 'scrollen' wollen. Vielmehr wird es wünschenswert sein, auf Grund einer Suchanfrage dynamisch eine Ergebnisliste zu erhalten. Wie man das erreichen kann, soll im folgenden am Beispiel einer Gerichtssuche gezeigt werden.
Zuerst wird wieder das Formular erweitert *(vgl. Abb. 6, auf der nächsten Seite)*.
Hinzugekommen ist ein Textfeld, in das man den Namen des gesuchten Gerichts einträgt. Ausgelöst wird die Suche durch den entsprechenden Befehlsbutton, hinter dem sich der entscheidende Programmcode verbirgt. Diesen Code gilt es jetzt zu kommentieren.

JurPC-Archiv: http://www.jurpc.de




```
list2.Clear
table1.Index = "gericht"
gericht = text2.Text
table1.Seek "=", gericht
If table1.NoMatch Then
   MsgBox "Kein Treffer!"
   Else
   Do While Not table1.EOF
      If table1("gericht")  gericht Then
       Exit Do
      else
       eintrag=Str(table1("lfdnr"))+"= "+table1("gericht")+" vom
       "+Str(table1("datum"))+", Az.: "+table1("az") \\alles eine Zeile
       list2.AddItem eintrag
       table1.MoveNext
      End If
   Loop
End If
```

Zuerst wird die Liste gelöscht, damit sich die Ergebnisliste nicht zu der bestehenden Liste hinzuaddiert. Sodann wird der für die Suche nötige Index gesetzt, die Sucheingabe aus dem Textfeld übernommen und die Suche ausgeführt.
Danach gilt es den Fall der fehlgeschlagenen Suche von dem der erfolgreichen Suche zu unterscheiden. Wenn die Suche fehlschlägt *(table1.nomatch)* wird in einer Messagebox eine diesbezügliche Mitteilung ausgegeben, und der Suchversuch ist zuende.
Ist die Suche erfolgreich, wird eine Schleife begonnen. Diese Schleife wird mit "exit do" verlassen, sobald keine Übereinstimmung mehr mit der Suchanfrage besteht. Ansonsten wird jeweils ein Eintrag der Liste hinzugefügt. (Zur Bildung dieses Eintrags vgl. man das oben Gesagte.) *table1.movenext* bewegt den Datensatzzeiger zum nächsten Datensatz, solange die Suchbedingung wahr bleibt.
Die Ergebnisliste für eine Suche nach "BGH" stellt sich dann wie aus Abb. 7 ersichtlich dar.

Ausblick
Der eigendiche "Härtetest" für eine Indexverwaltung besteht im Neueinfügen und im Löschen von Datensätzen. Wenn die Indexverwaltung in Ordnung ist, werden die Indizes beim Einfügen und beim Löschen aktualisiert. Dieser Erweiterung des vorliegenden Programms ist u.a. der folgende Teil gewidmet.

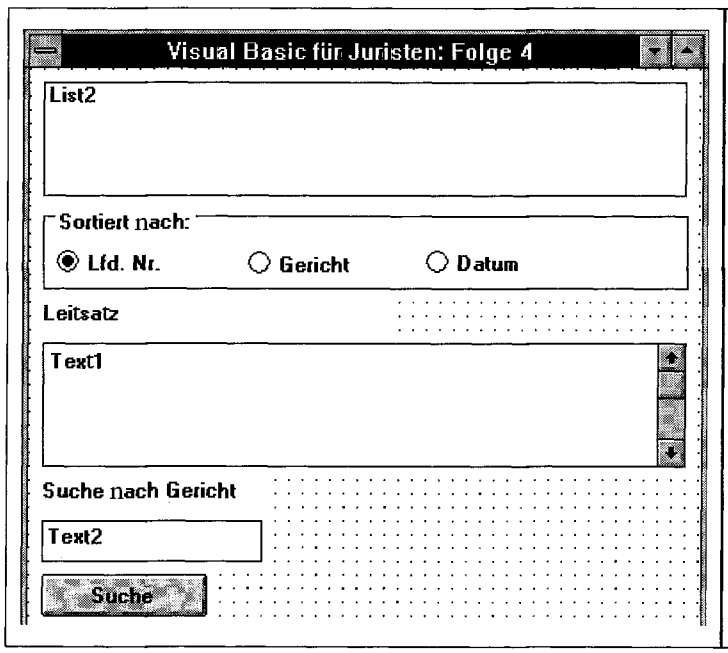


Abb. 6: Formularerweiterung für suchgestützte Ergebnisliste

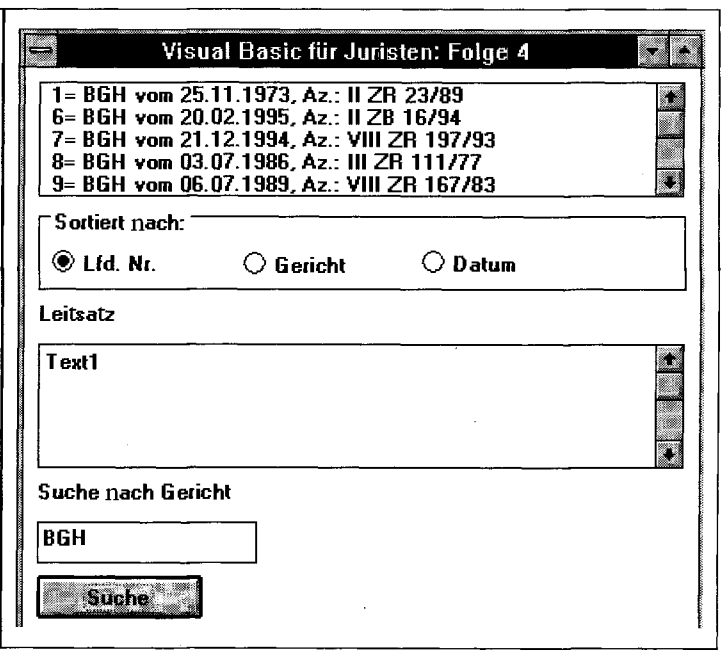


Abb. 7: Ergebnisliste für die Suche nach "BGH"

JurPC-Archiv: http://www.jurpc.de