

Zugang zum Objekt

Erfahrungen mit Microsoft ACCESS

Robert Suermann

Warum ACCESS?

Neugier und Skepsis kennzeichnen die erste Annäherung des eingefleischten dBASE-Clipper-Programmierers an eine WINDOWS-Datenbank.

Wieso überhaupt dieser Versuch? Lange war die Einstellung zu der grafischen Benutzeroberfläche WINDOWS eher ablehnend, bestenfalls ambivalent: einerseits alles ja ganz hübsch und nett und vielleicht auch für Anfänger hilfreich, andererseits für den altgedienten Computerianer ein Hindernis bei der ernsthaften Arbeit, dazu noch eins mit absurden Hardware-Anforderungen zum Hervorbringen banaler Ergebnisse.

Jedoch: alles fließt, und nirgends schneller als in der EDV-Szene. Und bei – zunächst notgedrungen – Beschäftigung mit WINDOWS zeigten sich doch Vorteile, die das Hantieren mit den Fenstern und Bildchen ("Icons") lohnend erscheinen ließen. Vor allem jene: die (einigermaßen) gleichartige Bedienung der verschiedensten WINDOWS-Programme und der Komfort eines (einigermaßen) funktionierenden Task-Swappings.

Nun also WINDOWS! Und als "Datenbanker" mithin eine WINDOWS-Datenbank. Aber welche? Erstaunlich lange war das Angebot an WINDOWS-Datenbanken gering, auch heute ist es noch nicht gerade üppig. Die vorhandenen sind überwiegend Anpassungen bewährter Systeme an die grafische Oberfläche. Microsoft brachte hingegen mit ACCESS (nach langer Entwicklungszeit) eine wirkliche Neuheit auf den Markt, die objektorientierte WINDOWS-Datenbank ACCESS. Setzen wir uns also an die Spitze des Fortschritts und wählen für ein anstehendes Personalverwaltungsprogramm für Referendare MS ACCESS.

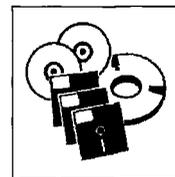
Das aber heißt – wie sich zeigen sollte – kräftig umdenken. Der Anfang eines Datenbankprojekts ist freilich auch hier rein gedankliche Arbeit: die Definition der Anforderungen, die Strukturierung der Datenbank, die zweckmäßige Auf- und Verteilung der zu speichernden Informationen unter Beachtung sämtlicher "Normalisierungstheoreme": alles Dinge, die nach wie vor optimal mit den Werkzeugen Bleistift und Papier zu erledigen sind.

Nun aber wird endlich das Programm angeworfen. ACCESS zeigt sich mit einem erfreulich kargen Layout. Nichts von den Horror-Visionen dreizeiliger "Funktions"-Leisten mit 50 und mehr winzigen Icons und tiefgestaffelten Menu-Ungetümen, die die propagierte Benutzerfreundlichkeit ad absurdum führen. ACCESS bietet nur das "Datenbankfenster". Hierin sind die Hauptobjekte der gewählten Datenbank – und zwar zum Umschalten immer diejenigen einer der 6 Arten – aufgelistet. Als da sind: Tabellen (in denen die Daten enthalten sind), "Abfragen" (die z.B. Selektionskriterien speichern), "Formulare", "Berichte", "Makros" sowie (ACCESS-BASIC-) "Module". Zunächst ist natürlich alles leer, weil wir noch kein "Objekt" erzeugt haben.

Erster Schritt: Einrichtung der Tabellen für die Daten. Eine freudige Überraschung: Feldnamen können in ACCESS (fast) beliebig lang sein und Groß- und Kleinbuchstaben, Umlaute (!), Sonderzeichen inkl. -Zeichen (!!) und sogar Leerzeichen (!!!) enthalten. "Die Vornamen der Hüter" wäre z.B. eine zulässige Feldbezeichnung. Wer sich jahrelang das Hirn zerbrochen hat, um in 10 Zeichen eine sinnvolle Bezeichnung unterzubringen (Beispiel: HUEPRIT_VN), wird diese Neuerung sehr zu schätzen wissen. Ebenso übrigens die Kommentarzeile, die für jedes Feld beschrieben werden kann. Und da wir ja unter WINDOWS arbeiten, kann man natürlich auf leichteste Weise die Feldreihenfolge per "drag and drop" mit der Maus ändern, Felder in und aus der Zwischenablage kopieren etc.. Die Feldtypen sind die heute üblichen, abgesehen vielleicht von den Arten "Währung" und "Zähler". Letztgenannter verpaßt automatisch jedem Datensatz eine singuläre Zahl und eignet sich damit hervorragend als Beziehungsfeld für relationale Verknüpfungen mehrerer Tabellen.

"Fremd"-Tabellen einbinden

An dieser Stelle ist auch einer der bahnbrechenden Besonderheiten von ACCESS zu erwähnen: die Möglichkeiten, Daten aus Datenbanken "fremder" Formate nicht nur zu importieren, sondern auch "einzubinden". Es ist z.B. überhaupt kein Problem, eine dBASE-, eine PARADOX, eine BTRIEVE oder eine FOXPRO-Datei (um nur die wichtigsten For-



Am Anfang war die Skepsis ...

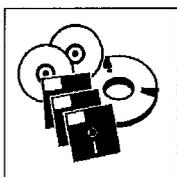
... aber auch die Neugier

*MS ACCESS – objektorientierte
Windows-Datenbank*

Nicht jeder Anfang ist schwer

Daten in Tabellen

*Robert Suermann ist Richter am
Oberlandesgericht Oldenburg.*



mate zu nennen) in eine ACCESS-Datenbank einzubinden. Das bedeutet, daß die Daten ACCESS so zur Verfügung stehen, als befänden sie sich in einer ACCESS-Datei, tatsächlich aber unverändert an ihrem Ort und in ihrem jeweiligen Format verbleiben und natürlich auch weiterhin mit ihren "eigenen" Programmen bearbeitet werden können. Möglich macht dies der von Microsoft entwickelte ODBC ("Open DataBase Connectivity")-Treiber. Das ist nicht nur ein äußerst geschickt ausgeworfener Köder für potentielle Programmwechsler, sondern eine wirklich feine Sache. Er ermöglicht nämlich, bewährte zeichenorientierte Datenbanken mit ihren zugehörigen Anwendungen oder Programmen, etwa dBASE-Code oder CLIPPER-Exe-Files, weiterhin zu nutzen und zusätzlich ein schicken grafisches WINDOWS-Front-End unter ACCESS zur Verfügung zu stellen.

Formulare

Die Tabelle ist fertig definiert. Mit einem Mausklick wird vom Entwurfs-Modus in den Datenblatt-Modus gewechselt, der eine Listendarstellung bringt und Eingaben und Editieren erlaubt. Im Kern ist die Datenbank nun schon fertig. Rudimentär ließe sich – da auch Blätter- und Suchfunktionen eingebaut sind – damit schon arbeiten.

*Ungeahnte technische
Möglichkeiten zur
"künstlerischen" Gestaltung tun
sich auf!*

Dem Geist von ACCESS wird aber nur gerecht, wer sich jetzt ans Entwerfen von Formularen macht. Das geht per Hand oder (für Standard-Layouts) komfortabel mit einem "Assistenten" (so die schlappe Übersetzung für den englisch "wizzard", also Zauberer). Hier nun schlägt die Grafik voll zu: Mit Maus und ein paar Hilfsmitteln wie "Toolbox", "Palette" sowie Schriftart- und Schriftgrößenwahl sind der Kreativität kaum Grenzen gesetzt. In kürzester Zeit lassen sich Formulare erzeugen, die mit 10 Schriftarten und 20 Farbkombinationen den Benutzer mit Gewißheit in den Wahnsinn treiben, zumal sich auch per OLE noch Bilder einbinden lassen. Als bald aber merkt der "Designer" (hoffentlich) zweierlei: daß nämlich zum einen weniger mehr ist, und sich dringend ein sparsamer Umgang mit den Gestaltungsmitteln empfiehlt (vielleicht kehrt er auch reumütig zu den zunächst als langweilig empfundenen dezenten Farbwerten zurück, die der "Assistent" vorschlägt), und zweitens, daß man unheimlich viel Zeit verplempern kann mit der Feinarbeit am grafischen Formularentwurf. Man ertappt sich etwa dabei, daß man eine geschlagene Stunde eigentlich nur damit zugebracht hat, das eine Feld um drei Millimeter (oder besser nur zwei Millimeter?) nach oben zu verschieben, ein anderes Feld immer wieder neu zu kolorieren, diverse Schriftarten auszuprobieren etc..

Sinnvoller ist da schon der (leicht mögliche) Einbau der bekannten WINDOWS-Komponenten wie Radio-Buttons, List-Boxen oder Kombo-Boxen in Formulare. Hiermit kann man dem Benutzer wirklich einen Dienst erweisen. Muß die Anwendung etwa alle (ach so wichtigen) dienstlichen Anreden von "Frau Vizepräsidentin des Oberlandesgerichts" bis zum "Herrn Justizoberamtsrat" korrekt absondern können, so lassen sich diese schnell in eine List-Box einstellen; der Benutzer kann dann den gesuchten Dienstgrad durch einfaches Anklicken komfortabel auswählen.

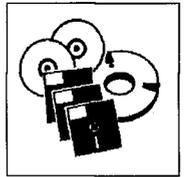
Objekte mit Eigenschaften

Zurück zur Nüchternheit

Ist die Graphik-Euphorie abgeklungen wird der Blick frei für die Objekt-Orientierung von ACCESS, die einem beim Formular-Entwurf am deutlichsten entgegentritt. Was ist darunter zu verstehen? Unter ACCESS gibt es nur "Objekte": z.B. ist eine Datenbank als ganzes ein Objekt, eine Tabelle ist ein Objekt, ein Formular ist ein Objekt, aber auch ein Feld zum Anzeigen von Daten in einem Formular ist ebenso ein Objekt wie ein dort gemaltes Rechteck oder ein im Formular angebrachter Push-Button. Und weil dies alles Objekte sind, haben sie alle "Eigenschaften", und zwar nicht zu knapp! Ein Formular hat z.B. die Eigenschaft sichtbar zu sein (immer, nie, auf dem Bildschirm, beim Drucken); es kann Bildlaufleisten haben, es kann vergrößerbar sein, es kann aktivierbar sein, es kann eine bestimmte Hintergrund-Farbe haben, es kann einen Formulkopf besitzen usw. usw. Diese Eigenschaften können alle über Eintragungen bzw. Anklicken im "Eigenschaftsfenster" bestimmt werden.

Befremdliche Begriffe ...

Neben diesen Eigenschaften, die das Äußere des Objekts "Formular" bestimmen, hat dieses noch eine Reihe zunächst befremdlich klingender Eigenschaften, zB. die Eigenschaft "beim Öffnen". Ein Objekt vom Typ "Textfeld" (das Feldinhalte einer Tabelle und/oder berechnete Werte in einem Formular anzeigt) hat u.a. die Eigenschaften "beim Hingehen", "vor Aktualisierung", "beim Verlassen". Diese Eigenschaften erlauben es, auf die entspre-



chenden Ereignisse in beliebiger Weise zu reagieren. Die Reaktion wird festgelegt durch den Namen eines Makros oder einer ACCES-Basic-Funktion (dazu s.u.), deren Namen in der entsprechenden Eigenschaftszeile eingetragen werden.

Da bleibt nun kein Wunsch offen. Angefangen vom simplen Ausstoßen eines Warntons beim Betreten eines bestimmten Feldes, über das Sichtbarmachen von weiteren Feldern oder Hinweisen bei Eingabe bestimmter Werte, über ausgeklügelte Gültigkeits- und Plausibilitätskontrollen bis hin zu umfangreichen Datenoperationen: alles kann über die Eigenschaften der selbstgeschaffenen Objekte ausgelöst werden. Größere Operationen würde man an die Eigenschaft "beim Klicken" oder "beim Doppelklicken" eines Push-Buttons (=Objekt vom Typ "Schaltfläche") binden. So läßt sich z.B. das nächstbereite Aktenzeichen durch Klicken auf eine Schaltfläche mit der Aufschrift "Nächstes Aktenzeichen" holen; durch Anklicken eines Buttons mit der Beschriftung "Alle Vorgänge des Vorjahres löschen" kann man ebendieses bewerkstelligen etc..

... und immer wieder neue Möglichkeiten

Chaos?

Wenn man in dieser Weise eine zeitlang schöne und hoffentlich auch praktische Formulare entwickelt hat, macht sich allerdings immer deutlicher eine gewisse Unübersichtlichkeit bemerkbar. Dutzende von Objekten sind geschaffen worden; Hunderte von Eigenschaften lösen mannigfache Aktionen aus: wie da den Überblick behalten? Denn die furchtbare Wahrheit ist: ES GIBT KEINEN PROGRAMMCODE. Das muß der Programmierer alter Schule erst mal verdauen. Kein Listing, das den Programmablauf wiedergäbe! Nichts, was man ausdrucken und für die Programm-Dokumentation (oder auch die Fehlersuche!) verwenden könnte, sondern nur haufenweise Objekte, Eigenschaften und hiervon ausgelöste Einzelaktionen! Ein wenig Orientierung kann man sich immerhin verschaffen durch: a) systematisches Benennen der Objekte und Eigenschafts-Aktionen nach einem mit eisernen Konsequenz eingehaltenen Schema, und b) fleißiges Anbringen möglichst vieler Kommentare.

Nur durch Systematik zur Übersicht

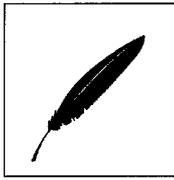
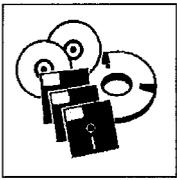
Makros

Letzteres gilt vor allem für die Makros, die glücklicherweise für jede Aktion gleich eine Kommentarspalte bereithalten. Makros sind in ACCESS die Arbeitspferde, die die Masse der Aktionen durchführen, die von irgendwelchen Objekt-Eigenschaften ausgelöst werden. Mit Makros kann man (fast) alles machen. Das geht vom Öffnen und Schließen von Formularen über das Setzen der Werte von Eigenschaften (etwa Größen- oder Farbänderungen) bis hin zu diversen Datenmanipulationen wie "Satz speichern", "Satz Löschen" etc.. Allerdings gibt es keinen sogenannten Makro-Recorder, mit dessen Hilfe man Tastenanschläge aufzeichnen und später wieder "abspielen" lassen kann, wie man das von vielen Programmen her kennt. Stattdessen werden die einzelnen Makro-Befehle in Zeilen einer Tabelle eingetragen, deren wichtigste Spalte die jeweils auszuführende Aktion ist. Diese kann bequem aus einer sich öffnenden Liste ausgewählt werden, die alle zulässigen Aktionen anbietet. In einem Extra-Fenster werden dann zu jeder Aktionsart die weiteren Einzelheiten festgelegt. Soll die Aktion etwa im Aufblenden eines Hinweis-Fensters bestehen, so ist dort der Wortlaut der Meldung einzugeben. Die Bedienung ist hier – wie übrigens fast überall in ACCESS – sehr gut durchdacht und vorbildlich. Ein Makro läßt sich recht flott und komfortabel zusammenstellen.

ACCESS-Basic

Es gibt – im normalen Rahmen – nicht viele Dinge, die man mit einer Datenbank anstellen möchte, die sich in ACCESS nicht über ein Makro realisieren lassen. Einiges geht aber doch nur – oder eleganter – mithilfe eines Stückchens BASIC-Code. Hier angekommen wird manch altgedienter Programmierer aufatmen: endlich hat er wieder vertrauten Boden unter den Füßen. Aber: zu früh gefreut, Objekte lauern auch hier, und zwar besonders heimtückische. Zwar ist der größte Teil von ACCESS-BASIC "normaler" BASIC-Code à la Quick-Basic etc.. So lassen sich in der üblichen BASIC-Weise Variable definieren, Werte abfragen und berechnen, Texte anzeigen u.s.w., wobei natürlich WINDOWS-Funktionen wie etwa "MsgBox" zum Ausgeben einer Meldung in einem WINDOWS-Fenster komfortabel zur Verfügung stehen. Vielen dürfte dergleichen von der Makrosprache in WINDOW bekannt sein.

ACCESS kann – fast – alles, aber gänzlich ohne BASIC-Code geht es doch nicht.



Es bleibt dabei: Alles ist anders.

Etwas ganz anderes – und völlig neues – sind hingegen alle auf die Datenbank bezogenen Befehle. Hier muß ja deren Objektbezogenheit in der Programmiersprache repräsentiert werden. Das führt zu folgender Vorgehensweise: zuerst müssen für Datenbank und (z.B.) eine Tabelle Variable eines besonderen Typs eingerichtet werden, denen dann eine konkrete Datenbank und eine konkrete Tabelle zugeordnet werden. Dann kann man mit Eigenschaften und Methoden dieser Datenbankobjekte sich in der Datenbank bewegen und die Daten manipulieren. Ein kleines Beispiel:

Es sollen alle Sätze einer Datenbank ("kunden") der Reihe nach durchlaufen werden und zu den Werten eines dort enthaltenen Zahlenfeldes ("betrag") jeweils 100 addiert werden. Unter ACCESS müßte man dafür etwa schreiben:

```
DIM datbank as Database
DIM tab as Table
SET datbank = CURRENTDB()
SET tab = datbank.CreateTable("kunden")
DO UNTIL NOT tab.eof
    tab.edit
    tab![betrag] = tab![Betrag] + 100
    tab.update
    tab.MoveNext
LOOP
```

Im herkömmlichen xBASE-Code ginge das kürzer und sähe etwa so aus:

```
USE kunden
DO WHILE NOT EOF()
    betrag = betrag + 100
    SKIP
ENDDO
```

Hier soll nicht in die Tiefen der Programmierung eingedrungen werden. Das kleine Beispiel zeigt aber wohl schon die ganz andere Ansatzweise, eben die Objektorientierung, von ACCESS. Zum nächsten Satz springt man z.B. in ACCESS nicht durch einen Befehl wie SKIP, sondern durch Aktivierung der Methode "MoveNext" des entsprechenden Datenbankobjekts (im Beispiel: tab.MoveNext). Dieser Unterschied ist dabei keineswegs nur der einer unterschiedlichen Schreibweise, wie man anhand des obigen Mini-Beispiels vielleicht glauben könnte, sondern hat sehr tiefgehende Auswirkungen, auf die hier aber nicht näher einzugehen ist. (Die obige Beispielaufgabe wurde übrigens nur zur Verdeutlichung gewählt; in der Praxis würde man die Anpassung des Feldes "betrag" in ACCESS ohne jede Programmierung durch "Zusammenklicken" einer Aktualisierungsabfrage in maximal ca. 20 Sekunden realisieren.)

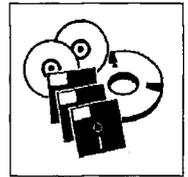
Wenn man sich mit ACCESS-BASIC vertraut gemacht hat (was freilich eine gehörige Zeit beansprucht), läßt sich das folgendes Resümee ziehen: Mit ACCESS-BASIC lassen sich alle erforderlichen Datenbankoperationen programmieren: viele sehr elegant, manche eher umständlich und wenig übersichtlich. Der (von den zwar exakten, aber knapp gehaltenen Instruktionen im Handbuch nicht gerade erleichterte) Zugang zur Datenbankprogrammierung ist definitiv nicht ohne Mühe.

Abfragen: die reine Freude

Das genaue Gegenteil trifft zu auf das Entwerfen von Formularen (s.o.) und Auswertungen ("Berichten") sowie auf das Erstellen von "Abfragen". Gerade letztere lassen sich – auch bei komplizierteren Selektionen – mit mehrfachen UND/ODER Verknüpfungen und diversen Sortierungen extrem einfach erstellen. Hier ist höchstes Lob angebracht. Dem Verfasser ist noch keine Datenbank untergekommen, die an dieser – zentralen – Stelle eine so gut durchdachte Benutzerführung aufgewiesen hätte. Nach wenigen Minuten hat man den Trick raus und kann fortan blitzschnell die ausgefallensten Filter "zusammenklicken". Und man kann diese Abfragen in beliebiger Zahl zur Wiederverwendung speichern und mit so aussagekräftigen Namen versehen wie "Kunden in Köln sortiert nach Namen". Man kann auch eine solche Abfrage wiederum zur Basis einer weiteren Abfrage machen. Abfragen über mehrere Tabellen hinweg sind überhaupt kein Problem. Relationale Verknüpfungen

*Nicht unerhebliche
Einarbeitungszeit: Vieles ist
elegant gelöst, aber wo Licht ist,
ist auch Schatten.*

*Glanzstücke des Programms:
Entwerfen von Formularen,
Auswertung und Erstellung von
"Abfragen"*



von zwei oder mehr Tabellen werden durch "drag and drop" eines der Verknüpfungsfelder auf das andere erzeugt und von ACCESS sofort durch eine Verbindungslinie augenfällig gemacht. Das geht alles sehr komfortabel und ist höchst effektiv (außerdem macht es Spaß). Und da man jede so erstellte Abfrage zur Grundlage eines Formulars machen kann, kann man sich das Ergebnis der Selektion in jeder gewünschten Form zu Gemüte führen.

Fazit

Die überragende Stärke von ACCESS ist die Qualität der Benutzerführung, die ein Datenbank-Handling mit bisher nicht gekannter Leichtigkeit und Effizienz ermöglicht. Hier ist Microsoft wirklich ein großer Wurf gelungen. Das Programm kann daher als allererste Wahl für diejenigen bezeichnet werden, die interaktiv von der Programmoberfläche aus praktisch mit Datenbanken arbeiten.

Für die Entwickler von Datenbank-Anwendungen ist das Bild insofern etwas getrübt, als diese zunächst ein gehöriges Maß an Lern- und Umstellungsaufwand in Kauf nehmen müssen. Ist dieser Schritt aber bewältigt und braucht man – wie wohl zumeist – kein System, bei dem – etwa wegen riesiger Datenbestände – ein rasantes Zeitverhalten im Vordergrund steht, dann ist auch hier ACCESS uneingeschränkt sehr zu empfehlen. Sind die Anfangshürden erst einmal genommen, lassen sich mit ACCESS in kurzer Zeit äußerst ansprechende professionelle Anwendungen erstellen. Hinzu kommt noch der Vorzug, zahlreiche Fremdformate ohne Datenbankeingriffe verarbeiten zu können. ACCESS ermöglicht es so, auch angejahrte Datenbanken mit einem modernen Front-End zu versehen. Wichtig für Entwickler ist auch, daß nach einmaligem Erwerb des "Distribution Kit" ACCESS-Anwendungen ohne Lizenzgebühren beliebig oft weitergegeben werden können.

Auch im Justiz-Kontext sind die geschilderten Vorzüge von ACCESS gut zu gebrauchen, zumal, wenn die Zugehörigkeit des Programms zur WINDOWS-Familie für die automatisierte Zusammenarbeit mit einer WINDOWS-Textverarbeitung ausgenutzt wird. So verwundert es nicht, daß in einigen Landesjustizverwaltungen (Bremen, Nordrhein-Westfalen, Niedersachsen) aktiv an der Erstellung von Geschäftsstellenverwaltungs-Programmen mit integrierter Schreibwerk-Erstellung unter ACCESS und WINWORD gearbeitet wird, bzw. sich schon einige Systeme im Pilot- oder Echtbetrieb befinden.

Benutzerführung: ein großer Wurf!

Landesjustizverwaltungen zeigen sich bereits interessiert.