

# Visual dBASE 5.5 – Appendix oder Tools für juristische Arbeiten?

Rainer Koitz

Frühere dBASE-Versionen sind auch im juristischen Bereich genutzt worden.<sup>1</sup> Die Präsentation von dBASE 5.0 für Windows Ende 1994 wurde von Juristen zwar vermerkt<sup>2</sup>, sie führte jedoch zu keinen in ihrer Sphäre bekannten Anwendungen. Das mag einmal aus der späten Verfügbarkeit resultieren; Microsoft Windows wurde bereits seit Jahren genutzt, mehrfach angekündigte Erscheinungstermine einer dBASE-Version für Windows wurden nicht eingehalten, und viele Anwender nutzten inzwischen andere Windows-Datenbanken bzw. andere nutzerfreundliche Technologien. Für juristische Aufgabenstellungen dürfte letzteres namentlich die VBA-Komponenten<sup>3</sup> einiger Microsoft-Produkte betreffen. Zum anderen war dBASE 5.0 zwar durch wesentliche Erweiterungen, aber auch zahlreiche Unzulänglichkeiten gekennzeichnet.<sup>4</sup> Letztere führten zur überwiegend kritischen Aufnahme<sup>5</sup>, bei der jedoch die leistungsfähigen Neuerungen oft vernachlässigt wurden. Borland hat nun bereits im Frühherbst 1995 eine Nachfolgeversion offeriert – Visual dBASE 5.5. Der Autor hat diese wie dBASE 5.0 für Windows getestet und wird sie im jetzt beginnenden Sommersemester 1996 in die Wirtschaftsinformatik-Ausbildung von Studenten der Betriebswirtschaft einbeziehen. Gewissermaßen in Fortschreibung der früheren Nutzung von dBASE für juristische Arbeiten<sup>6</sup> könnten seine Erkenntnisse jedoch auch interessierte Juristen anregen. Es versteht sich bei dem erheblichen Leistungsumfang, daß die Untersuchung, noch mehr die Darstellung auf wesentliche Komponenten von dBASE 5.5 beschränkt bleiben muß. Wesentlich ist zunächst die Nutzbarkeit unter Windows (s. 1.). Charakteristisch für alle dBASE-Versionen ist die gleichzeitige Nutzbarkeit als relationales Datenbankbetriebssystem (dazu 2.) und Programmiersystem. Dabei kann individuell oder mittels vorgefertigter Bausteine programmiert werden. Wurden diese Bausteine bisher über Generatoren bereitgestellt, so wird ihre Nutzung nunmehr über sog. Designer und Experten qualifiziert, worauf unter 3. eingegangen wird. Die Programmierung kann wie bisher prozedural erfolgen, sie ist jedoch insbesondere bei der Eingabesteuerung, den sog. Formularen, auch objektorientiert möglich. Diese auch für die juristische Arbeit gravierende Erweiterung wird unter 4. skizziert. Den Abschluß (5.) bildet dann eine Einschätzung der Textverarbeitung unter dBASE 5.5. Nicht eingegangen wird auf den für Entwickler unverzichtbaren, zusätzlichen Compiler zur Erstellung ausführbarer (.EXE) Programme für die lizenzfreie Weitergabe und das für diese sinnvolle Client/Server-Entwicklerpaket, die extra zu erwerben sind und dem Verfasser nicht zur Verfügung standen.

## 1. Benutzeroberfläche

Die Oberfläche von Visual DBase 5.5 bereitet dem mit Windows Vertrauten keine Probleme: Sie enthält den Windows-typischen Desktop mit Titelleiste, (sparsamer) Menüleiste, Leiste mit Fensterbedienelementen und Statuszeile. Zusätzlich umfaßt sie zwei permanent geöffnete Komponenten, die den mit dBASE Vertrauten nicht überraschen: das Fenster des Regiezentrum und das zweigeteilte Befehlsfenster. Diese wie jedes weitere geöffnete Fenster weisen wiederum die Windows-typischen Elemente auf: Systemmenü, Schalterleiste, Bildlaufleiste(n) und Statuszeile.

Vertraute Umgebung

Prof. Dr. rer. oec. habil. Rainer Koitz lehrt an der Hochschule für Technik und Wirtschaft Dresden (FH) DV-Recht, Datenschutz und Wirtschaftsinformatik.

<sup>1</sup> S. etwa König, jur-pc 1/89, 23; Koitz, jur-pc 2/89, 67; Rühle, jur-pc 4/89, 156; Hoffmann, jur-pc 9/89, 322; Zurek, jur-pc 10/91, 1250.

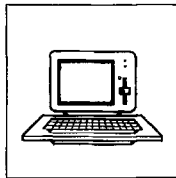
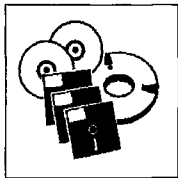
<sup>2</sup> S. Herberger, jur-pc 12/94, 2915.

<sup>3</sup> Visual Basic for Applications; s. Herberger, jur-pc 4/95, 3080 und Fortsetzungen.

<sup>4</sup> Beispielsweise fehlte im Menü Datei die Option Exportieren o. ä. Den wiederholt angezeigten Hinweisen auf Instabilitäten und Systemabstürzen begegnet man unter Visual dBASE 5.5 vergleichsweise selten (und in einem für komplexe Operationen in anderen Windows-Anwendungen gewohnten Umfang).

<sup>5</sup> S. etwa Kunz, Datenbank Fokus 2/95, 70; Parthier, Datenbank Fokus 3/95, 3; Gladis/Hartnick, c't 7/95, 214.

<sup>6</sup> S. etwa Koitz/Kemper, Rechtsinformatik, Berlin 1989, 142.



Die drei Bereiche des Regie-Zentrums

Zwei Modi: Entwurfs- und Ausführungsmodus

Parallelität von Regiezentrum und Befehlsfenster: Nicht nur historisch bedingt ...

Weitere Pro-Argumente:

“Multi-Task-Zustand”

Anpassungsfähige Benutzeroberfläche

Menü-Struktur

Visual dBASE ist in vielerlei Hinsicht aufwärtskompatibel: Dem Nutzer früherer dBASE-Versionen wird Gewohntes angeboten, möglichst in besserer Weise. In diesem Sinne wird die Fensterschnittstelle weiter als Regiezentrum bezeichnet, sie ist jedoch gegenüber dBASE IV konsequent weiterentwickelt:

- Der obere Bereich dient der (permanent möglichen) Einstellung des aktuellen Verzeichnisses bzw. – soweit bei Tabellen (s. 2.) definiert – der aktuellen Datenbank.
- Der linke untere Bereich ermöglicht die Auswahl des gewünschten Datentyps (Tabellen; Abfragen, Formulare, Reports, Etiketten – s. 3.; Programme, Grafiken, Sounddateien, Kataloge und individuell gewünschte andere Dateien).
- Im rechten unteren Bereich erfolgen die Anzeige und Auswahl der zum fixierten Datentyp im aktuellen Verzeichnis (bzw. in der aktuellen Datenbank) gehörenden Dateien.

Anders als zahlreiche Microsoft-Produkte unterscheidet Visual dBASE bei der Arbeit mit den verschiedenen Dateien grundsätzlich zwischen Entwurfs- und Ausführungsmodus, woraus sich auch die permanente Wählbarkeit der Dateigesamtheit erklärt. Im Entwurfsmodus werden die notwendigen, individuell unterschiedlichen Voraussetzungen für die eigentliche Dateinutzung getroffen: Strukturdefinition für Tabellen, Fixierung der Bedingungen für Abfragen, Objektbeschreibung/-positionierung für Formulare, Befehlsfolge für Programme u. a. Für diesen Modus stellt Visual dBASE jeweils wahlweise sog. Experten und Designer zur Verfügung (s. 3.). Im Ausführungsmodus erfolgt dann die eigentliche Anwendung der vorher entworfenen Dateien: Eingabe/Änderung von Datensätzen in Tabellen, Auswahl von Datenfeldern/Datensätzen mit Abfragen, Interaktion zwischen Nutzer und Formularen, Abarbeitung von Programmen u. a.

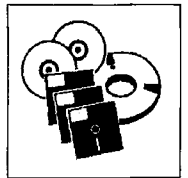
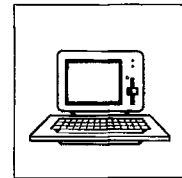
Das Befehlsfenster mit Eingabe- und Ausgabebereich und die alternativ zum Regiezentrum mögliche Befehlssteuerung scheinen in einer Windows-Applikation auf den ersten Blick nur historisch bedingt zu sein, nämlich hinsichtlich der Kompatibilität zum dBASE-Punktmodus. Zusätzlich zum historischen Moment, mit dem wir bei der Ausführung von dBASE IV-Programmen bei Beachtung der wenigen Einschränkungen – s. 2. – keine Schwierigkeiten hatten, lassen sich folgende Argumente für die Parallelität von Regiezentrum und Befehlsfenster, im eigentlichen von Fenster- und Befehlssteuerung anführen (auch gegen einen vermeintlichen Windows-Standard):

- Im Eingabebereich des Befehlsfensters erfolgt durch Visual dBASE eine schritthaltende Anzeige der Befehle, die den Operationen in anderen Fenstern entsprechen. Damit wird die Kenntnis der dBASE-Sprache und wohl auch das Systemverständnis gefördert. Zugleich steht so ein nachvollziehbares Protokoll der Arbeit zur Verfügung.
- Verschiedentlich ist eine Befehlseingabe einfacher als die entsprechende fenster- und menügesteuerte Operation; im Einzelfall wie der Katalogerstellung ist die Befehlseingabe unverzichtbar. Dabei lassen sich auch im Befehlsfenster die Windows-Technologien zur Textbearbeitung nutzen.
- Im Ausgabebereich ist die Anzeige wichtiger, sonst nicht verfügbarer Informationen möglich, wie des Verzeichnisinhalts oder des Systemstatus. Zusätzlich kann der Ausgabebereich als Alternative oder Zusatz zur Druckerausgabe herangezogen werden.
- Das Befehlsfenster erlaubt auch die Fixierung und Ausführung von Programmteilen. Es kann daher zusätzliche Hilfe beim Programmtest sein.

Neben der Hierarchie der Verzeichnisse, Dateien und der weiterhin nutzbaren Arbeitsbereiche und Kataloge ist eine weitere Neuerung zu beachten: Jedem geöffneten Fenster wird, falls nicht anders angewiesen, eine Arbeitssitzung zugeordnet. Folgerichtig bewirkt die gewohnte Option *Datei/Schließen* das Schließen der aktiven Sitzung. Damit entsteht ein multi-task-Zustand, der beispielsweise verschiedene Operationen mit der gleichen Tabelle ermöglicht, etwa das gleichzeitige Arbeiten mit unterschiedlicher Indizierung. Das erhöht die Vielfalt der Bearbeitung, kann aber auch irritieren: Wer im Befehlsfenster und in anderen Fenstern arbeitet, erkennt aus der Anzeige im Befehlsfenster nicht, welche Arbeitssitzung (und damit regelmäßig welche Datei) in welchem Fenster aktiv ist.

Die Benutzeroberfläche läßt sich teilweise individuell anpassen, etwa über das Menü *Eigenschaften/Desktop*. Bei früheren dBASE-Versionen gingen diese Einstellungen verloren, wenn dBASE verlassen wurde. Visual dBASE dagegen speichert die Einstellungen sofort, so daß sie beim nächsten dBASE-Start berücksichtigt werden.

Wer die Systemmenüs der einzelnen Fenster untersucht, stellt zunächst Änderungen gegenüber anderen Windows-Applikationen fest, die aus den erwähnten Modi resultieren. Das auf den ersten Blick fehlende *Datei/Speichern unter* ist so mit dem Entwurfsmodus verzichtbar, der regelmäßig mit der Anforderung des Dateinamens beginnt. Und für das zusätzliche Abspeichern steht das Angebot *Datei/Exportieren* zur Verfügung. Zum anderen



fällt ins Auge, daß die Systemmenüs der meisten Designer (Ausnahme: Crystal Reports) zunächst grobe Klassen bilden und dann verschiedentlich ungewohnt viele Hierarchieebenen durchlaufen werden müssen. So erfordert das Anlegen eines Tabellenindex die Optionen *Tabelle/Tabellenoperationen/Indizes verwalten/Neu* (in der Dialogbox *Indizes verwalten*)/(Dialogbox) *Index definieren*. Für den geübten Datenbanknutzer überraschend fehlt dann auch ein Button zum Indizieren in der Schalterleiste des Tabellendesigners, und im Objektmenü ist selbst die Option *Tabellenoperationen* nicht enthalten.

Wünschenswert wäre verschiedentlich die in vielen Windows-Applikationen verfügbare Option *Datei/Seitenansicht*. Sonst läßt sich zwar über die Angebote teilweise diskutieren, die erforderlichen Datenbankoperationen sind jedoch alle durch die kontextsensitiven Pulldown-Menüs der Systemleiste erreichbar. Auch die mittels der rechten Maustaste zu öffnenden Objektmenüs offerieren meist – wie von fortgeschrittenen Windows-Applikationen gewohnt – die für die jeweilige Arbeitssituation typischen Möglichkeiten.

*Seitenansichts-Funktion fehlt noch.*

## 2. Datenbanksprache

Visual dBASE bleibt bei aller Visualisierung und objektorientierter Programmierung ein relationales Datenbankbetriebssystem. Dem relationalen Konzept entsprechend werden die eigentlichen Dateien als Tabellen bezeichnet, und sie können (alternativ zu einem Formular- und einem Maskenlayout) so auch angezeigt werden. Tabellen lassen sich dann auch in Datenbanken zusammenfassen; der in früheren dBASE-Versionen und anderer Standardsoftware weiterhin verbreiteten unsinnigen Bezeichnung einer Datei als Datenbank wurde ein Ende gesetzt.

*Weiterhin relationales Datenbankbetriebssystem*

Andere "alte Zöpfe" wurden dagegen nicht abgeschnitten. Nur die mit dem gewählten Konzept unvereinbaren dBASE IV-Instruktionen werden nicht abgearbeitet. Doch selbst bei der Nutzung nicht unterstützter (alter) Befehle wird regelmäßig eine Warnung angezeigt, und für nicht unterstützte Funktionen wird ein definierter Wert übergeben. Alle anderen dBASE IV-Befehle und -Funktionen können herangezogen werden, auch wenn ihnen eine modernere Entsprechung gegenübersteht. Insbesondere betrifft dies alle Instruktionen für die Fenster- und Menüarbeit in dBASE IV. Das mag für manche der bisherigen dBASE-Nutzer vorteilhaft sein, kann aber alten und neuen Interessenten die Sicht auf die zahlreichen Erweiterungen verstellen.

*"dBASE IV in Visual dBASE"*

Da auf die wichtigsten Neuerungen in den folgenden Abschnitten eingegangen wird, seien hier noch die weitgehende SQL-Unterstützung, die für versierte Programmierer hilfreichen Debugger, Preprozessor und Einfügung externer Funktionen (Prototyping, z. B. aus der Windows-API) sowie die von uns noch nicht getestete Windows 95-Unterstützung und die Instruktionen zur Zugriffsregelung auf Tabellen erwähnt. Letztere definieren insbesondere im Netzbetrieb wichtige Rechte verschiedener Nutzer, wobei zusätzlich die mögliche und teilweise notwendige Einschaltung des Exklusivmodus zu beachten ist. Zu den von Visual dBASE mit weiteren Befehlen und Funktionen unterstützten Netzwerken gehört auch Novell Netware, in dem (Version 3.11) wir keine netzspezifischen Schwierigkeiten erkennen konnten. Windows NT ist übrigens nicht in der Liste unterstützter Netzwerke ausgewiesen, was bei dessen derzeit mangelnder Objektorientierung nicht verwundert.

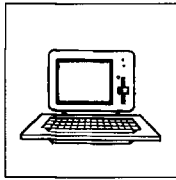
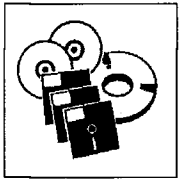
*Wichtige Neuerungen*

Die eigentlichen Tabellenoperationen sind in Programmen oder im Befehlsfenster wie unter dBASE IV anzuweisen. Jedoch ergeben sich deutlich geringere Operationszeiten, was für umfangreiche Anwendungen etwa mit Tabellen-(so auch Hypertext-)Verknüpfungen trotz immer leistungsfähigerer Prozessoren hervorzuheben bleibt. Auch insoweit hat sich die vollständig neue Codierung von Visual dBASE (wie von dBASE für Windows) mittels Borland C++ gelohnt. Wir haben durchschnittliche Laufzeitverbesserungen für das Indizieren sowie das Hinzufügen, das Löschen und das Ersetzen jeweils in einer indizierten Datei von 1,75:1 (Indizieren) bis 1,4:1 (Ersetzen) zwischen dBASE IV und Visual dBASE ermittelt. Allerdings waren die Laufzeiten für die jeweiligen Operationen nicht nur unterschiedlich, sondern sie lagen im Durchschnitt auch über denen von dBASE 5.0 für Windows. Abgesehen vom ebenfalls erkennbaren Einsatz von Optimierungsroutinen sind beide Effekte mit der C++-Objektorientierung plausibel. Beiläufig sei vermerkt, daß auch Benchmark-Tests mit dem Befehl GENERATE und den Funktionen SECONDS und ELAPSED besser als in dBASE IV unterstützt werden.

*Performance-Steigerung*

Bei den Datenfeldern bezieht Visual dBASE alle unter dBASE IV möglichen Elemente in die Tabellen ein. Daß dabei Gleitkommfelder nur formal unterstützt werden, tangiert den hier interessierenden Bereich nur marginal. Wesentlich sind dagegen die Erweiterungen mit

*Datenfelder*



*Textverbindungen mit  
OLE-Feldern*

*Sowohl DDE-Client als auch  
DDE-Server*

*Perfekt: Interaktivität zwischen  
Benutzeroberfläche und der  
Befehlssprache*

*Experten(?) und Designer*

den Feldtypen OLE und binär. Beide knüpfen an die Konzeption der Memofelder an, indem sie eine feste Länge von 10 Bytes haben und auf zusätzliche (.DBT-)Dateien zeigen. Binärfelder beziehen beliebige im Binärformat gespeicherte Dateien ein. Sinnvoll (und so auch angezeigt) ist die Verknüpfung mit Bild- (.BMP oder .PCX) und Tondaten (.WAV). Sicher sind diese für Juristen verzichtbar, doch können sie Wahlmöglichkeiten, Verknüpfungen und Hinweise in juristischen Informationssystemen untersetzen und bei geschicktem Einsatz die Nutzerakzeptanz erhöhen.

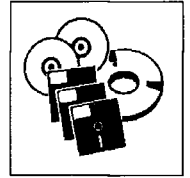
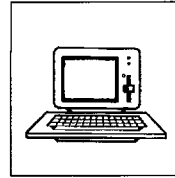
OLE-Felder können Objekte anderer Windows-Anwendungen einbeziehen, eine auf den ersten Blick auch für Juristen gewaltige Unterstützung, wenn man an Textverbindungen denkt. Erste Voraussetzung ist die OLE-Fähigkeit der interessierenden Anwendung – keinesfalls eine Selbstverständlichkeit, wie etwa ein Versuch mit der Microsoft WORKS-Textverarbeitung zeigt. Hilfe gibt insoweit das Windows-Programm REGEDIT.EXE mit einer Liste verfügbarer OLE-Server. OLE erlaubt nun das Einbetten (Embedding) und das Verknüpfen (Linking; auch DDE – Dynamic Data Exchange) von Objekten. Die in Visual dBASE einfachere, aber im Ressourcenverbrauch sichtbar aufwendigere Technik ist das Einbetten, wobei Visual dBASE nur Objekte aufnehmen kann, die als OLE-Client fungieren. Das Einbetten von Microsoft WORD-Dokumenten, verschiedenen Tabellen und Grafiken gelingt ohne Schwierigkeiten. Es sei jedoch noch einmal auf die enorme Zeit- und Speicherbelastung hingewiesen. Bei der Vielzahl denkbarer Verknüpfungen in juristischen Anwendungen scheinen daher die in Visual dBASE unkompliziert möglichen statischen Verbindungen praktisch eher zu favorisieren sein.

Diffiziler sind die DDE-Beziehungen einzuschätzen. Zu einer EXCEL-Tabelle konnten wir sie ohne Schwierigkeiten herstellen, zu WORD 6.0-Dokumenten gelangen sie im Regiezentrum mit der (im meist ausreichend informativen Hilfe-System empfohlenen) Option *Bearbeiten/Verknüpfung einfügen* erst auf Umwegen. Die Schwierigkeiten dürften darin begründet sein, daß Visual dBASE nicht nur DDE-Objekte empfangen, sondern auch bereitstellen, sowohl DDE-Client als auch DDE-Server sein kann. Deren für Textdokumente nicht einfache Unterscheidung und handhabbare Nutzung wird wohl erst bei der Programmierung mit den Klassen DDELink und DDETopic deutlich.

### 3. Visualisierung

Die bereits für dBASE IV charakteristische alternative Nutzbarkeit einer interaktiven Benutzeroberfläche und der Befehlssprache ist in Visual dBASE perfektioniert worden: Durch Transformation der Sprache wie der Oberfläche in die event-gesteuerte, fensterorientierte Umgebung von Microsoft Windows kann der Nutzer zu jedem Zeitpunkt entweder leistungsfähige visuelle Werkzeuge oder die direkte Befehlseingabe für von ihm gewünschte Entwicklungsarbeiten heranziehen. Neben die neuen Tools – Designer, Experten und über weitere Fenster erreichbare Hilfsmittel – tritt dann zusätzlich die Möglichkeit, generierten Code nicht nur wie bisher mit dem Programmeditor zu bearbeiten, sondern auch umgekehrt (im Abfrage-Designer, im Menü-Designer und im Formular-Designer) Programmtext zur Modifikation mit den visuellen Hilfsmitteln zu präsentieren. Dieser genialen Technologie wegen werden die Werkzeuge inzwischen als two-ways-tools bezeichnet, wobei die "Wege" nicht die Bearbeitungsmöglichkeiten bezeichnen, sondern die Bewegung in beide Richtungen – vom Designer zum Programmeditor und umgekehrt – dank der Kraft der Hilfsmittel. Ginge es nämlich nur um die Bearbeitungsmöglichkeiten, könnte man mit dem teilweisen Nebeneinander von Experten, Designern und Befehlssprache auch von drei Wegen sprechen. Hervorzuheben für die praktische Entwicklungsarbeit bleibt also die Aufhebung der bisherigen "Einbahnstraße" zwischen den Generatoren und der Befehlssprache.

Wer von den Experten in Visual dBASE detaillierte Unterstützung in besonders schwierigen Situationen erwartet, sieht sich getäuscht: Die Experten sollen die Entwicklungsarbeit (nur) für praktisch häufig auftretende Voraussetzungen rationalisieren. Der Tabellenexperte stellt einige, insbesondere in der Betriebswirtschaft typische Tabellen zur Manipulation der Tabellenstruktur und den Entwurf eigener Tabellen zur Verfügung. Selbst in diesem Anwendungsbereich dürfte ein solcher "Experte" bestenfalls für unqualifizierte Mitarbeiter eine Unterstützung darstellen. Im unverzichtbaren (weil auch durch CREATE bzw. MODIFY STRUCTURE aufgerufenen) Tabellen-Designer werden dagegen konzeptionell schlüssig und windowsgemäß (bei Auswahlmöglichkeiten mittels Listenfenstern bzw.



Wippreglern) unkompliziert Strukturanzeigen und -änderungen vorgenommen. Daß allerdings nach Änderung einer Tabellenstruktur die Tabelle geschlossen und dann wieder geöffnet werden sollte, falls nicht eine Performancereduzierung eingestellt wurde, gehört zu den wenigen Ungeschicktheiten<sup>7</sup> von Visual dBASE (nicht der entsprechende Hinweis in der README-Datei).

Für Reports – die formatierte Ausgabe von Tabellendaten (über einen Drucker) – und Etiketten – die Schablonenerzeugung für die Beschriftung mit Tabellendaten – nutzt Visual dBASE ein zusätzlich lizenziertes Modul: Crystal Reports. Da diese Arbeiten Juristen weniger interessieren dürften, seien hier nur der im Vergleich zu dBASE IV verbesserte Leistungsumfang und die alternativ zu den umfassenden Designern nutzbaren Experten mit typischen Entwurfsoptionen erwähnt.

Die herauszuhebenden Neuerungen bei der Dateneingabe und -präsentation sind die Formulare und für diese verfügbare Werkzeuge: Formular-Designer und Formularexperte. Formulare sind gestaltbare Anwendungsoberflächen, die die Datenanzeige ebenso wie beliebige Nutzereingaben ermöglichen. Im Zusammenhang mit der Präsentation von Tabellendaten wird für die Erstellung oft schon der Formularexperte ausreichen. Komfortable Dialogfenster lassen sich einfacher und anschaulicher als durch befehlsweise Programmierung mittels des Formular-Designers entwickeln. Im von diesem bereitgestellten Anwendungsfenster ist die Gestalt des Formulars zu erkennen und zu ändern, und im wahlweise zu öffnenden Dialogfenster "Eigenschaften und Methoden" werden die Merkmale der Formularobjekte angezeigt und modifiziert. Zusätzlich kann der Menü-Designer aufgerufen werden zur Definition eigener Menüleisten, Pulldown- und Popup-Menüs, wobei wiederum das Dialogfenster "Eigenschaften und Methoden" zur Darstellung und Änderung der jeweiligen Menümerkmale herangezogen werden kann. Die Art dieser auch aus juristischer Sicht interessanten Fenster und Merkmale hängt unmittelbar mit der in Visual dBASE möglichen objektorientierten Programmierung zusammen, auf die in 4. eingegangen wird.

Eine gute Hilfe bei der Tabellenauswertung (VIEW) und -verknüpfung (JOIN) bietet schließlich der Abfrage-Designer. Wer jemals die entsprechenden Programmbeefehle mit ihren syntaktischen und semantischen Fallen aufgeschrieben hat, wird die einfache und auch anschauliche Arbeit mit diesem Hilfsmittel schätzen lernen. Denkt man beispielsweise an Tabellen zu Verträgen oder Vertragskomponenten, so erlaubt der Abfrage-Designer die Auswahl interessierender Elemente nach Vorgabe entsprechender Bedingungen und die Verbindung mit Allgemeinen Geschäftsbedingungen oder Datensammlungen eventueller Vertragspartner. Für die Filterdefinition kann mittels *Abfrage/Bedingungen definieren* ein feststehendes Fenster zur Bedingungeingabe geöffnet werden. Die Bedingungen können dann neben den üblichen Vergleichsoperatoren, wie vom dBASE IV-Regiezentrum gewohnt, einen \$-("Inhalt von") und LIKE-Operator zur Trunkierung enthalten, für Zeichenkettenauswertungen kräftige Verknüpfungen.

In Zeichenkettenausdrücken generell können aber \$- und LIKE-Operator weiterhin nicht genutzt werden, was beispielsweise ein Test mit dem Ausdrucksfenster bestätigt. Dieses Dialogfenster ist kontextabhängig durch Mausklick zu öffnen. Die zahlreichen Dialogfenster vermitteln (lokal) weitere, im Vergleich mit den Designern einfachere visuelle Werkzeuge. Das aus verschiedenen Fenstern aufrufbare Ausdrucksfenster gestattet so die individuelle Eingabe jedweder Ausdruckskomponenten ebenso wie deren Übernahme durch Anklicken mit der Maus aus Listen, die auch auf die konkrete Arbeitssituation abstellen. Beim Aufruf während des Programmeditierens bricht allerdings die Visualisierung für das Feld-einfügen ab, wenn die entsprechende Datei vor dem Editieren nicht geöffnet wurde!?

#### 4. Objektorientierte Programmierung

Objektorientierung ist ebenso modern wie vielschichtig.<sup>8</sup> Bereits bei Beschränkung auf das hier interessierende Datenbanksystem werden Objekte aus Sicht der Datenmodellierung als die externen, abzubildenden Elemente betrachtet. Aus Sicht der in Visual dBASE möglichen objektorientierten Programmierung sind dagegen alle Elemente von Benutzeroberflä-

*"Crystal Reports"*

*Formular-Experte und  
Formular-Designer*

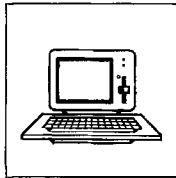
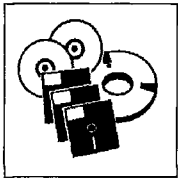
*Hilfreich: der Abfrage-Designer*

*Operatoren und Ausdrücke*

*Vielschichtiger Begriff:  
Objektorientierung*

<sup>7</sup> Dazu gehören auch formale Unzulänglichkeiten, so der fehlerhafte Installationshinweis in der Beilage zur Upgrade-CD-ROM, die Agenten anderer Anbieter vereinfachende Kritiken erleichtern.

<sup>8</sup> Zutreffend meint M. Wagner, der "Technologiebegriff Objektorientierung ist endgültig verwässert"; Computer Zeitung 8/96, 32.

*Klassen und Methoden*

*Objektorientierung: nur  
Vereinfachung der  
Softwareentwicklung?*

*Features des Programm-Editors*

chen Objekte. Programmiertechnisch umfaßt jedes Objekt eine Gruppe von Speichervariablen, wobei jede Variable eine Objekteigenschaft beschreibt. Damit ergibt sich auch gegenüber dBASE IV ein erweitertes und hinsichtlich der PRIVATE-Variablen geändertes Konzept.

Ein Ansatz objektorientierter Programmierung ist, nicht wie bei prozeduraler Programmierung vorzugeben, wie etwas ausgeführt werden soll, sondern nur, was geschehen soll. Die dazu erklärten, weitgehend selbständigen und abgeschlossenen Gebilde werden Objekte genannt und beschränken sich in Visual dBASE auf Datenbankoperationen einschließlich der dazu sinnvollen Nutzeraktivitäten. Da zahlreiche Objekte gleiche bzw. gleichartige Eigenschaften aufweisen, werden sie zu Klassen zusammengefaßt. Visual dBASE stellt dem Entwickler vorbereitete Klassen wie Eingabefelder, Formulare oder Fenster zur Verfügung, und es können aus ihnen weitere Klassen abgeleitet werden. Objekte sind damit Vertreter (Instanzen) der jeweiligen Klasse, und sie übernehmen die entsprechenden Eigenschaften. Zur Kommunikation mit seiner Umwelt (andere Objekte, Nutzer, Hardware) kann ein Objekt auf definierte Ereignisse (beispielsweise Mausklick oder Verhalten eines anderen Objekts) mit festzulegenden Methoden reagieren. Diese Methodenfestlegung geschieht mit einer Prozedur, quasi der Verbindung zwischen objektorientierter und bisheriger prozeduraler Programmierung.

Objektorientierte Technologien werden oft als Wege zu schnellerer und einfacher zu organisierender Softwareentwicklung angesehen, als Ausweg aus der sich vertiefenden Kluft zwischen immer leistungsfähigerer Hardware und nicht ausreichender Software. Es darf jedoch daran erinnert werden, daß ein historischer Ausgangspunkt der Objektorientierung auch die Vereinfachung des Mensch-Maschine-Dialogs war.<sup>9</sup> Wenn beispielsweise G. Braak fordert, "die interaktive IT-Unterstützung hat das weitgehende Fehlen wiederkehrender Arbeitsablaufstrukturen zu berücksichtigen, da die Qualität der staatsanwaltschaftlichen Tätigkeit unmittelbar mit dieser Individualität im Zusammenhang steht"<sup>10</sup>, so mag das manchem als Verzicht auf Informationstechnologien erscheinen. Erfordert aber nicht rechnerunterstützte juristische Entscheidung, wenn man über punktuelle Betrachtung hinausgeht, ein der Struktur jeweiliger Rechtsnormen und Normensysteme entsprechendes Verhältnis zwischen Automatisierung und juristischer Tätigkeit? Was die Strukturen betrifft, so können diese durchaus mittels relationaler Datenbanksysteme widergespiegelt werden. Bezüglich der Automatisierung wird bei rechnergestützter Rechtsanwendung nur auf den ersten Blick wie in anderen Bereichen vorgegangen. Selbstverständlich werden unter dem Aspekt der Rationalisierung Sequenzen des Entscheidungsprozesses automatisiert. Notwendige menschliche Tätigkeit ist hier aber nicht Überbleibsel, sondern iterative Voraussetzung partieller Automatisierung. Es gab in den 70er Jahren kluge Beiträge, die diese Voraussetzung bei anderer technischer und technologischer Umgebung hinterfragt hatten.<sup>11</sup> Aktuell dürfte sich die Frage nach geeigneter juristischer Software bei Objektorientierung auf die nach geeigneten Klassen, Objekten und Ereignissen verlagern. Visual dBASE bietet dazu mit den Formularen, den modalen und den nicht-modalen Fenstern Antworten an.

Angedeutet sei noch, daß der in Visual dBASE verfügbare Programm-Editor neben herkömmlichen Editorfunktionen zusätzlichen Service offeriert, der Codierung und Test weiter erleichtert. Dazu gehören Optionen zur automatischen Farbgebung (Variablenzuweisungen blau; unvollständige bzw. fehlerhafte Ausdrücke rot; Kommentare grau; restlicher Programmtext schwarz), zum automatischen Zeileneinzug im Umfang vorher erfolgter Einrückung, zum Einfügen von Ausdrücken mittels des bereits erwähnten Ausdrucksfensters, zum Syntaxeinfügen aus dem Hilfesystem und zum abschnittweisen Ausführen von Programmcode. Unverständlich ist, daß die Zeilennummern nicht angezeigt werden, auch wenn syntaktische und erkannte semantische Fehler wahlweise zur sofortigen Korrekturmöglichkeit führen.

<sup>9</sup> S. etwa Kempel/Pfander, Praxis der objektorientierten Programmierung, München/Wien 1990, 8.

<sup>10</sup> jur-pc 1/96, 14.

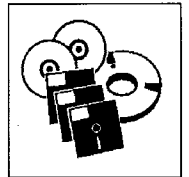
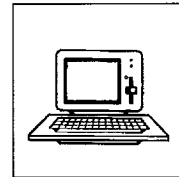
<sup>11</sup> Quellenangaben zu den Arbeiten von D. Suhr, L. Reisinger, W. Popp und B. Schlink s. etwa sub Fußnote 6.

## 5. Textverarbeitung

Juristische Arbeit ist (auch) Arbeit mit Texten. Software zur Textverarbeitung gehört daher unzweifelhaft zu entsprechender Rechnerunterstützung. Wird diese nun, selbst bei Einbeziehung externer Kommunikation, als einziges informatives Hilfsmittel eingeordnet<sup>12</sup>, so bleiben Potenzen zur Strukturierung und zum Dialog ungenutzt. Selbstverständlich müssen diese Potenzen im juristischen Bereich Texte einbeziehen. Und in Art und Umfang der Einbeziehung liegen die hauptsächlichen Probleme der dBASE-Nutzung für die juristische Arbeit.

Interessanterweise lassen sich Texte sowohl als externe Objekte im Sinne der Datenmodellierung wie als (interne) Objekte im Sinne der eben skizzierten objektorientierten Programmierung einordnen. Den externen Objekten wird mit den Memo- und auch den neuen OLE-Datenfeldern entsprochen. Die Klasse TEXT hingegen erlaubt in Visual dBASE nutzerdefinierte Textfestlegung bzw. Formularverweisung. Zwar kann für Memofelder ein einfacher Editor mit den gängigen Windows-Technologien zur Textbearbeitung (Ausschneiden, Kopieren, Einfügen, Löschen, Datei einfügen, Abschnitt speichern, Suchen, Konvertieren, Zeilen verbinden) genutzt werden. Zwar stehen für die Erreichbarkeit und Modifikation aller Textelemente in Memofeldern die von dBASE IV bekannten, unwesentlich erweiterten Funktionen der Befehlssprache und die sinnvollen neuen Befehle STORE MEMO... TO ARRAY... und REPLACE MEMO... WITH ARRAY... zur Verfügung. Zwar können TEXT-, DDE- und weitere Klassen herangezogen werden. Immer muß der Nutzer bei diesen Hilfsmitteln jedoch mehr tun, als vom Standpunkt möglicher Automatisierung und möglichen Services erforderlich ist.

Visual dBASE zeichnet sich durch ein sehr gutes Preis-/Leistungsverhältnis aus; eine Einzelplatzversion ist bereits für weniger als 400,- DM zu erhalten. Den für die juristische Arbeit wichtigen Komponenten steht jedoch eine unvollkommen vorbereitete Textverarbeitung gegenüber. Nimmt man letztere in Kauf oder erst einmal zu entwickelnde Formulare, weitere Klassen und Prozeduren entgegen, so bietet Visual dBASE wertvolle Unterstützung für Datenbanken zu interessierenden Klienten, Vertragspartnern, Verträgen oder diesen übergeordneten Rechtsformen, für Autorensysteme, zur Vertragsgestaltung, zur Unterstützung der Vertragserfüllung, für individuelle Recherchesysteme u. a. Und Visual dBASE wäre ein Versuch des Anbieters zur Qualifizierung der Textverarbeitung (für Juristen) wert.



*Juristische Arbeit = Text-Arbeit*

*Text-Funktionen noch unzureichend*

*Fazit*

<sup>12</sup> Eine entsprechende Konzentration findet man beispielsweise in Müller/Schallbruch, PC-Ratgeber für Juristen: Textverarbeitung, Datenbanken, Internet, Berlin/New York 1995.