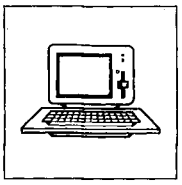
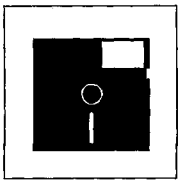



Berechnungsprogramme für Juristen mit Microsoft Windows 3.0

Volker Nilgens

Teil 3: Erstellung und Verwendung von DLLs (Dynamic-Link-Library) am Beispiel der Berechnung der Hebegebühr

Im Mittelpunkt des nachfolgenden Beitrags zur Windows-Programmierung werden die Erstellung und Verwendung einer sogenannten Dynamischen Linkbibliothek (DLL = Dynamic-Link-Library) stehen. Die wesentlichen Elemente einer DLL werden anhand eines Beispieles zur Berechnung der Hebegebühr aufgezeigt.
Im Anschluß hieran werden die Einsprungpunkte der Datei 'GEB.DLL', die in der jur-pc Mailbox zum Download bereitgehalten wird, offengelegt. Mit Hilfe dieser DLL kann jeweils die volle Gebühr nach §§ 11, 22, 123 BRAGO, § 11 Abs. 2 GKG, §§ 12 Abs. 2, 12 Abs. 3 ArbGG, § 32 KostO und § 13 GvKostG abhängig von dem übergebenen Wert berechnet werden.

Was sind DLLs?

DLLs sind ausführbare Bibliotheksmodule, die Programmcode, Ressourcen und/oder Daten enthalten können. Sie können während der Laufzeit in den Arbeitsspeicher eines Computers ein- und ausgelagert werden. Die Möglichkeit einer gemeinsamen Verwendung von mehreren Windows-Anwendungen verringert den erforderlichen Speicherplatz.

Fertige DLLs können, unabhängig von der Programmiersprache, in der sie erstellt wurden, von unterschiedlichen Programmiersprachen eingebunden werden. Die „Funktionseinheit" DLL kann, ohne eine erneute Kompilierung des aufrufenden Hauptprogramms erforderlich zu machen, geändert werden.

Unterschiede zwischen DLL und gewöhnlichem Programm

Der Quellcode einer mit Turbo-Pascal für Windows (TPW) erstellten DLL unterscheidet sich von einem gewöhnlichen Programm zunächst in der Kopfzeile durch das Word 'LIBRARY' anstelle des Wortes 'PROGRAM'. Hierdurch wird der Kompiler angewiesen, eine ausführbare Datei mit der Endung '.DLL' zu erstellen.

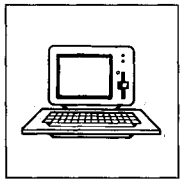
Funktionen und Prozeduren einer DLL müssen durch eine Export-Deklaration auf den Export vorbereitet werden. Die nachfolgende Anweisung 'exports' bewirkt letztendlich die gewünschte Bereitstellung der Einzelroutinen einer DLL.

Nachfolgend ist beispielhaft der Quelltext einer DLL zur Berechnung der Hebegebühr wiedergegeben.

Kopfzeile einer DLL

```
LIBRARY HebGeb;                                          { Kopfzeile einer DLL }
function HebeGeb(Wert: Real): Real; EXPORT;              { Export-Deklaration }
var
  tmp: Real;
begin
  if (WERT > 20000) then
    tmp := (((WERT - 20000) * 0.25 / 100) + 125)
  else
  begin
    if ((WERT <= 20000) and (WERT > 5000)) then
      tmp := (((WERT - 5000) * 0.5 / 100) + 50)
    else
      tmp := WERT * 0.01;
  end;
  HebeGeb := Tmp;
end;
exports                                                  { exports-Anweisung }
  HebeGeb  index 1;
BEGIN
END.
```

Volker Nilgens arbeitet als wissenschaftlicher Mitarbeiter am rechtswissenschaftlichen Seminar der Universität Köln und studiert Informatik. Einigen Lesern ist er durch seine Programme für Juristen bekannt, die auch in der jur-pc Mailbox zum Download bereitliegen.



Die Einbindung einer DLL

Die Einbindung der vorstehenden DLL-Funktion könnte in einem selbst erstellten TPW-Programm durch die Deklaration

```
function HebeGeb(Wert: Real): Real; far; external 'HEBGEB' index 1;
```

oder unter Verwendung einer speziellen Unit erfolgen. Die Einbindung mittels einer speziellen Unit wird nachfolgend zur Offenlegung der Einsprungpunkte der Datei 'GEB.DLL' dargestellt. Eine Unit zur Einbindung der in der Mailbox bereitgehaltenen Berechnungs-Unit könnte wie folgt aussehen:

```
{*****************************************}
{                                         }
{  Geb-DLL interface unit                 }
{  Copyright (c) 1992 by V. Nilgens, Köln }
{                                         }
{*****************************************}

UNIT GebDLL;

INTERFACE

function GerGeb    (Wert: Real): Real;          { § 11  Abs. 2 GKG     }
function AnwGeb    (Wert: Real): Real;          { § 11         BRAGO   }
function BeiGeb    (Wert: Real): Real;          { § 123        BRAGO   }
function GVGeb     (Wert: Real): Real;          { § 13         GvKostO }
function ArbGGeb   (Wert: Real): Real;          { § 12  Abs. 2 ArbGG   }
function FGGGeb    (Wert: Real): Real;          { § 32         KostO   }
function HebeGeb   (Wert: Real): Real;          { § 22         BRAGO   }
function LArbGGeb  (Wert: Real): Real;          { § 12  Abs. 3 ArbGG   }


function GerGebPtr   (Wert: Double): Pointer;   { Datentypen für einen  }
function AnwGebPtr   (Wert: Double): Pointer;   { Zugriff auf die DLL   }
function BeiGebPtr   (Wert: Double): Pointer;   { mit Object Vision.    }
function GVGebPtr    (Wert: Double): Pointer;
function ArbGGebPtr  (Wert: Double): Pointer;
function FGGGebPtr   (Wert: Double): Pointer;
function HebeGebPtr  (Wert: Double): Pointer;
function LArbGGebPtr (Wert: Double): Pointer;


IMPLEMENTATION

function GerGeb;   external 'GEB' index 1;
function AnwGeb;   external 'GEB' index 2;
function BeiGeb;   external 'GEB' index 3;
function GVGeb;    external 'GEB' index 4;
function ArbGGeb;  external 'GEB' index 5;
function FGGGeb;   external 'GEB' index 6;
function HebeGeb;  external 'GEB' index 7;
function LArbGGeb; external 'GEB' index 8;


function GerGebPtr;    external 'GEB' index 9;
function AnwGebPtr;    external 'GEB' index 10;
function BeiGebPtr;    external 'GEB' index 11;
function GVGebPtr;     external 'GEB' index 12;
function ArbGGebPtr;   external 'GEB' index 13;
function FGGGebPtr;    external 'GEB' index 14;
function HebeGebPtr;   external 'GEB' index 15;
function LArbGGebPtr;  external 'GEB' index 16;

END.
```

Mit der Einbindung der vorstehenden Unit – durch eine USES-Anweisung – wäre es möglich, ein Programm unter Verwendung der angegebenen Berechnungs-DLL zu erstellen, ohne die eigentlichen Berechnungen programmieren zu müssen.
Wie unterschiedlich derartige Windows-Programme aussehen können, wird deutlich, wenn man sich die beiden in der jur-pc Mailbox abgespeicherten Windows-Programme 'GEBUEHR.EXE' und 'TABELLE.EXE', die beide 'GEB.DLL' erfordern, betrachtet.