

### 3. Das erste Programm

Nach diesen langwierigen Vorbereitungen beschäftigt sich der folgende Abschnitt damit, wie man ein Programm schreibt. Eine Programmiersprache ist vom Aufbau her im Prinzip so ähnlich gestaltet wie jede gesprochene Sprache auch. Der wesentliche Unterschied ergibt sich aus der Tatsache, daß der Sprachschatz einer Programmiersprache weitaus geringer ist als der einer richtigen Sprache.

Zum Vergleich: Ein durchschnittlicher Erwachsener hat beim Sprechen ca. 30.000 Wörter zur Verfügung, aus denen er auswählen kann. Demgegenüber gibt es in Borland Pascal gerade einmal 51 Wörter, die die gesamte Sprache bilden. Der Wortschatz einer Programmiersprache wird von deren Erfindern nicht ohne Grund so klein gehalten. Schließlich soll die Programmiersprache einfach und schnell erlernbar sein. Kritiker werden jetzt anmerken, daß es doch unmöglich sein muß, mit nur 51 Wörtern ein Programm zu schreiben, welches beliebige Aufgaben erfüllen soll.

Sie werden jedoch bald an praktischen Beispielen erfahren, daß es nicht auf die absolute Anzahl der Wörter einer Programmiersprache ankommt, sondern vielmehr auf die Kombinationsmöglichkeiten der einzelnen Worte untereinander. Dadurch ist es nämlich möglich, nahezu jede Aufgabe in eine Programmiersprache zu übersetzen.

Doch was passiert nun, wenn Anweisungen und Befehle in der Sprache Pascal erstellt worden sind, und diese auf dem Rechner als sogenanntes Programm ausgeführt werden sollen? Hier kommt der Compiler (Übersetzer) ins Spiel. Er übersetzt die Anweisungen, die Sie in Pascal definiert haben, in Befehle, die der Computer bzw. das Betriebssystem versteht und ausführen kann. Solche Befehle, die der Computer direkt ausführen kann, nennt man auch Maschinensprache-Befehle. Diese für Computer direkt lesbare Sprache ist jedoch für uns Programmierer weitaus schlechter zu handhaben als Borland Pascal. Besteht sie doch nur aus Nullen und Einsen. Bleiben wir also lieber bei den vertrauteren Gefilden, den Wörtern und Zahlen bei Borland Pascal, und den Mindestanforderungen, die ein Programm in dieser Sprache erfüllen muß. Das hierfür mehrere Zeilen erforderlich sind, dafür spricht schon die Überschrift zum nächsten Kapitel.

#### 3.1 Die Struktur, ein Vierzeiler muß sein

Jede Programmiersprache hat seine eigenen Konventionen, was den Aufbau eines Programms angeht. Borland Pascal ist da keine Ausnahme. Es sind einfache Vorarbeiten, die wohl jede komplexere Handlung in unserem Leben einleiten und nach ein paar Wiederholungen als selbstverständlich gelten. Mindestens vier Ausdrücke sind notwendig, die teils vor, teils hinter die eigentlichen Programmanweisungen geschrieben werden müssen.

Der Vierzeiler beinhaltet mehrere in Großschrift gehaltener Wörter (PROGRAM, USES, BEGIN und END). Diese Wörter gehören zum Sprachschatz von Turbo Pascal und haben deshalb eine bestimmte Bedeutung.

Ein geschriebenes Programm ganz ohne diesen "Rahmen" würde nicht funktionieren. Hier hilft also nur eins: die Worte samt Reihenfolge auswendig lernen! Begonnen wird mit dem Wort PROGRAM. Dieses dient als Rahmen, aus dem der minimal notwendige Pascal-Programmcod erzeugt wird. Mit Hilfe dieses Ausdrucks geben Sie dem Programm außerdem einen Namen. Neben reinen Fantasienamen und Namen von bekannten Fabelwesen gehen neuerdings immer mehr Programmierer/innen dazu über, hier einen Ausdruck einzusetzen, der den Sinn und Zweck des Programms beschreibt.

Haben Sie erst einmal mehrere Programme erstellt und diese mit den obligaten DOS-Kürzeln (anders kann man die acht Buchstaben, die einem zur Verfügung stehen, wohl nicht nennen) in Ihrem Quelltext-Verzeichnis abgelegt, so werden Sie diese Möglichkeit zur ausführlichen Programmbeschreibung wohl zu schätzen wissen.

Der Name des Programms darf nur aus einem zusammenhängenden Ausdruck bestehen. Möchte man dem Programm einen Titel aus mehreren Wörtern geben, so muß man diese anstatt durch Leerschritt mit dem Unterstreichen-Zeichen " \_ " voneinander trennen.

Als nächstes steht der Begriff USES auf dem Plan. Mit seiner Hilfe können Funktionsbibliotheken eingebunden werden. Es handelt sich hierbei um bereits programmierte und compilierte - also in Maschinensprache übersetzte - Routinen (Programmteile), die man in seinen Programmen einsetzen kann. Jeder Programmierer muß für sich das Rad also nicht neu erfinden, sondern kann durchaus auf die mitgelieferten Toolboxen wie Turbo Vision, Objekt Windows oder Toolboxen von Fremdanbietern

zurückgreifen.

Diese Funktionsbibliotheken bezeichnet man auch oft mit dem englischen Wort Unit.

Units gibt es für vielerlei Bereiche der Programmierung. Crt bzw. WinCrt (s. o.) sind beispielsweise für die Erleichterung der Textausgabe entwickelt worden. Sie werden im nächsten Abschnitt davon Gebrauch machen. Andere Units wie Dos bzw. WinDos erleichtern durch ihre einfach zu benutzenden Routinen die Arbeit mit Dateien, wie z. B. das Öffnen, Lesen oder Schreiben.

Borland Pascal stellt sogar noch viel mächtigere Routinen in Form von Units zur Verfügung. Mit diesen ist es auf vergleichsweise einfache Art möglich, mehrere Fenster in einem Programm gleichzeitig darzustellen oder Programme mit so komfortablen Elementen wie Rollbalken, Buttons oder einem Menüsystem auszustatten. Der beste Hinweis auf die Leistungsfähigkeit der bereitgestellten Units gibt Borland Pascal selber. Dessen IDE wurde mit genau denselben Hilfsmitteln erstellt, die nun auch Ihnen zur Verfügung stehen.

Haben Sie erst einmal eine Reihe von allgemein verwendbaren Routinen selbst geschrieben, so können Sie daraus ebenfalls eine Unit bilden, um diese in Ihren Programmen zu verwenden.

Kommen wir nun zum letzten Begriff dieses Beispielprogramms. Der Ausdruck Begriffspaar wäre an dieser Stelle allerdings besser angebracht. Tatsächlich treten die Wörter BEGIN und END stets paarweise auf, d. h., wo vorher ein BEGIN steht, muß hinterher auch ein END stehen. In Turbo Pascal wird dieses Begriffspaar benutzt, um den Beginn und das Ende des eigentlichen Programms zu markieren. Bitte beachten Sie, daß auf das END ein Punkt folgen muß. Dieses Programm einzutippen und auszuführen hat wenig Sinn. Da zwischen dem BEGIN und END nichts steht, gibt es natürlich auch nichts zu sehen. Das soll sich jedoch sofort ändern.