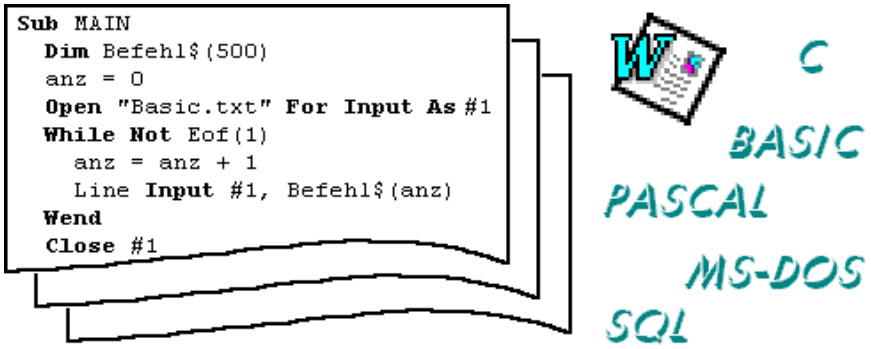
Thomas Angielsky • Alpenweg 5a • 89257 Illertissen •

Microsoft Word

Programm-Listings formatieren



Viele Programmiersprachen sind deshalb oft so schwer zu verstehen, weil ihre Programm-Listings nicht formatiert sind und dadurch ihre Befehlswörter nicht auffallen. Viele Entwicklungsoberflächen arbeiten deshalb bereits schon jetzt mit einer farbigen Darstellung von Befehlswörtern. Beim Ausgeben der Listings und Quellcodes auf einem Drucken verwenden die meisten jedoch nur einfache Textdarstellungen. Auch Microsoft Word (bis Version 95) macht da bei WordBasic keine Ausnahme.

Hier greifen die folgenden WordBasic Makros ein. Sie formatieren die Befehlswörter Ihres Programm-Listings nach Ihren Wünschen. Das besondere an den Makros ist, daß die Befehlswörter in einer Textdatei gespeichert und damit auf einfache Weise ausgetauscht werden können. Sie haben damit mit einem Makro ein Utility für alle Ihre Programm-Listings von Basic- bis SQL-Dialekten.

Beispiel: Sie möchten Ihre WordBasic Quelltexte formatieren. Erstellen Sie sich dazu die Datei `BASIC.TXT` mit folgendem Inhalt:

```
sub
end
function
while
wend
for
to
next

...
```

Wichtig ist nur, daß jedes Befehlswort in einer seperaten Zeile steht. Die zugehörigen Makros haben einen einfachen Aufbau. Die Bearbeitung erfolgt in nur zwei Schritten.

- Befehlswort aus Textdatei lesen
- Befehlswort im Text mit Formatierung ersetzen

Variante 1

Zuerst werden die Befehlswörter aus einer beliebigen Textdatei in den Speicher gelesen. Standardmäßig ist bei dieser Lösung Platz für 500 Befehlswörter.

```
Sub MAIN
Dim Befehl$(500)
anz=0
Open "BASIC.TXT" For Input As #1
```

Am günstigsten ist es, wenn Sie hier noch zusätzlich eine Pfadangabe zum Dateinamen machen:

```
Open "C:\WORD\BASIC.TXT" For Input As #1
```

Andernfalls müssen Sie immer vorher das Verzeichnis setzen (Datei öffnen - Verzeichnis auswählen - Abbrechen), daß die Textdatei enthält.

```
While Not Eof(1)
 anz=anz+1
 Line Input #1, Befehl$(anz)
Wend
Close #1
```

Ab hier beginnt die Formatierung des Programm-Listings. Jedes Wort wird darauf geprüft, ob es ein Befehlswort ist. Verantwortlich für das Auslesen jedes einzelnen Wortes sind die WordBasic-Funktionen `WordRechs` und `Markierung$`. Zeichen wie “`.,:;`” werden ausgeschlossen, um die Bearbeitungsgeschwindigkeit zu steigern.

```
BearbeitenAllesMarkieren
BearbeitenTextmarke "Marke", .Hinzufügen
MarkierungArt 1
While TextmarkenVergleichen("\Sel", "Marke")=6 Or TextmarkenVergleichen("\Sel", "Marke")=8
 MarkierungAktuellWort
 a$=UCase$(Markierung$())
 If Not(a$="." Or a$="," Or a$=":" Or a$=";") Then
   For i=1 To anz
     If a$=UCase$(Befehl$(i)) Then
```

In diesem Abschnitt können Sie das Format des Befehlswortes festlegen, in diesem Beispiel wird die Schriftart Courier New-fett verwendet.

```
       Fett 1
```

```
       Schriftart "Courier New",10
       Einfügen Befehl$(i)
       i=anz

     EndIf
   Next i
 EndIf
 WortRechts
Wend
BearbeitenGeheZu "Marke"
BearbeitenTextmarke "Marke", .Löschen
End Sub
```

Variante 2

Die zweite Lösung hat einen noch einfacheren Aufbau. Anstatt den Text “manuell” per Makro nach Befehlswörtern zu durchsuchen, verwendet diese Variante die Suchen und Ersetzen Funktion von Word.

```
Sub MAIN
Open "Basic.txt" For Input As #1
 While Not Eof(1)
   Line Input #1, st$
   BearbeitenErsetzenZeichen .Punkt = "12", .Unterstrichen = - 1, .Farbe = - 1, .Durchstreichen = -
1, .Hochgestellt = - 1, .Tiefgestellt = - 1, .Verborgen = - 1, .Kapitälchen = - 1, .Großbuchstaben = -
1, .Laufweite = "", .Position = "", .Unterschneidung = - 1, .UnterschneidungMin = "", .Registerkarte = "0",
.Schriftart = "Courier New", .Fett = 1, .Kursiv = - 1, .Gliederung = - 1, .schattiert = - 1
   BearbeitenErsetzen .Suchen = st$, .Ersetzen = st$, .Richtung = 0, .GroßKleinschreibung = 0, .GanzesWort
= 1, .Mustervergleich = 0, .Reserviert23 = 0, .AllesErsetzen, .Format = 1, .Textfluß = 1, .Reserviert31 = 0
   Wend
 Close #1
End Sub
```

Einsatzgebiete der beiden Varianten

In Punkto Geschwindigkeit ist die 2. Variante der ersten in nahezu jedem Einsatzfall überlegen. Die 1. Variante spielt Ihre Stärken dann aus, wenn besondere Formatierungen nötig sind. Wenn Ihnen also die normale Ersetzen-Funktion von Word ausreicht, sollten Sie die 2. Variante einsetzen. Möchten Sie hingegen außergewöhnliche Formatierungen, brauchen Sie die frei programmierbare Variante 1.