

# Spezifikation für die Kommentierung von VBA-Modulen

Christoph Jüngling  
[www.Juengling-EDV.de](http://www.Juengling-EDV.de)

## 1 Allgemeines

In dieser Dokumentation wird der Begriff *Funktion* gleichrangig, teils an Stelle des Begriffes *Subroutine* verwendet.

### 1.1 Zeilendefinition

Eine Zeile im Sinne dieser Dokumentation entspricht einer Kommentarzeile in VBA. Diese wird grundsätzlich durch ein *Apostroph* eingeleitet. Die alternative Syntax `REM . . .` ist hier nicht zugelassen.

### 1.2 Folgezeilen

Folgezeilen in den für die Dokumentation relevanten Kommentaren sind durch die einleitende Sequenz `<Apostroph><Space><Space>` definiert. Sie werden von der analysierenden Software an die vorhergehende Zeile angehängt und zählen somit auch zu deren Rubrik.

### 1.3 Explizite Zeilenwechsel

Explizite Zeilenwechsel in der Dokumentation werden durch Verwendung eines Semikolons (;) am Zeilenende erzeugt. Sollte eine Folgezeile gefunden werden, wird diese nicht wie üblich an die vorherige angehängt, sondern ein Zeilenwechsel vorgenommen.

Das Semikolon ist somit von seiner üblichen syntaktischen Funktion ausgenommen und sollte nicht verwendet werden.

### 1.4 Kommentarzeilen, die ignoriert werden

Folgende Zeilenanfänge definieren reine Kommentarzeilen, die in die beschreibende Tabelle nicht übernommen werden:

- `' ==`
- `' --`
- `' <Space>**`

Andere Kommentarzeilen werden nur übernommen, wenn sie der Spezifikation nach 1.2 oder 2.1 entsprechen.

## 2 Funktionskopf

Zum Funktionskopf im Sinne dieser Dokumentation zählen:

- der programmtechnische Funktionskopf  

```
[Private|Public] (Function|Sub|Property) \(...
```
- ein führender Kommentar mit Gleichheitszeichen
- mehrere Rubriken, die spezielle Merkmale der Funktion beschreiben
- ein abschließender Kommentar mit Gleichheitszeichen

Auf diesen so definierten Funktionskopf folgt der Deklarationsteil, sodann der Anweisungsteil, der durch ein

```
End (Function|Sub|Property)
```

abgeschlossen wird.

```
1 '=====
2 Public Function DatensatzKopieren (ActForm As Form) As Integer
3 ' Aufgabe:   Aktueller Datensatz wird gespeichert
4 ' Parameter: ActForm: Formular, mit dem gearbeitet wird
5 ' Bemerkung: Der Datensatz befindet sich danach noch in Bearbeitung
6 ' Autor:     Jüngling
7 '=====
```

Listing 1: Funktionskopf

Dem Funktionskopf wird eine Kommentarzeile aus Gleichheitszeichen (=) vorangestellt (s. Listing 1, Zeile 1), um eine leichtere Lesbarkeit im Programmlisting zu erreichen.

Der eigentliche Funktionskopf (s. Listing 1, Zeile 2) entspricht zwangsläufig der VBA-Syntax. Soweit ein Rückgabewert der Funktion definiert und verwendet wird, erfolgt grundsätzlich eine Typisierung. Parameter werden ebenfalls grundsätzlich typisiert.

### 2.1 Syntax der Rubriken im Funktionskopf

Eingeleitet von der Sequenz <Apostroph><Space> folgt eine beliebige Bezeichnung, die durch einen Doppelpunkt (:) abgeschlossen wird. Diese Bezeichnung wird als *Rubrik* verwendet, die nachfolgenden Informationen als *Information*.

Beispiel:

```
' Autor:      Jüngling
```

## 2.2 Besonderheiten für die Rubriken „Parameter“ und „Ergebnis“

```
' Parameter: ActForm: Formular, das bearbeitet werden soll;  
'           Check:   wird binär ausgewertet (siehe CONST-Deklarationen)
```

In diesen Fällen ist eine zusätzliche Unterrubrik notwendig, da für jeden Parameter eine gesonderte Beschreibung erfolgen muß. Auch die Ergebnisse müssen dergestalt unterteilt werden, daß z.B. TRUE und FALSE beschrieben werden können.

Die Auswertungssoftware erfasst diese Unterrubriken jedoch als normalen zusammengefassten Kommentartext. Explizite Zeilenwechsel werden wie anderweitig beschrieben berücksichtigt.

## 3 Funktionsrumpf

### 3.1 Ablaufbeschreibung im Funktionsrumpf

```
1 ' *****  
2 ' 1: Datensatz speichern, falls nötig;  
3 '   Nach einem Doppelklick auf DS-Zeiger ist das zwar nicht  
4 '   erforderlich, aber es wird auch noch einen Menübefehl und evtl.  
5 '   ein Symbol oder einen Tastenschlüssel für diese Funktion geben ...  
6 ' *****
```

Listing 2: Beschreibung im Rumpf

Im Funktionsrumpf können in gleicher Weise weitere Beschreibungen vorgenommen werden, wie dies im Funktionskopf möglich ist.

Eine gegenüber den Rubriken im Funktionskopf zusätzliche Ablaufbeschreibung im Anweisungsteil ist durch die Verwendung einer „numerischen Rubrik“ möglich (s. Listing 2, Zeile 2). Folgezeilen sind wie in 1.2 beschrieben zu behandeln. Wird eine derartige Beschreibung gefunden, so wird sie in der Dokumentation der Rubrik *Ablauf* zugeschlagen, sofern diese existiert (andernfalls wird eine solche Rubrik generiert).

### 3.2 Abschlusscode

Falls eine Art Abschlusscode für die Funktion erforderlich ist, wird hierfür das Label `Ausgang:` verwendet.

### 3.3 Fehlerbehandlung

Fehlerbehandlungsroutinen werden am Ende des Funktionsrumpfes „gesammelt“ und von dem restlichen Code durch eine Kommentarzeile abgetrennt:

'-----

Die für diese Routinen verwendeten Labels tragen grundsätzlich die Bezeichnung `Fehler:`, ggf. gefolgt von einer laufenden Nummer:

`Fehler:`