ÜBERSICHTSBLATT:
PROZEDUREN UND FUNKTION IN VBA

Hintergrund: Modulare Programmierung und Wiederverwendbarkeit

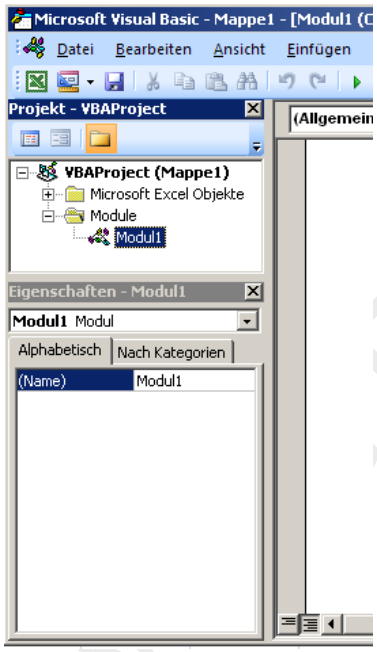
Wenn man ein Programm schreibt, so tut man dies in der Regel um eine Aufgabe, die häufig anfällt, zu automatisieren. Das Programm wird dann immer wieder ausgeführt. Wenn man viele Programme schreibt, dann bemerkt man schnell, dass es gewisse (Teil-) Aufgaben gibt, die man immer wieder automatisiert.

Es bietet sich also an den Grundgedanken der Programmierung – die Erstellung von wiederverwendbaren Automatisierungen – auch auf die Programme selbst anzuwenden. Dazu zerlegt man die Programme in Teile (man spricht hierbei oft von Modulen) die jeweils eine Bestimmte Aufgabe erfüllen. Diese Module kann man dann immer wieder verwenden.

Beispiel:

Wir haben bereits die Funktion SVERWEIS aus Excel kennen gelernt. Wann immer wir diese Funktion in einer Excel-Tabelle verwenden, um automatisiert nach einem Bestimmten Datensatz in einer Tabelle zu suchen, wird im Hintergrund ein Programm-Modul ausgeführt. Tatsächlich kann man dieses Modul auch in VBA-Verwenden.

Module in VBA



Wenn wir in VBA von einem Modul sprechen, so meinen wir damit eine Sammlung von Prozeduren und/oder Funktionen. Jedes VBA-Projekt (d. h. jede Arbeitsmappe) kann mehrere Module enthalten. Es ist so z.B. möglich ein Modul speziell für Dateioperationen ein weiteres für Zugriffe auf Datenbanken und vielleicht noch eins für die Ein- und Ausgabe im Userkontext zu schreiben.

Je nach dem mit welcher Sichtbarkeitseinstellung die Prozeduren und Funktionen innerhalb des Moduls angelegt wurden kann man entweder von außen oder nur innerhalb des Moduls auf sie zugreifen. Die Schlüsselworte um dies zu steuern heißen `private` (dann kann man nur innerhalb des Moduls darauf zugreifen) und `public` (dann ist der Zugriff auch von außen möglich).

Dadurch ergibt sich die Möglichkeit innerhalb eines Moduls Hilfsfunktionen zu schreiben, die von außen (aus dem Hauptprogramm) nicht angesprochen werden können.

Prozeduren

Eine Prozedur ist ein Programmabschnitt der als logische Einheit aufgerufen werden kann. Prozeduren sind arbeiten dabei immer eine gewisse Funktionalität ab, ohne dabei ein konkretes Ergebnis zu liefern. Sie können als Arbeitsschritte verstanden werden. Sie werden z.B. verwendet wenn es darum geht etwas in eine Datei zu schreiben, oder einem Benutzer seine Nachricht auszugeben. Die Syntax zum erstellen einer Prozedur ist:

```
<public/private> sub <prozedurName>([<parameter1> As <Datentyp>])
      'Toller Code
end sub
```

Um eine solche Prozedur aufzurufen bedient man sich dann der folgenden Syntax:

```
prozedurName parameter1
```

www.eTeachers.de

© Marco Ranzinger

ÜBERSICHTSBLATT:
PROZEDUREN UND FUNKTION IN VBA

Funktionen

Eine Funktion ist, ähnlich wie eine Prozedur, auch eine Logische Sammlung von Anweisungen die über einen Namen aufgerufen werden kann. Der Unterschied besteht im Wesentlichen darin, dass eine Funktion auch ein Ergebnis liefert. D. h. Funktionen haben einen Rückgabewert den man auswerten kann. Darum werden Funktionen oft verwendet um Teilaufgaben wie komplizierte Berechnungen, das Einlesen von Daten oder Abfragen von Eingaben des Users auszulagern.

Die Syntax zur Erstellung einer Funktion sieht dabei folgendermaßen aus:

```
<public/private> function <functionName>([<parameter1> As <Datentyp>])
      ' Toller Code
      ' und dann wird das Ergebnis zurückgegeben
      functionName = ergebnis
end sub
```

Um die Funktion dann im Code aufzurufen schreibt man:

```
ergebnis = functionName(parameter1)
```

Ein paar Anmerkungen zum guten Stiel

Alleine das Anlegen von Prozeduren und Funktionen um Teilausgaben auszulagern kann bereits als ein Schritt in die richtige Richtung betrachtet werden. Denn diese Modularisierung des Codes erhöht drastisch dessen Wartbarkeit. Um dieses Ziel konsequent verfolgen zu können, muss man sich allerdings auch über die Art und Weise Gedanken machen, wie man Prozeduren und Funktionen schreibt.

Generell sollten solche Programmteile nie zu lang werden. Am besten kann man eine Prozedur lesen wenn sie aus höchsten 15 Zeilen Code besteht (Deklaration von Variablen und Fehlerbehandlung würde ich hier allerdings noch nicht mitzählen).

Auch mit Kommentaren sollte man nicht sparsam sein. Vor jeder Prozedur und vor jeder Funktion sollte mindestens die Folgenden Informationen in einem strategischen Kommentar zu finden sein:

- Eine verbale Beschreibung was das innerhalb der Prozedur passiert und wofür sie gut ist
- Welche Parameter werden von der Prozedur erwartet und wofür werden diese dann in der Prozedur verwendet.
- Bei Funktionen sollte immer angegeben werden, was die jeweilige Funktion zurückgibt.
- Wenn mehrere Entwickler an einem Projekt arbeiten bietet es sich außerdem an, anzugeben wer die Prozedur geschrieben hat. Außerdem sollte bei jeder Änderung ein entsprechender Vermerk hinzugefügt werden (Wer hat wann was geändert?).

www.eTeachers.de

© Marco Rancinger