

Frage deutsch

~~~~~

Was sind MAK-Module und wie gehe ich mit ihnen um?

## Question English

~~~~~

What are MAK-modules good for and how do I handle them?
How to split my program into modules?

Antwort 1

~~~~~

[ von Christoph Borschensky ("Hawk") und Thomas Antoni, 1.8.2003 - 30.11.2005 ]

### \*\*\* Was ist ein Modul?

Ein Modul bei QuickBASIC ist eine Quellsprache-Programmdatei, die auch SUBs und FUNCTIONS enthalten kann. Man kann mit QuickBASIC 4.5 zwei oder mehr Programme bzw. Module laden und gemeinsam zu einer EXE-Datei kompilieren.

MAK-Module werden nur von QuickBASIC unterstützt, nicht jedoch von QBasic. Wenn Du Dein Programm mit QBasic in mehrere Dateien aufteilen willst, dann bleibt Dir nur die wesentliche unkomfortablere Methode der Programmverkettung mit CHAIN und RUN (siehe den FAQ-Eintrag "[Wenn der Speicher knapp wird -> Wie teile ich mein Programm mit CHAIN auf 2 Dateien auf?](#)" ).

Modularisierte Programme enthalten immer ein Hauptmodul und ein oder mehrere Untermodule. Das Hauptmodul ist das Modul, welches bei "Ausführen | Start" wirklich ausgeführt wird. Im Normalfall enthält nur das Hauptmodul auf der "obersten Ebene" ausführbare Befehle. Die Untermodule bestehen üblicherweise nur aus einer oder mehrerer SUBs und FUNCTIONS, die vom Hauptmodul oder auch untereinander aufgerufen werden.

Auf der obersten Modulebene enthalten die Untermodule normalerweise keine ausführbaren Befehle, sondern nur

- • die Deklarationen der enthaltenen SUBs/FUNCTIONs
- • Deklarationen für die globalen Daten mit DIM, SHARED und COMMON
- • Deklarationen anwenderdefinierter Felder mit TYPE...END TYPE
- • Metabefehle wie '\$STATIC, '\$DYNAMIC usw.
- • Fehlerbehandlungs- und Ereignisroutinen, die mit ON...GOSUB/GOTO angesprungen werden

Die Verwendung von Modulen hat zwei Vorteile:

- • Man kann ausgetestete Standardroutinen in SUBs und FUNCTIONS in Untermodule hinterlegen und für mehrere Softwareprojekte verwenden - ähnlich wie bei Bibliotheken.

- Mit Modulen kann man die Speichergrenze von 64 KB pro EXE-Programm auf maximal 100...250 KB noch oben verschieben (siehe unten).

### \*\*\* Wie lädt und verbindet man innerhalb eines Programms mehrere Module?

Man lädt wie gewöhnlich ein Programm über "Datei | Programm laden..." und fügt das zweite Modul über "Datei | Datei laden..." -> Laden als: <Modul>" oder "Datei | Datei erstellen... -> als: <Modul>" hinzu (dies kann man beliebig wiederholen, um viele Module zu laden). Speichert man jetzt ab, wird eine MAK-Datei mit den Dateinamen der beiden Module zur Erinnerung für QB erstellt.

### \*\*\*Wie bestimmt man das Hauptmodul ?

Falls mehrere Module geladen sind, kann man das Hauptmodul unter "Ausführen | Hauptmodul bestimmen..." auswählen. Die MAK-Datei erhält als vorderen namensbestandteil immer den Namen der Hauptmodul-Datei.

### \*\*\* Können SUBS/FUNCTIONs gemeinsam benutzt werden?

SUBs und FUNCTIONs sind global und können von anderen Modulen benutzt werden. Deshalb kann man auch keine zwei Module laden, die gleichnamige Prozeduren haben. In allen aufrufenden Modulen müssen die Subroutinen mit DECLARE SUB(...) und die Funktionen mit DECLARE FUNCTION(...) deklariert (d.h. bekannt gemacht) werden.

### \*\*\* Können Variablen und Felder gemeinsam (global) benutzt werden?

Auf Variablen und Felder kann über Module hinweg zugegriffen werden. Die Variablen und Felder sind dann in allen Modulen auf "oberster Modulebene" mit COMMON SHARED oder DIM SHARED zu deklarieren.

### \*\*\* Welche Rolle spielen die MAK-Dateien?

Speichert man ein Quellspracheprogramm ab, wenn zwei oder mehrere Module geladen sind, wird automatisch eine .MAK- Datei mit den Dateinamen des Hauptmoduls erstellt, also desjenigen Moduls, das bei "Ausführen | Start" ausgeführt wird. Auch in der MAK-Datei wird die Datei zuerst genannt, die das Hauptmodul ist. Hat man also z.B. MODUL\_1.BAS mit MODUL\_2.BAS zusammengebunden, so steht in der MODUL\_1.MAK-Datei bzw. MODUL\_2.MAK-Datei:

```
MODUL_1.BAS  
MODUL_2.BAS
```

oder

```
MODUL_2.BAS  
MODUL_1.BAS
```

Die Reihenfolge wird durch das Hauptmodul bestimmt; im ersten Fall ist MODUL\_1.BAS das Hauptmodul, im zweiten MODUL\_2.BAS.

Lädt man nun die BAS-Datei, nach der die MAK-Datei benannt ist (bei MODUL\_1.MAK wäre das MODUL\_1.BAS), dann werden beide Dateien als Module geladen. Unter "Ansicht | SUBs..." kann man sie betrachten. Sie sind die, die IMMER großgeschrieben und nicht wie die SUBs und die FUNCTIONS eingerückt sind. Würde man diese MAK-Datei löschen, werden die Dateien wieder einzeln geladen.

### \*\*\* Wie groß kann ein EXE-Programm, das aus Modulen besteht, maximal sein?

Bei der Erstellung des EXE-Programms werden automatisch alle Module zusammengebunden. Den QuickBASIC-Linker benötigt man dafür nicht, im Gegensatz zu den Bibliotheken. Jedes Modul darf max. 64 KB Maschinencode umfassen. Für das gesamte EXE-Programm besteht eine Speichergrenze von 100...250 KB. Der größere Wert gilt bei einer Compilierung mit dem externen Kommandozeilen-Compiler BC.EXE statt innerhalb der Entwicklungsumgebung QB.EXE. Jedes der Module darf aber trotzdem nie zu einem ausführbaren Maschinencode von über 64 KB führen.

### \*\*\* Beispiel

Im Folgenden ein einfaches Beispiel für die Verwendung von zwei Modulen, dem Hauptmodul HAUPTMOD.BAS und dem Untermodul UNTERMOD.BAS. Die Einzelheiten kannst Du den Kommentar entnehmen

```
'*****
' HAUPTMOD.BAS = Demonstriert die Verwendung von MAK-Modulen in QuickBASIC
' =====
' Die Programmmodul-Dateien HAUPTMOD.BAS und UNTERMOD.BAS demonstrieren
' die Verwendung von Modulen in QuickBASIC (funktioniert nicht mit QBasic).
' HAUPTMOD.BAS ermittelt die Potenzen 1^2...10^2 und ruft dazu die
' FUNCTION Quadrat() auf, die sich im Untermodul UNTERMOD.BAS befindet
'
' Das erstmalige Laden der beiden Module in die QuickBASIC-Entwicklungs-
' Umgebung erfolgt so:
' - HAUPTMOD.BAS laden ueber "Datei | Programm laden"
' - HAUPTMOD.BAS als Hauptmodul definieren ueber
'   "Ausfuehren | Hauptmodul bestimmen
' - UNTERMOD.BAS laden ueber "Datei | Datei laden"
'
' Die einzelnen Module, SUBs und FUNCTION lassen sich durchblaettern
' ueber "Ansicht | Subs". Die Modul-Dateien koennen mit "Datei | Alles
' speichern" gemeinsam abgespeichert werden. Dabei wird automatisch die
' Datei HAUPTMOD.MAK erstellt.
'
' Beim naechsten Laden des Hauptmoduls HAUPTMOD.BAS ueber "Datei |
' Programm laden" wird das Untermodul UNTERMOD.BAS automatisch mit geladen
' - aufgrund der in der .MAK-Datei hinterlegten Informationen.
'
' (c) Thomas Antoni, 30.11.2005 - www.qbasic.de
'*****
DECLARE FUNCTION Quadrat! (n!)
CLS
FOR i = 1 TO 10
  PRINT Quadrat(i)
NEXT
END
```

Die Programme HAUPTMOD.BAS, UNTERMOD.BAS und HAUPTMOD.MAK stehen steht im Verzeichnis Progs\module.zip zur Verfügung sowie online unter [www.antonis.de/faq/progs/module.zip](http://www.antonis.de/faq/progs/module.zip) .

Das Untermodul UNTERMOD.BAS sieht so aus:

```
DECLARE FUNCTION Quadrat! (n!)
' *****
' UNTERMOD.BAS = Demonstriert die Verwendung von MAK-Modulen in QuickBASIC
' =====
' Die Programme HAUPTMOD.BAS und UNTERMOD.BAS demonstrieren die Verwendung
' von Modulen in QuickBASIC (funktioniert nicht mit QBasic).
' Der Modul UNTERMOD enthaelt eine FUNCTION Quadrat(n), die die zweite
' Potenz der uebergebenen Zahl n zurueckliefert. Quadrat wird vom Haupt.
' Modul aufgerufen.
'
' (c) Thomas Antoni, 30.11.2005)
' *****
FUNCTION Quadrat (n)
    Quadrat = n ^ 2
END FUNCTION
```

Die Programme HAUPTMOD.BAS, UNTERMOD.BAS und HAUPTMOD.MAK stehen steht im Verzeichnis Progs\module.zip zur Verfügung sowie online unter [www.antonis.de/faq/progs/module.zip](http://www.antonis.de/faq/progs/module.zip) .

Beim erstmaligen gemeinsamen Abspeichern der Module erstellt QuickBASIC automatisch die folgenden .MAK-Datei HAUPTMOD.MAK:

```
HAUPTMOD.BAS
UNTERMOD.BAS
```

Die Programme HAUPTMOD.BAS, UNTERMOD.BAS und HAUPTMOD.MAK stehen steht im Verzeichnis Progs\module.zip zur Verfügung sowie online unter [www.antonis.de/faq/progs/module.zip](http://www.antonis.de/faq/progs/module.zip) .

### \*\*\* Anmerkungen

- Ich persönlich benutze mehrere Module z.B. für Testzwecke. Wenn ich also z.B. den Code eines Spiels als ein Modul geladen habe, benutze ich das zweite um z.B. die Figuren zu zeichnen oder um Farben zu testen.
- Alle Informationen beruhen auf eigenen Tests mit QuickBASIC 4.5!

### Antwort 2

~~~~~

[von Ch@rly (karl.pircher@gmx.net) im QB-Forum 28.5.2004]

Hier eine kurze Anleitung, wie das mit den Modulen funktioniert:

Module funktionieren glaube ich mit allen QuickBASIC Compilern, sicher aber ab QB 4.0. Sie funktionieren nicht mit dem Interpretern, also Qbasic 1.1, bzw. GW-BASIC.

Zum Anlegen zweier Module gehst Du wie folgt vor: Zuerst startest Du ganz normal QuickBasic. In die leere Datei schreibst Du ein paar beliebige Befehle, z.B. DEFINT A-Z.

Dann speicherst Du die Datei ab. Nun kannst Du ein Modul anlegen. Dazu rufst Du einfach den Menüpunkt "Datei – Datei erstellen" auf (engl. "File | Create File"). Wenn man jetzt F2 drückt oder den Menüpunkt "Ansicht | SUBs" aktiviert, dann sieht man, dass 2 Module existieren. Mit dem Menüpunkt "Ausführen – Hauptmodul bestimmen" (engl. "RUN – Set Main Module") kannst Du festlegen, welches Modul das Hauptmodul sein soll. Das Hauptmodul ist dasjenige, welches beim Starten aufgerufen wird. Bei der Vorgehensweise wie oben beschrieben, ist das nicht notwendig, da das erste Modul automatisch das Hauptmodul wird.

Jetzt kann man im Modul irgend eine FUNCTION oder SUB programmieren. Um diese aufzurufen, muss im Hauptmodul die DECLARE Anweisung eingefügt werden. Diese wird im Falle einer FUNCTION automatisch beim Speichern des Moduls angelegt. Sie kann dann ganz einfach ins Hauptmodul kopiert werden. Mit einem Trick kann man QB auch dazu bewegen, die DECLARE Anweisung für SUB zu erstellen. Dazu einfach auf der Hauptebene ein CALL der SUB einfügen und das Modul abspeichern. Dann erstellt QB die DECLARE Anweisung, welche ebenfalls ins Hauptmodul kopiert werden muss.

Beim Speichern erzeugt QB übrigens eine Datei mit dem selben Name wie die BAS Datei des Hauptmoduls, aber mit der Endung MAK. Es gibt in diesem Fall also 3 Dateien, die MAK Datei, die BAS Datei des Hauptmoduls und die BAS Datei des Moduls. Der Vorteil der Moule liegt einmal darin, dass sie einzeln kompiliert werden und damit größer sein können, als ein einziges Modul und vor allen auch in der Wiederverwendbarkeit. So kann man z.B. ein Modul Mouse erzeugen, wo man alle Mouseroutinen unterbringt. Bei jedem neuen Projekt braucht man nur dieses Modul hinzufügen und schon hat man alle Mouse Funktionen zur Verfügung. Gleiches gilt natürlich für viele andere Funktionen, welche immer wieder benötigt werden.

Hier noch ein Beispiel:

```
'*** Hauptmodul: MAIN.BAS
DECLARE FUNCTION Test% (Wert%)
DEFINT A-Z
b = Test(a) ' Zu a 1 dazuzzaehlen
PRINT b
```

```
'*** Unterm modul: Modul1.BAS
DECLARE FUNCTION Test% (Wert%)
DEFINT A-Z
FUNCTION Test (Wert)
Wert = Wert + 1
END FUNCTION
```

Antwort 3

~~~~~

[ von Ch@rly ( [karl.pircher@gmx.net](mailto:karl.pircher@gmx.net) ) im QB-Forum 10.4.03 ]

Unter QuickBasic 4.5 und 7.1 kann seine Programme sehr elegant in mehrere Module aufteilen, um der Speicherbeschränkung zu umgehen. In QBasic steht dagegen nur die umständliche Verkettungsmethode mit CHAIN und RUN zur Verfügung.

Du kannst Module dazu benutzen, um Dein Programm aufzuteilen. In die Module schreibst Du einfach Deine SUBs und FUNCTIONS. Im Hauptmodul mußt Du dann noch die Deklaration dieser SUBs und FUNCTIONS einfügen, dann kannst Du sie ganz normal aufrufen, als ob sie sich in Deinem Hauptmodul befinden würden. Auf der Hauptebene der Module kannst Du keine ausführbaren Code eingeben, oder besser gesagt Du kannst ihn zwar eingeben, aber nicht aufrufen. Dort kannst Du nur Dimensionierungen, Deklarationen usw. vornehmen. Durch das Benutzen von Modulen kannst Du ein Programm strukturierter aufbauen und vor allen Dinge viel größere Programme erstellen.

### \*\*\* Frage dazu

Ok aber wie rufe ich denn die Hauptmoduldatei auf? Ich habe 2 BAS-Dateien, einmal das Hauptmodul und einmal den Untermodul:

```
'--- Hauptmodul
PRINT "Nun Kommt der Text aus dem 2ten Modul..."
SLEEP 2
```

```
'--- Untermodul
PRINT "Modul2 Aufgerufen."
```

So und wie rufe ich den PRINT-Befehl nun auf?

### \*\*\* Antwort von Thomas Antoni

Du muss die Befehle des Untermoduls in eine Subroutine packen, die Du vom Hauptprogramm aus aufrufst, z.B. so

Du erstellst in QuickBASIC eine Datei folgenden Inhalts

```
'--- Hauptmodul
DECLARE SUB druck ()
PRINT "Nun Kommt der Text aus dem 2ten Modul..."
CALL druck
SLEEP 2
```

Speichere diese datei unter dem Namen MODUL1.BAS mit "Datei | Speichern unter | Text - Von anderen Programmen lesbar" ab. Dann wählst Du "Datei | Datei erstellen | Modul | modul2.bas" und gibst ins leere Programmfenster den Untermodul wie folgt ein:

```
'--- Untermodul
DECLARE SUB druck ()
'
SUB druck
  PRINT "Modul 2 aufgerufen."
```

END SUB

Nun bestimmst Du den Hauptmodul über "Ausführen | Hauptmodul bestimmen | modul1 | OK". Speichere auch die 2. Moduldatei unter MODUL2.BAS ab. Nun kannst Du das aus 2 geladenen Modulen bestehende Programm starten und erhältst die erwartete Bildschirmanzeige, die so aussieht:

Nun Kommt der Text aus dem 2ten Modul...  
Modul 2 aufgerufen.

### **\*\*\* Weitere Erläuterung von Tobi:**

Du kannst mehrere Module unter QB4.5 und QB7.1 benutzen. Das kann man so machen (Mit QB7.1): Wenn Du das Hauptmodul geladen, klickst Du auf File, und dann auf Create Modul (oder File). Dann gibst Du den Namen des Moduls ein. Wenn Du fertig bist, hat QB eine Datei mit dem eingegeben Namen angelegt. Die Moduldatei kannst Du dann unter F2 aufrufen. Das Modul kann allerdings nur Subs und Funktionen beinhalten. Die kannst Du dann einfach im Hauptmodul aufrufen. Vorher musst Du sie aber noch im Hautmodul deklarieren.