

TASCHENRECHNER IM EIGENBAU

NACHDEM WIR IN DER LETZTEN FOLGE schon einen ersten Blick auf Realbasic geworfen und dabei gelernt haben, aus sieben Zeilen Code einen kompletten Textanzeiger zu basteln, wollen wir uns in dieser Folge komplexeren Anwendungen widmen. Unser Ziel ist, den mageren Taschenrechner des Mac-OS durch leistungsfähige Eigenentwicklungen zu ergänzen

VON CHRISTIAN SCHMITZ



WIR STARTEN mit einem Additionsprogramm, das schlicht und einfach zwei Zahlen addiert und uns die Summe im Klartext in einem Dialog ausgibt.

Nach dem Doppelklick auf Realbasic erscheint wie immer ein neues, leeres Projekt, und Sie sehen ein leeres Fenster mit dem Titel „Namenlos“ vor sich. Ziehen Sie jetzt per Drag-and-drop von der Werkzeugleiste nacheinander drei Edit-Felder (Eingabefeld, Sie finden es in der Werkzeugleiste über der „Stoppuhr“) in das Fenster. Zwei der drei Edit-Felder dienen uns als Eingabefelder für die Zahlen. Das Dritte zeigt nachher das Ergebnis an, ist also ein reines Ausgabefeld, in das man nichts eintippen kann. Mit ein paar „Statiertexten“ (links oben in der Werkzeugleiste das Kontrollelement mit dem großen „A“) verschönern wir das Fenster noch um ein „+“ und ein „=“. Ein Separator, also eine Linie, die je nach dem im Mac-OS ausgewählten Erscheinungsbild anders aussieht, dient als dicker Trennstrich. Um Ausgabe- und Eingabefelder zu unterscheiden, ändern wir die Hintergrundfarbe des Ausgabefeldes (Eigenschaft „BackColor“ in der Palette „Eigenschaften“ rechts) mit dem Farbpicker. Damit niemand das Ergebnis nachträglich ändern kann, aktivieren wir die Checkbox

für „Readonly“ (nur lesen). Als Letztes ergänzen wir noch einen „Button“ (in der Werkzeugleiste mit „OK“ beschriftet), dem wir im Eigenschaftenfenster den Namen (Caption) „Addition“ geben.

Nach dem Speichern (Befehl-S) und dem Starten (Befehl-R) sollte das Fenster dann so aussehen wie in der Abbildung links unten auf dieser Seite.

Jetzt kann man noch überall herumklicken, ohne dass etwas passiert. Sie ahnen sicher, dass da noch ein wenig Programmcode fehlt. Also machen Sie bitte im Realbasic-Editor einen Doppelklick auf den „Addition“-Button. Dadurch gelangen Sie automatisch in den Code-Editor. Dort wählen Sie den „Action Event“ des Buttons aus und geben rechts Ihren Code ein.

Wenn der Benutzer auf den Knopf „Addition“ klickt, wird der Action-Event zur Laufzeit aufgerufen. Dann soll das Programm die zwei Edit-Felder oben auslesen, die Zahlen addieren und das Ergebnis in das untere Feld eintragen und anzeigen.

KLEINER EXKURS IN DATENTYPEN

Dazu müssen Sie zunächst wissen, was Datentypen sind. Es gibt Variablen, die wie Schubladen mit Namen versehen sind und bestimmte Werte speichern können. Beim Computer gibt es jedoch für jede Art von Werten unterschiedliche Datentypen. Für Zahlen existieren je nach Anzahl der (Nachkomma-) Stellen unterschiedliche Variablen. Die häufigste ist „Integer“, die wir für alle ganzen Zahlen von -2147483648 bis +2147483647 wählen (eine Integer-Zahl ist intern also eine 32-Bit-Zahl). Integer-Zahlen reichen jedoch nicht immer aus, und so gibt es den Datentyp „Double“, der jegliche Zahlen speichern kann. „Double“ reicht zwar für mehrere hundert Stellen große Zahlen, besitzt aber nur eine Genauigkeit von

zirka 20 Stellen. Außerdem lassen sich periodische Brüche wie zum Beispiel 1/3 nur ungefähr, also fehlerbehaftet, erfassen.

Der Datentyp „Single“ hat eine geringere Zahlenmenge und braucht weniger Speicher, ist aber auf dem Power-PC-Prozessor nicht direkt implementiert und wird jedes Mal zur Berechnung in eine Double-Variable umgewandelt, was ihn langsamer macht als die doppelt so großen Doubles. Daher empfiehlt es sich, nur Integer- oder Double-Variablen zu benutzen.

Um Texte wie beispielsweise „Hallo“ zu speichern, gibt es den Datentyp „String“. Er dient als Variable für jeglichen Text bis zu einer Größe von (theoretisch) 2 GB. Man

WAS SIND EVENTS?

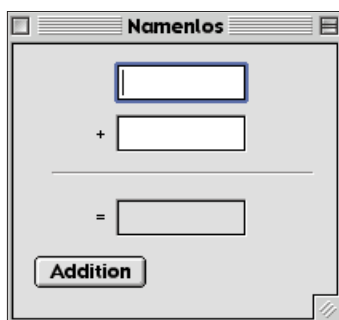
Ein Event ist für den Computer ein beliebiges Ereignis. Wenn der Benutzer eine Taste drückt oder die Maus bewegt und irgendwo hinklickt, bekommt das aktive Programm vom Mac-OS einen Event geschickt, der es über die Art des Ereignisses informiert. In Realbasic werden dadurch so genannte „Event-Handler“ aufgerufen. Letztere findet man in jedem Objekt, also in Fenstern, Buttons und Klassen. Man kann in die Event-Handler eines Objekts, zum Beispiel den „Action-Event“ des Push-Buttons etwas Code einfügen und damit auf das Ereignis reagieren. Ein Programm wird so lange von diesen Ereignissen gesteuert, bis es zum Ereignis „Quit“ kommt, dann beendet es sich selbst, und man landet im Finder.

beachte den feinen Unterschied zwischen 5 und "5", denn die 5 alleine ist ein Integer, und die "5" in Anführungszeichen ist ein Text, also ein String. Strings kann man einfach mit der Addition „+“ aneinander kleben. Der Befehl: `me.Text = "Hallo" + " Leute!"` im Open-Event eines Edit-Feldes zeigt das Aneinanderketten von Texten. Und ein `me.text="5"+"5"` ergibt eine "55" und keine 10! Dabei wird der fertige Text in die Variable „Text“ des Edit-Feldes gespeichert, die den aktuellen Inhalt darstellt.

RECHNEN MIT ZAHLEN

Variablen vom Typ „Zahl“ lassen sich für verschiedene mathematische Methoden benutzen. Nach einem „Dim a,b,c as double“ (drei Variablen anlegen mit dem Namen a, b und c, alle vom Typ Double, also für Zahlen mit Nachkommastellen) kann man inner-

VIER OBJEKTE REICHEN AUS, um einen kleinen Rechner für die Addition von Zahlen zu programmieren.



halb des Action-Event von einem Button rechnen. Möglich sind Konstruktionen wie „a=b+c“ oder „a=b*c“. Was es an weiteren möglichen Operationen gibt, können Sie dem Kasten „Mathematik in Realbasic“ auf dieser Seite links unten entnehmen.

ZAHLEN UND STRINGS UMWANDELN

Die Texte in Edit-Feldern sind Strings, aber gerechnet wird mit Double-Variablen. Wollen Sie eine Double-Variable aus dem Text eines Edit-Feldes erzeugen, müssen Sie den Variablentyp umwandeln. Dazu gibt es mehrere Funktionen. Für Zahlen im US-Format, also etwa „1,456.345“ für 1456,345, können Sie die Funktionen „Val()“ und „Str()“ nehmen. Val interpretiert einen String und versucht, daraus eine Zahl zu erzeugen, ignoriert aber alles, was nicht nach Zahl aussieht. Dazu einige Beispiele: Aus "1,234.5" wird 1234.5, aus "1.45" wird 1.45, und aus "1.5hallo" wird 1.5. Die Funktion „Str“ geht den umgekehrten Weg und macht aus einer Zahl einen String im US-Zahlenformat. Auch hierzu einige Beispiele: Aus 1.5 wird "1.5", aus 0.123456 wird "0.123456" und aus 123456789 wird "1.234568e+8". Beide Funktionen können mit der wissenschaftlichen Notation umgehen, wie 1e5 oder 8.643e-56.

Da man hier zu Lande das deutsche Zahlenformat bevorzugt, gibt es dafür die Funktionen „Cdbl()“ und „Format()“. Die Funktion Cdbl() interpretiert einen String und gibt die passende Zahl zurück, wobei sich Cdbl() dabei an die Vorgaben aus dem Kontrollfeld „Zahlenformat“ hält. Ansonsten entspricht diese Funktion der Funktion Val(). Die Funktion Format() dient zum Formatieren einer Zahl. Einige Beispiele: Format(-1.5,"0.0") ergibt "1.5", aus Format(-1.345,"-00.00") wird "-01.35", und Format(123456789,"####,###,##0.0") liefert "123.456.789.0" zurück.

Dabei richtet sich die Formatierung nach einem Formatierungs-String (im oben angeführten Beispiel sind das die vielen Doppelkreuze), wobei Punkte durch Dezimaltrenner und Kommas durch Tausendertrenner ersetzt werden. Eine Null (0) wird durch eine Ziffer ersetzt oder durch ein "0", wenn keine da ist. Aus "#" wird eine Ziffer

oder gar nichts, also ein Leer-String. Es gibt noch unzählige weitere Optionen, die den Rahmen dieses Artikels sprengen würden. Hier lohnt sich ein Blick ins Realbasic-Handbuch oder in die Online-Hilfe.

UND DER MAC RECHNET...

Jetzt kommt etwas Code in unseren Taschenrechner. Wir sind immer noch im Code-Editor, und der Action-Event ist ausgewählt. Dort deklarieren wir zuerst zwei Variablen für unsere Zahlen: „Dim Zahl1, Zahl2 as Double“. Wie gehabt reserviert „Dim“ ein Stück Arbeitsspeicher und gibt ihm den genannten Namen „Zahl1“ beziehungsweise „Zahl2“ mit dem Typ „Double“.



DIESE SIEBEN ZEILEN Realbasic-Code lesen die Edit-Felder aus, berechnen das Ergebnis und schreiben es zurück in das Ausgabefeld.

Danach haben wir zwei Variablen („Zahl1“ und „Zahl2“) für Zahlen mit Nachkommastellen. Eine weitere Double-Variable für das Ergebnis wird in der nächsten Codezeile mit „Dim Ergebnis As Double“ eingeführt. Man kann auch alle drei Variablen in einer einzigen Zeile deklarieren, aber so wird unser Code übersichtlicher.

Nachdem wir jetzt alle erforderlichen Variablen deklariert haben, fangen wir an, sie zu füllen. Dazu benutzen wir zwei Codezeilen mit: „Zahl1=cdbl(editfield1.text)“ und in der nächsten Zeile: „Zahl2=cdbl(editfield2.text)“. Was in diesen Zeilen passiert, nennt man eine Zuweisung. Der Variablen links vom „=“ wird ein Wert zugewiesen, der rechts ermittelt wird. Die Variable links ist hier einmal „Zahl1“ und einmal „Zahl2“. Um den String einer Double-Variable zuzuweisen, wandeln wir ihn mit der Funktion Cdbl() um. Anschließend wird der Text interpretiert und die passende Zahl in unsere Variable gespeichert.

In der Zeile „Ergebnis=Zahl1+Zahl2“ findet die eigentliche Addition statt, indem der Variable „Ergebnis“ der Wert der Summe von Zahl1 und Zahl2 zugewiesen wird. Wenn Sie diese Zeile später ändern, können Sie leicht aus dem Additionsprogramm ein Programm für Subtraktion, Multiplikation oder Division mit und ohne Rest machen.

Zum Schluss speichern wir das Ergebnis mit der Zeile „Editfield3.text=Format(Ergebnis,\"0.0\")“ in das Edit-Feld. In diesem Fall müssen wir aus der Zahl, die in der Variable Ergebnis steht, einen formatierten String machen, der dann schließlich im dritten Edit-Feld ausgegeben wird. Mit dem

Ausdruck "0.0" geben wir an, dass wir uns die Ausgabe in mindestens einer Vor- und einer Nachkommastelle wünschen.

WEITERE IDEEN

Durch weitere Knöpfe lassen sich relativ einfach Subtraktion, Multiplikation und Division hinzufügen. Das finden Sie auch in unseren Beispielen auf *Macwelt Online* (www.macwelt.de/_magazin). Das Beispiel dort hat zusätzlich einen Reset-Knopf, um alle Edit-Felder auf einmal zu löschen. Hierbei wird jeweils das Textobjekt des Edit-Feldes auf "" (Leer-String) gesetzt. Also EditField1.text="", EditField2.text="" und EditField3.text="".

Natürlich kann man dieses Programm auch dahin gehend erweitern, dass es für andere Aufgaben taugt. Nehmen Sie so viele Edit-Felder zur Eingabe wie nötig, einen Knopf „Rechnen“ und ein paar Edit-Felder zur Ausgabe. Jetzt können Sie ähnlich wie oben eine passende Rechnungszeile erzeugen, wie im Beispiel „Mehrwertsteuer“ (auch auf *Macwelt Online* zu finden).

MENÜS IN REALBASIC

Jede Mac-OS-Applikation bietet eine Menüleiste, auch wenn diese nur den Eintrag „Über dieses Programm...“ enthält, mit dem man eine so genannte Aboutbox aufruft.

Sie haben das Additionsprogramm noch in Realbasic vor sich offen? Dann klicken Sie auf den Eintrag „Menü“ im Projektfenster (man kann das Projektfenster mit Befehl-0 aufrufen, wenn es nicht zu sehen sein sollte). Nun öffnet sich ein Fenster mit dem Menüeditor. Wie Sie sehen, bietet jedes Realbasic-Projekt von Anfang an eine Menüleiste, die bereits voll funktionstüchtige Menüeinträge für „Quit“, „Kopieren“, „Ein-“



JEDES NEUE REALBASIC-Projekt besitzt bereits eine Menüleiste mit Einträgen für „Ablage“ und „Bearbeiten“.

MATHEMATIK IN REALBASIC

Folgende Operatoren gelten in Realbasic für Variablen mit den Typen Integer, Single und Double:

+	Addition
-	Subtraktion
*	Multiplikation
/	Division (mit Nachkommastellen)
\	Division (mit Ganzzahlergebnis)
mod	Division (Rest von \)

fügen“, „Ausschneiden“ und „Löschen“ enthält. „Undo“ ist vorhanden, funktioniert jedoch nicht ohne zusätzlichen Code.

RECHNEN MIT DEM MENÜ

Unser Programm sollte auch über den Menüpunkt „Rechnen“ arbeiten und auf das Tastenkürzel „Befehl-R“ reagieren.

Dazu klicken Sie aufs Ablagemenü im Menüeditor und dann auf den leeren Eintrag unter dem Beenden-Befehl. Nun geben Sie in der Eigenschaftenpalette oben beim leeren Feld für Text den Text „Rechnen“ ein und drücken Enter. Damit haben Sie den Menüeintrag „Ablage/Rechnen“ erstellt. Nie fehlen sollte ein leicht zu merkendes Tastenkürzel, in unserem Beispiel also Befehl-R. Geben Sie in der Palette bei „CommandKey“ ein „R“ ein. Auch andere Kürzel, die Sie dem Kasten „Tastenkürzel in Realbasic“ entnehmen können, sind möglich. Wenn Sie möchten, geben Sie noch einen Erklärungstext bei „Balloonhelp“ ein. Dieser erscheint dann automatisch, wenn man mit der Maus bei eingeschalteten Erklärungen (findet man im „Hilfe“-Menü) über den Menüeintrag fährt.

EIN BISSCHEN CODE MUSS SEIN

Wir wechseln zum Dialogfenster unseres Programms. Mit Wahl- und Tabulatortaste öffnet sich der Code-Editor des Fensters. Wählen Sie in der Realbasic-Menüleiste unter „Bearbeiten“ den Eintrag „Neuer Menü-

Handler“ aus und im Dialog „Ablage/Rechnen“. Hier geben wir als Programmcode lediglich eine Zeile ein: „pushbutton1.push“. Dieser Code simuliert den Mausklick auf unseren Additions-Button, löst also den Rechenbefehl aus. Sollte Sie das Drücken des Push-Buttons optisch stören, können Sie seinen kompletten Programmcode in den Menü-Handler kopieren. Dann rechnet das Programm, ohne den Button zu betätigen. Alternativ lässt sich eine „Methode“ (Unterprogramm) schreiben und aufrufen. Wie das geht, zeigen wir anhand weiterer Beispielprogramme im Internet unter www.macwelt.de/_magazin.

Wenn Sie das Programm nun starten, fällt auf, dass sich der neue Menüeintrag „Rechnen“ nicht auswählen lässt. Er ist grau dargestellt, also „disabled“. Es fehlt noch etwas Code, mit dem man Menüeinträge freischaltet. In den Events vom Dialogfenster (im Code-Editor) gibt es den Event „Enable MenuItems“, den Realbasic immer dann aufruft, wenn Sie in die Menüleiste klicken. Dann müssen Sie im Programmcode entscheiden, welche Menüeinträge Sie aktivieren, denn per Default ist zunächst einmal alles „disabled“. Für unseren Menüeintrag ergänzen wir: „AblageRechnen.enabled=true“. Wir schalten also den „Enabled“-Schalter des Menüeintrags „AblageRechnen“ ein. Damit ist der Eintrag auswählbar.

PLATZ ZUM SPIELEN: ABOUTBOX.

Was noch fehlt, ist die Verewigung. In einem Dialog im Apple-Menü finden sich in der Regel die Namen der Programmierer und die Programmversion. Außerdem lässt sich die Aboutbox mit einem Bild verschönern.

Dazu gehen Sie wieder in den Menüeditor und ergänzen wie oben einen Eintrag für „Über das Additionsprogramm...“. Es ist nicht üblich, dafür ein Kürzel einzubauen. Stattdessen sollten Sie an den Text noch drei Punkte (drei Mal „...“ oder einmal „...“), Wahl taste-Punkt) anfügen, denn bei solchen Punkten sieht der Benutzer sofort, dass sich eine neue Dialogbox öffnen wird. Benennen Sie den Menüeintrag auf jeden Fall in „AppleAbout“ um, damit er nicht den Default-Namen „AppleÜberdasAddierprogramm...“ behält. Deutsche Umlaute und die drei Punkte „...“ sind für Objektnamen nicht zugelassen und führen zu Fehlermeldungen.

IN ZWEI MINUTEN ZUR ABOUTBOX

Für eine ansprechende Aboutbox erzeugen wir ein neues Fenster (Menüeintrag „Neues Fenster“ im Ablagemenü von Realbasic). Geben Sie ihm einen aussagekräftigen Titel und füllen Sie es mit Objekten Ihrer Wahl. Es sollte ein paar Textobjekte (Statictext) bekommen und mindestens einen Push-Button, der einen „Close“-Befehl im Action-Event enthält, um das Fenster zu schließen.



SO KÖNNTE eine einfache Aboutbox mit integriertem Bild aussehen.

Per Drag-and-drop können Sie ein beliebiges Bild (PICT, JPEG, GIF, usw.) in Ihr Projektfenster ziehen. Dazu legen Sie im About-Fenster ein Canvas an (Sie finden es in der Werkzeugleiste. Es ist das Bildobjekt mit dem stilisierten „Sonnenuntergang“) und wählen als „Backdrop“ für das Canvas das Bild aus. Für den Schließen-Button empfiehlt es sich, die Eigenschaften „default“ und „cancel“ zu aktivieren, damit man den Dialog auch per Zeilenschalter (Return-Taste) und Escape-Taste verlassen kann.

Jetzt müssen wir nur noch den Menüeintrag freischalten, damit er anwählbar ist. Dazu ergänzen wir im „EnableMenuItems“-Event die Codezeile „AppleAbout.enabled=true“ und legen einen Menü-Handler für „AppleAbout“ an, den wir mit dem Code „dialog1.showmodal“ füllen. Durch diesen Befehl zeigt Realbasic unsere Aboutbox auf dem Bildschirm an.

Noch ein kurzer Tipp: Sie müssen das laufende Additionsprogramm nicht immer mit „Befehl-Q“ beenden, um Änderungen im Quellcode oder in der Optik vorzunehmen. Ein Klick auf ein beliebiges Realbasic-Fenster im Hintergrund genügt, um das laufende Programm zu unterbrechen. Es ist dann allerdings noch nicht beendet. Erst durch „Befehl-K“ (Kill) wird es dauerhaft angehalten und aus dem Speicher entfernt.

















FAZIT

Ähnlich wie im letzten Teil unserer Realbasic-Serie genügen uns einige wenige Zeilen Code, um einen funktionsfähigen Taschenrechner inklusive Menüsteuerung zu programmieren. Doch nur Text und Zahlen zu bearbeiten ist langweilig. In der nächsten Folge wird es deshalb bunt und grafisch. Wir basteln uns kleine Grafikprogramme, um die Leistung der Grafikkarte im 2D-Bereich zu messen und lernen außer den Grafikbefehlen auch, wie man Zeitmessungen in Realbasic programmiert. *cm*

Serie Realbasic

- 1 EinführungHeft 9/2000
- 2 Taschenrechner im EigenbauHeft 10/2000
- 3 Grafik totalHeft 11/2000
- 4 Quicktime-ProgrammierungHeft 12/2000

TASTENKÜRZEL IN REALBASIC

 oder  + 	„X“ oder „command-X“
 + 	„shift-X“
 + 	„option-X“
 + 	„control-X“
 +  + 	„shift-option-X“
 +  +  + 	„control-shift-option-X“

Im Realbasic-Menüeditor kann man einzelnen Menüpunkten ein bestimmtes Tastenkürzel beziehungsweise eine Tastenkombination zuweisen. Dabei tauchen Bezeichnungen wie command (Befehl), shift (Umschalt), option (Wahl) und control (Control) auf, die man exakt so in das Feld „commandkey“ eintippt.

Bedenken Sie außerdem, dass unter Microsoft Windows die Tastenkombinationen mit der Umschalt-, Wahl- und Control-Taste nicht funktionieren, denn dort ist „Control“ gleich „Befehl“, und die Wahl taste ist für die Navigation in der Windows-Menüleiste reserviert.

Stören Sie sich nicht daran, wenn diese Kürzel im Menüeditor seltsam aussehen – sie funktionieren dennoch. Real Software wird eigenen Aussagen zufolge den Menüeditor noch verbessern.