

7. Einführung in die Visual Basic - Programmierung

1. Starten Sie die Entwicklungsumgebung Microsoft **Access** und legen Sie in einem Verzeichnis auf dem **Z-Laufwerk**: (z.B. unter Z:\Info) nach Anleitung ein neues Datenbankprojekt **info1.accdb** an.
2. Erstellen Sie nach Anleitung ein Formular **Test1** und definieren Sie in diesem eine Überschrift „**Einführung in die VB –Programmierung**“ und **2 Ein-/Ausgabefelder** mit den Namen **Wert1/Wert2** (sowohl als Label in der Oberfläche als auch als interner Feldname der Eingabefelder). Mit den Formatierungsoptionen können Sie das Design selbst anpassen.
3. Definieren Sie einen **Button** rechts neben den Eingabefeldern mit der Aufschrift „**Test1**“ und dem internen Namen „**Test1**“. Rufen Sie nach Anleitung die integrierte Arbeitsumgebung von Visual Basic hinter dieser Taste auf.
4. Schreiben Sie in der Ereignisprozedur der Taste den Befehl **MsgBox „Willkommen bei Visual Basic“** und testen Sie den Aufruf.
5. Schreiben Sie in der gleichen Prozedur eine zweite Programmzeile mit der Sie den Wert des ersten Eingabefeldes anzeigen:
MsgBox „Wert1 =“ & Me!Wert1
Erweitern Sie den Ausdruck selbst um die Anzeige auch des 2. Wertes
6. Da die modale (=blockierende) Anzeige der Messageboxen für normale Werteausgaben nicht sehr günstig ist, soll eine Ausgabe des 1. Wertes im 2. Feld erfolgen mit
Me!Wert1 = „Erster Wert ist “ & Me!Wert2
7. Erstellen Sie auf der Formularoberfläche eine neue Taste „**Quadrat**“ und definieren Sie in der zugehörigen VB-Prozedur den dazu notwendigen Code. Beachten Sie bitte, dass zur wirklichen Arbeit mit Zahlen diese aus dem Eingabetext mit **val()** VORHER in eine Zahl gewandelt werden müssen, z.B. mit
Dim w1 as integer
i = val (Me!Wert1)
8. Definieren Sie zur Demonstration eines größeren Anwendungsbeispiels eine neue Taste „**Wurzelberechnung**“ auf der Formularoberfläche und bauen Sie für die Berechnung der Wurzel aus dem ersten Eingabefeld mit Ausgabe im 2. Feld den Algorithmus aus der 3. Übung ein (bzw. nutzen Sie das Struktogramm aus Übung 4). Testen Sie die Ausgabe auf Korrektheit. Lassen Sie sich alle Zwischenergebnisse im Direktfenster mit **debug.print** ausgeben!
9. Entwickeln Sie zur Berechnung der auf dem Rechner **verfügbaren maximalen Genauigkeit** eine Funktion unter der Taste „**Genauigkeit ermitteln**“. Definieren Sie dazu im Codeteil eine Rechnung $k = e + f$: Rem alle Werte **double** und setzen Sie $e=1.0$; und $f=0.1$ Verkleinern Sie in einer Schleife f durch Division durch 10, bis sich als Ergebnis der Berechnung 1 ergibt. Der Wert für f gibt dann näherungsweise die Genauigkeit von **double**-Rechnungen auf diesem Computer an und kann als konkreter Wert für ϵ verwendet werden!
Lassen Sie sich alle Zwischenergebnisse im Direktfenster mit **debug.print** ausgeben!