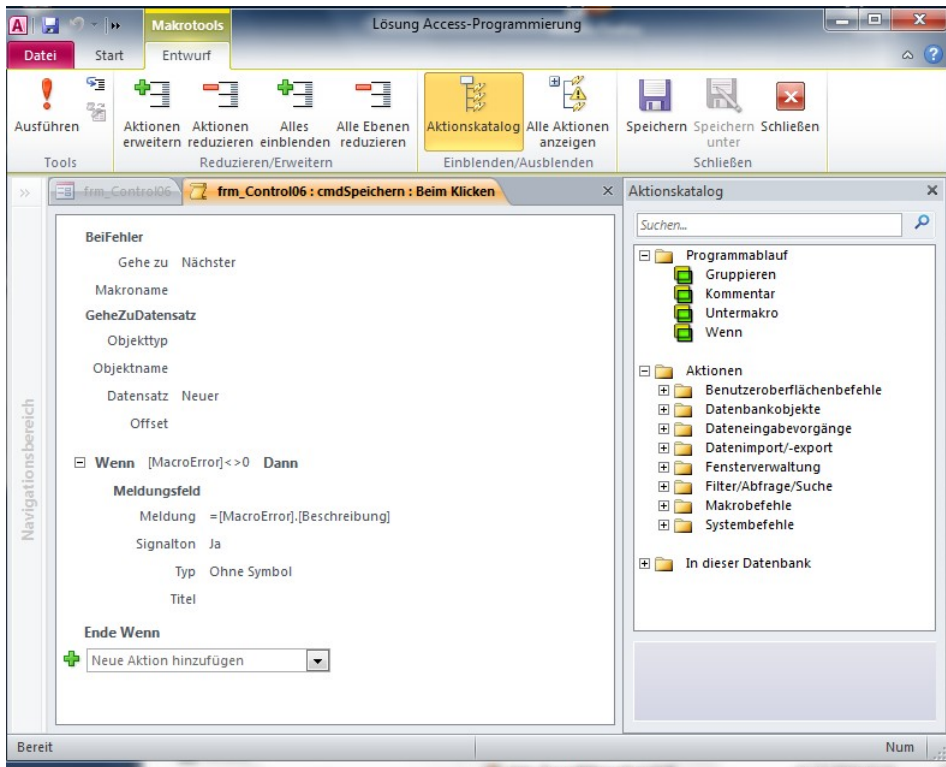


Access 2010 – Programmierung



Bücher

- Handbuch des RRZN: Access 2010; Automatisierung und Programmierung.
- Dobrenz, Gewinnus: Microsoft Access 2010 Programmierung; Microsoft Press
- Hölscher, Lorenz: Richtig einsteigen: Access 2010 VBA-Programmierung; Microsoft Press.
- Kuhlmann / Brudermanns: Programmieren mit VBA; Merkur Verlag Rinteln
- Henning, Cooper, Griffith, Dennison: Access 2010. Programmers Reference; Jon Wiley & Sons.

Links zu Webseiten

- Microsoft Developer Center:
<http://msdn.microsoft.com/de-de/library/ff604965.aspx>
- FAQ:
 - <http://www.donkarl.com/>
 - <http://allenbrowne.com/tips.html>
- Tutorial:
 - <http://www.functionx.com/vbaccess/index.htm>

Makros ...

- automatisieren Arbeitsschritte.
- fassen Aktionen zusammen.
- führen Aufgaben auf Knopfdruck zusammen.
- können beliebig oft aufgerufen werden.
- werden intern als V(isual)B(asic for)A(pplication)-Code gespeichert.

Vorteile

- Mit Hilfe von vorgefertigten Aktionen kann sehr schnell ein „Programm“ erstellt werden.
- In Steuerelementen und Formularen / Berichten eingebettete Makros werden automatisch mit kopiert.
- Makros können auch in webbasierten Datenbanken eingesetzt werden.
- Sichere Makro-Aktionen laufen auch, wenn VBA-Code aufgrund der Sicherheit deaktiviert ist.
- Makros können beim Start der Datenbank ausgeführt werden. Starteinstellungen können automatisiert gesetzt werden.

Nachteile

- Makros benötigen immer die gleichen Startbedingungen.
- Begrenzter Befehlsumfang. Schleifen für die Wiederholung von Aktionen sind nicht implementiert.
- Aktionen sind vorgefertigt. Die Schritte, die in einer Aktion zusammengefasst werden, können nicht vom Entwickler kontrolliert werden.
- Makros mit sehr vielen Aktionen sind langsamer als VBA-Code.
- Makros können nur eingeschränkt mit anderen Anwendungen interagieren.

Makrotypen

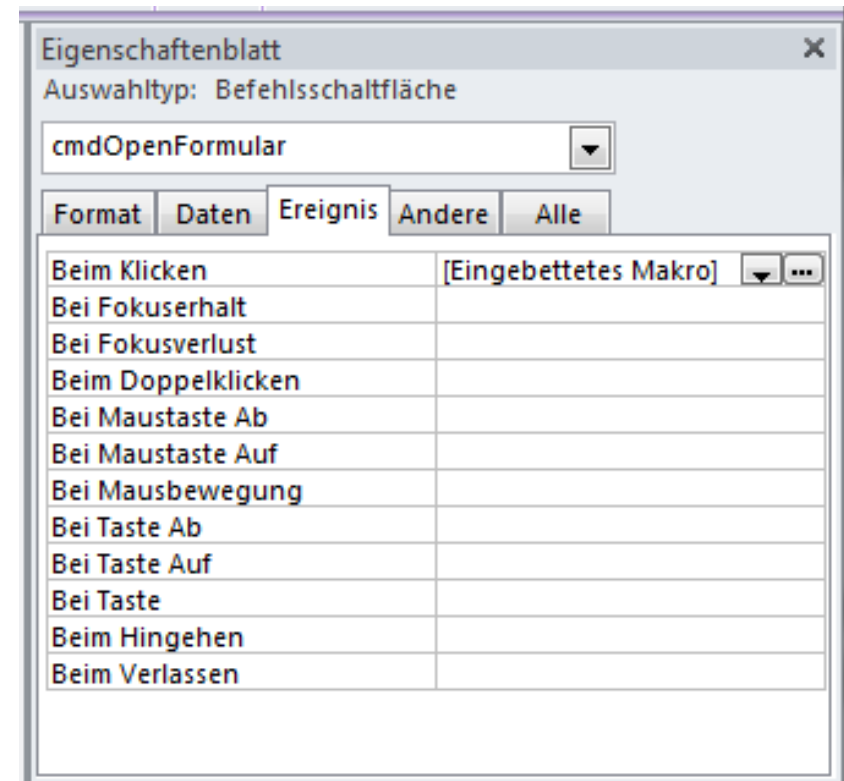
- Makro-Objekte werden in dem Navigationsbereich aufgelistet. Makro-Objekte können von jedem Formular, Bericht oder Steuerelement aufgerufen werden.
- Eingebettete Makros sind immer an ein bestimmtes Ereignis eines Formulars / Berichts oder Steuerelements gebunden. Die Makros werden automatisch mit dem dazugehörigen Objekt kopiert.
- Datenmakros wurden mit Access 2010 eingeführt und sind an eine Tabelle gebunden. Datenmakros reagieren auf Ereignisse wie Aktualisieren oder Löschen von Daten. Diese Art von Makros arbeitet wie ein Trigger in SQL.

Eingebettete Makros ...

- werden automatisch angelegt, wenn der Steuerelement-Assistent bei der Erstellung von Befehlsschaltflächen genutzt werden.
- können über die Registerkarte Ereignis im Eigenschaftenblatt eines Formulars / Berichts oder Steuerelements erstellt werden.

... im Eigenschaftenblatt

- Auf der Registerkarte Ereignis werden verschiedene Ereignisse eines Objekts angezeigt.
- Mit Hilfe der Schaltfläche *Drei Punkte* und der Auswahl Makro-Generator kann ein neues Makro erstellt werden.
- Mit Hilfe der Schaltfläche *Schwarzer Pfeil nach unten* kann ein vorhandenes Makro angebunden werden.



Ereignis (Event) ...

- tritt immer in Verbindung mit einem Formular, Bericht oder Steuerelement auf.
- wird durch die Maus, die Tastatur oder einem Zeitgeber ausgelöst.
- ist eine Reaktion auf eine Aktion, die der Benutzer gestartet hat.
- Jedes Objekt in Access hat bestimmte Ereignisse. Zum Beispiel ein Bericht hat das Ereignis „Keine Daten vorhanden“. Ein Formular hat dieses Ereignis nicht.
- werden zeilenweise auf dem Register *Ereignis* im Eigenschaften-Fenster angezeigt.

... für Formulare

Ereignis	Erläuterung
Beim Laden	Das Formular wird erstmalig geöffnet, wird aber noch nicht für den Benutzer angezeigt. Standwerte können gesetzt werden. Eigenschaften können geändert werden.
Beim Schließen	Das Formular wird geschlossen. Das Ereignis kann rückgängig gemacht werden.
Beim Anzeigen	Ein neuer Datensatz wird angezeigt.
Bei Aktivierung	Das Formular erhält den Focus. Das Formular wird in den Vordergrund geschoben.
Bei Deaktivierung	Das Formular verliert den Focus. Das Formular wird in den Hintergrund geschoben.

... für Berichte

Ereignis	Erläuterung
Beim Laden	Der Bericht wird erstmalig geöffnet.
Beim Schließen	Der Bericht wird geschlossen. Das Ereignis kann rückgängig gemacht werden.
Beim Anzeigen	Ein neuer Datensatz wird angezeigt.
Bei Aktivierung	Der Bericht erhält den Focus. Der Bericht wird in den Vordergrund geschoben.
Bei Deaktivierung	Der Bericht verliert den Focus. Der Bericht wird in den Hintergrund geschoben.
Bei Ohne Daten	Der Bericht enthält keine Daten.

... für Datensätze

Ereignis	Formular	Textfeld	Erläuterung
Bei Geändert	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Der Datensatz in einem Formular oder Textfeld ändert sich.
Bei Rückgängig	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Beim Anzeigen	<input checked="" type="checkbox"/>		
Vor Löschbestätigung Nach Löschbestätigung	<input checked="" type="checkbox"/>		Der Datensatz wird gelöscht.
Vor Eingabe Nach Einfügung	<input checked="" type="checkbox"/>		Ein neuer Datensatz wird eingefügt.
Vor Aktualisierung Nach Aktualisierung	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Nach einer Änderung werden die Daten aktualisiert.

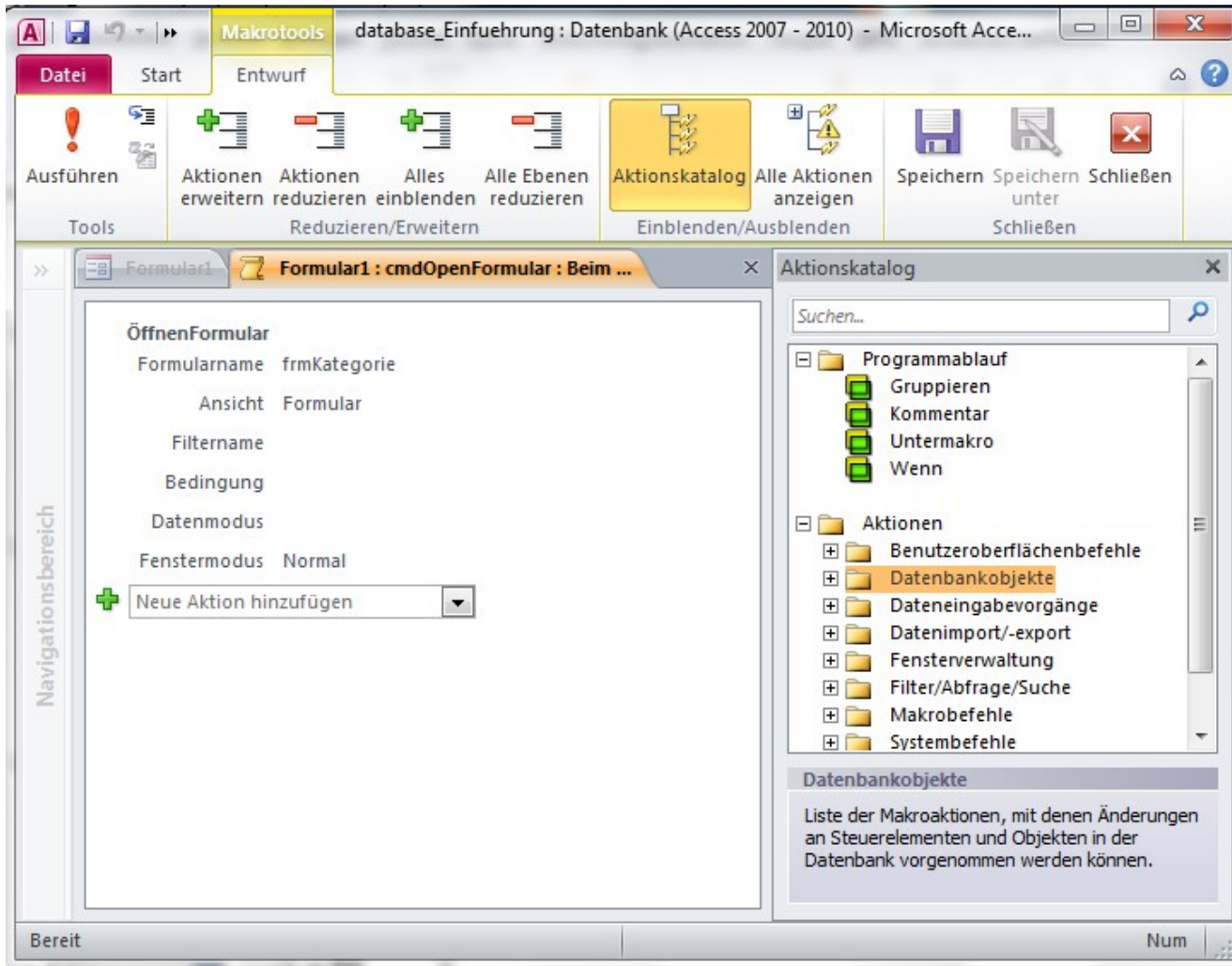
... für Steuerelemente

Ereignis	Steuerelemente	Erläuterung
Beim Hineingehen	alle	Das Steuerelement erhält den Focus.
Beim Verlassen	alle	Das Steuerelement verliert den Focus.
Bei Taste	alle	Eine Taste wird gedrückt.
Beim Klicken	Schaltflächen	Mit der linken Maustaste wird auf die Schaltfläche geklickt.
Bei nicht in Liste	Kombinationsfeld	Der eingegebene Wert ist nicht in der Liste vorhanden.

Eingebettete Makros öffnen

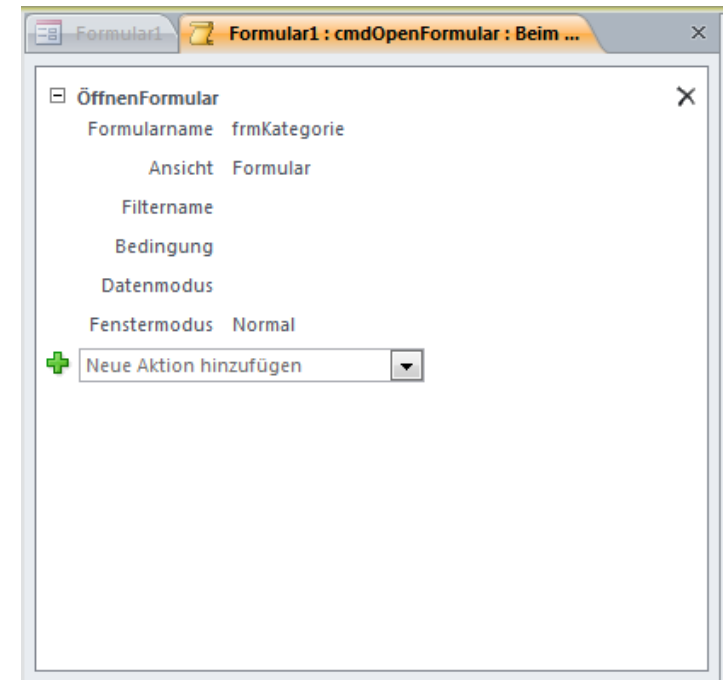
- Die Registerkarte Ereignis des ausgewählten Objekts ist im Eigenschaftenblatt geöffnet.
- Mit Hilfe der Schaltfläche *Drei Punkte* wird die Entwurfsansicht des Makros geöffnet.

Eingebettetes Makro in der Entwurfsansicht



Entwurfsbereich des Makros

- Der Reiter gibt Informationen über das zugehörige Objekt und Ereignis.
- Zeilenweise werden darunter die Aktion in dem Makro angezeigt.
- Mit Hilfe des Minuszeichens werden die Argumente zu den Aktionen eingeblendet. Mit Hilfe des Pluszeichens werden die dazugehörigen Argumente eingeblendet.
- Mit einem Klick auf das Kreuz am rechten Rand wird eine Aktion gelöscht.



Aktionen ...

- beschreiben Arbeitsschritte in einem Makro.
- sind vorgefertigter Code für den Entwickler von Datenbanken.
- werden automatisiert Zeile für Zeile abgearbeitet.
- In dem vorherigen Beispiel wird mit Hilfe der Aktion « ÖffnenFormular » ein bestimmtes Formular geöffnet.

Argumente von Aktionen ...

- werden benötigt, um eine Aktion korrekt auszuführen.
- beeinflussen die Ausführung der Aktion.
- werden eingerückt unterhalb des Aktionsnamens angezeigt.
- Die Anzahl und deren Typ ist abhängig von der gewählten Aktion.
- Der Aktion « ÖffnenFormular » wird
 - der Name des zu öffnenden Formulars,
 - die Darstellungsart des Formulars,
 - die Zugriffsrechte auf die angezeigten Daten übergeben.

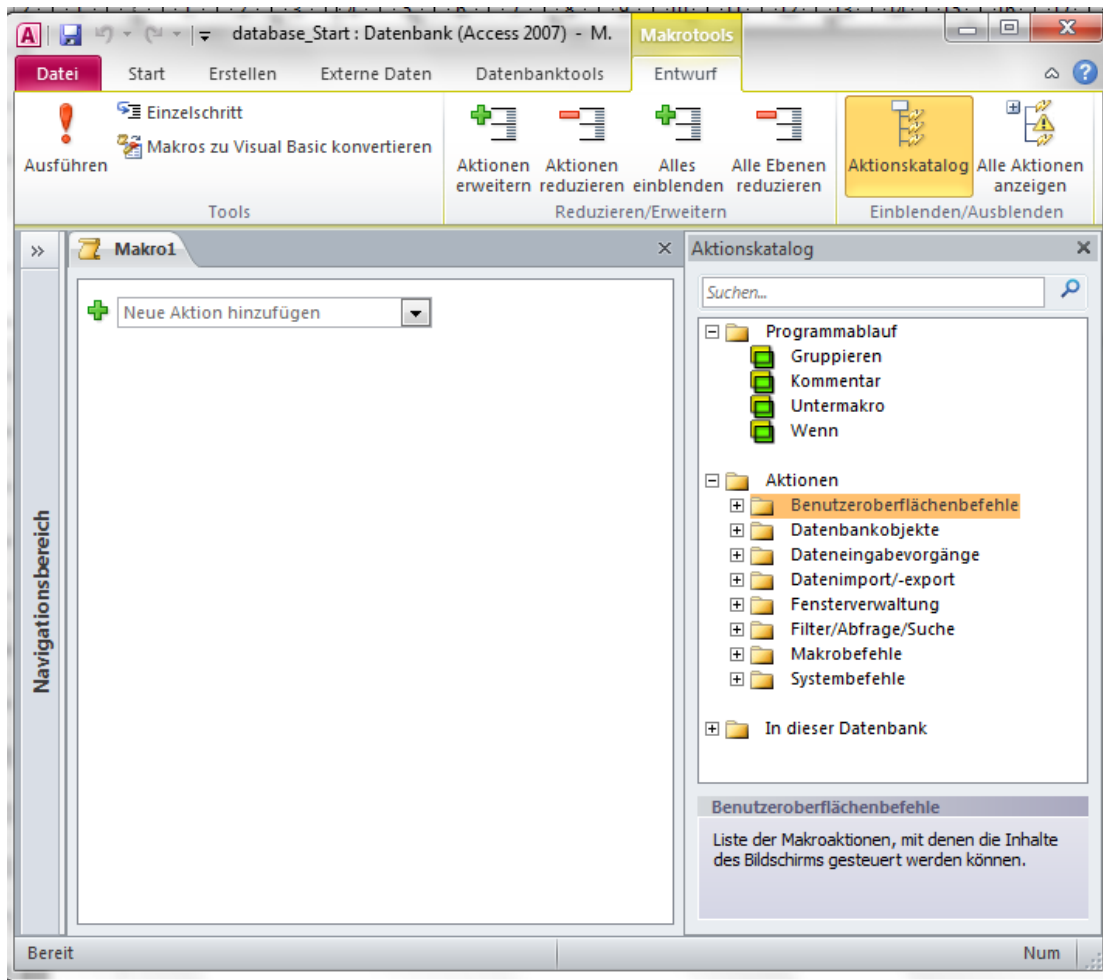
Makro-Objekte ...

- konfigurieren die Start-Einstellungen einer Datenbank.
- werden im Navigationsbereich als eigenständige Gruppe angezeigt.
- können über den Navigationsbereich gestartet werden.
- können von vielen verschiedenen Ereignissen in Formularen, Berichten und Steuerelementen gebunden werden.

... erstellen

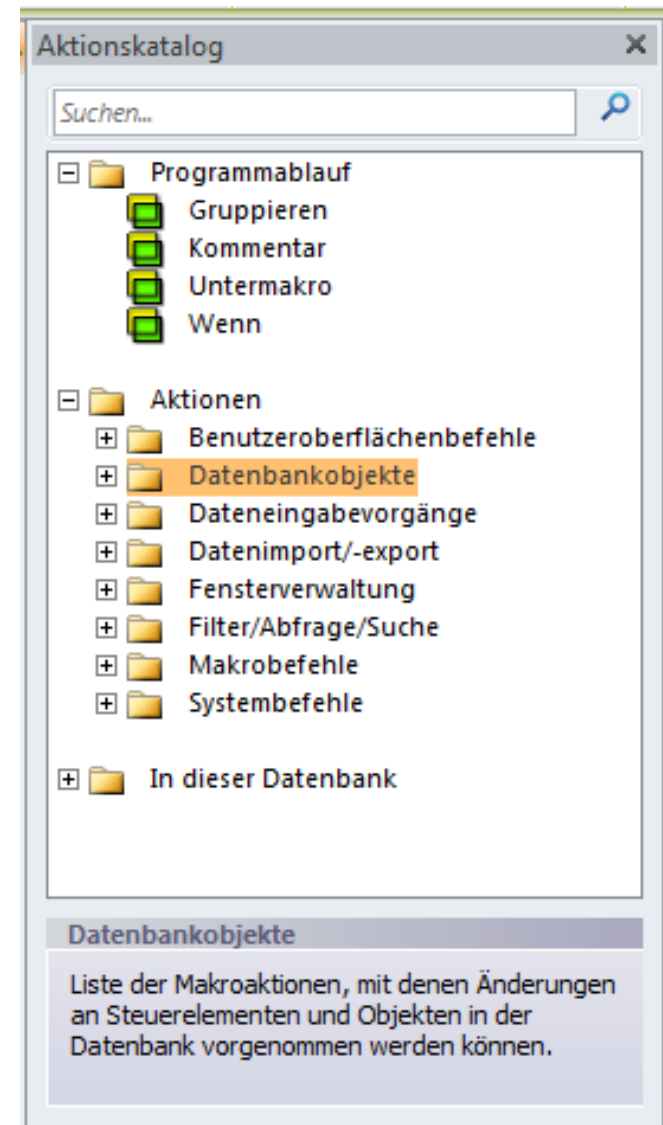
- Das Menüband *Erstellen* ist eingeblendet.
- Mit Hilfe des Befehls *Makro* in dem Bereich Makros und Code wird die Entwurfsansicht des Makros geöffnet.
- In der Entwurfsansicht werden die Aktionen zeilenweise abgelegt.

Entwurfsansicht eines neuen Makros



Aktionskatalog ...

- enthält alle in Access 2010 implementierten Aktionen. Aktionen, die Daten in Steuerelemente oder Objekte verändern, sind standardmäßig ausgeblendet.
- Im Ordner *Programmablauf* werden Elemente zur Strukturierung des Makros angeboten.
- Im Ordner *Aktionen* werden Aktionen in verschiedenen Kategorien angeboten.
- Zu einer markierten Aktion wird am unteren Rand eine Hilfe angeboten.

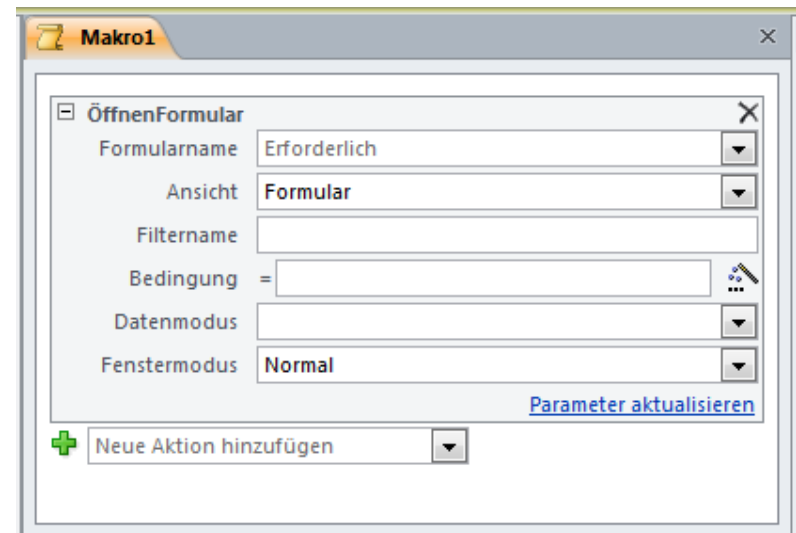


Aktionen mit Hilfe des Aktionskatalogs einfügen

- Mit einem Mausklick auf das Pluszeichen wird der, zur Aktion passende Ordner geöffnet.
- Anschließend kann mit einem Mausklick die gewünschte Aktion ausgewählt werden.
- Die Aktion wird mit Hilfe der gedrückt gehaltenen linken Maustaste in den Entwurfsbereich gezogen.
- Nach dem Loslassen der Maustaste wird die gewählte Aktion am Ende des Makros oder an der farblich markierten Position eingefügt.

Argumente für eine Aktion eingeben

- Die Aktion ist grau hinterlegt. Die Aktion ist aktiv. Durch ein Klick auf den Namen der Aktion wird diese aktiviert.
- In die Textfelder wird ein passender Wert mit Hilfe der Tastatur eingegeben.
- Mit Hilfe des schwarzen Pfeils nach unten, kann ein Wert aus einer Liste ausgewählt werden.
- Mit Hilfe des Zauberstabs wird ein Assistent für die Eingabe geöffnet.



Makro-Objekte in VBA konvertieren

- Das Makro ist in der Entwurfsansicht geöffnet und gespeichert.
- Das Menüband *Entwurf* der Makrotools ist eingeblendet.
- Mit einem Klick auf den Befehl *Makros zu Visual Basic konvertieren* im Bereich Tools wird ein Dialogfenster geöffnet. Mit Hilfe dieses Dialogfenster kann eine Fehlerbehandlung sowie Kommentare automatisiert hinzugefügt werden. Mit Hilfe der Schaltfläche *Konvertieren* wird die Umwandlung gestartet.
- Nach Abschluss der Konvertierung wird der Code angezeigt.

Visual Basic for Application (VBA) ...

- ist eine Programmiersprache, die in jeder Office-Anwendung eingebettet ist.
- erweitert die Funktionalität von Office-Anwendungen.
- passt eine Anwendung entsprechend der Wünsche der Benutzer an.
- automatisiert Befehlsreihenfolgen.

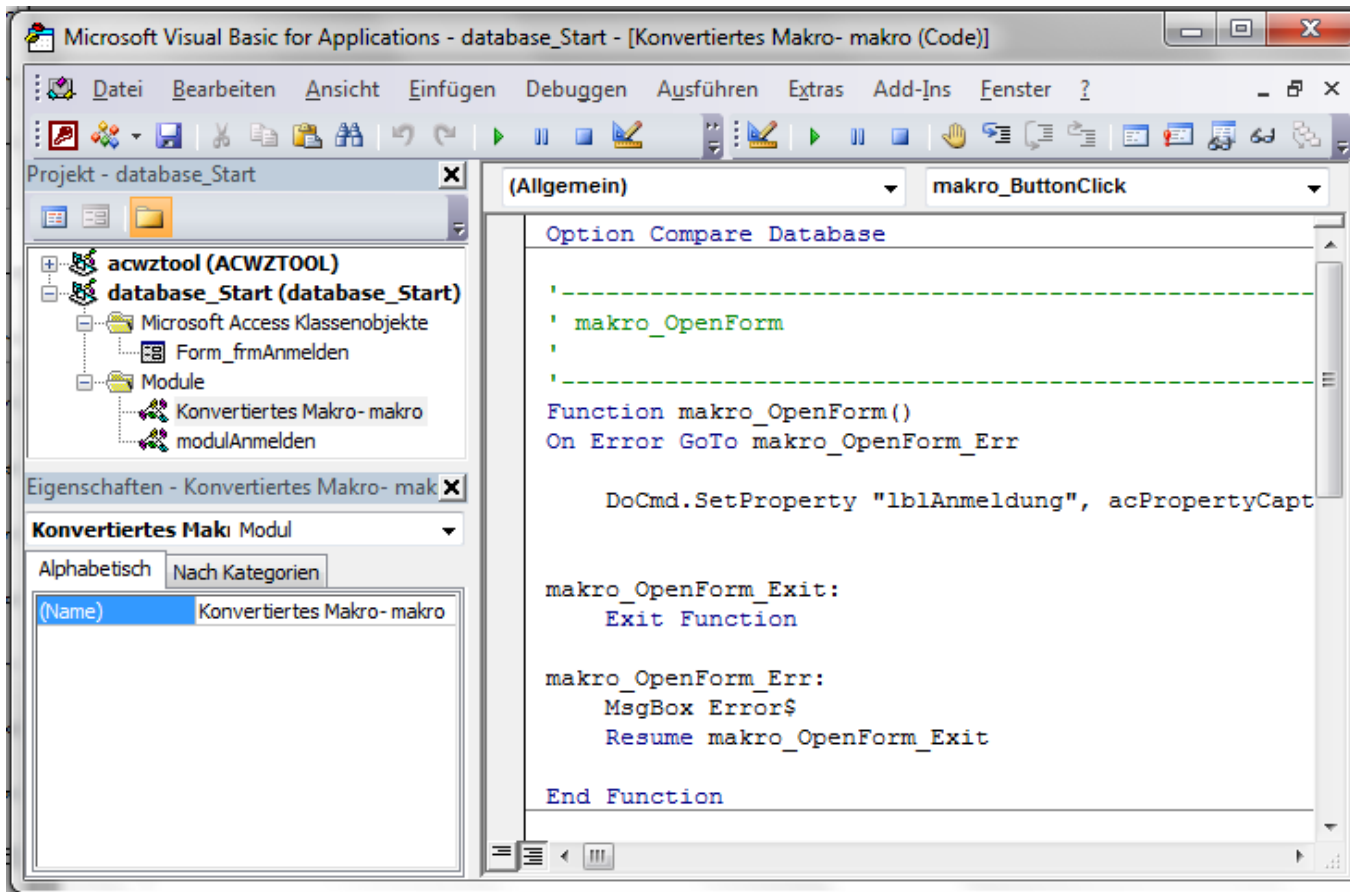
Programmiersprache ...

- hat wie jede Sprache eine bestimmte Syntax.
- hat Schlüsselwörter, die bestimmte Aktionen beschreiben.
- definiert Eigenschaften und Funktionen von Objekten. Objekte sind zum Beispiel Tabellen oder Formulare in Access.
- wird in einer integrierten Entwicklungsumgebung eingegeben.

Integrierte Entwicklungsumgebung (IDE) ...

- ist in jeder Office-Anwendung vorhanden.
- ist eine eigenständige Anwendung, die in der Taskleiste als Symbol eingeblendet wird.
- bietet die Möglichkeit VBA-Code zu lesen und zu bearbeiten.

Beispiel



Titelleiste ...

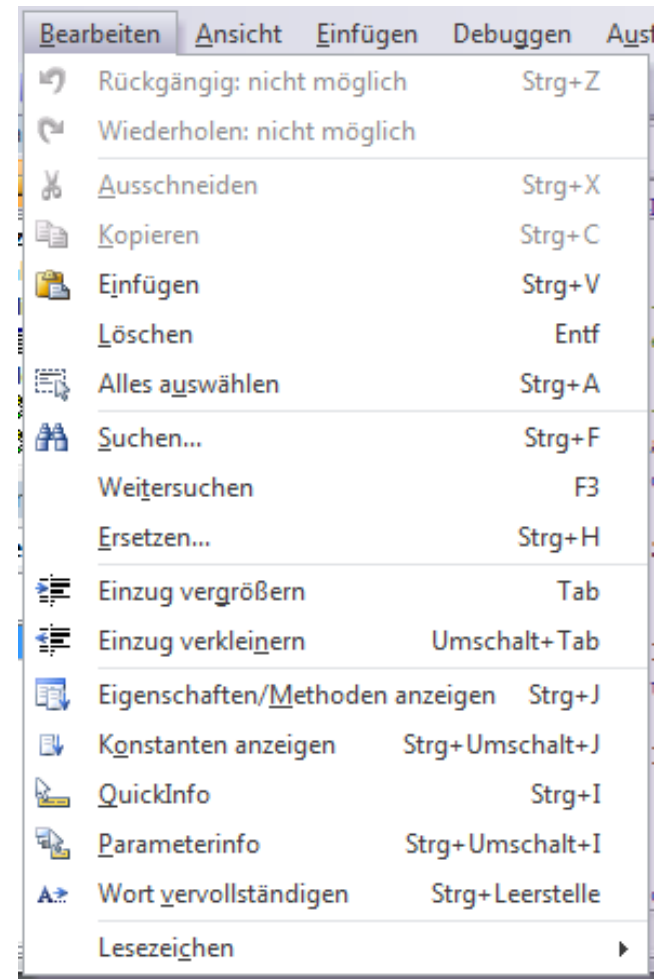
- befindet sich am oberen Rand der Anwendung.
- hat in der linken Ecke ein Icon, welches die Anwendung symbolisiert. Mit einem Klick auf das Icon wird das dazugehörige Systemmenü geöffnet.
- zeigt den Namen der Datenbank sowie den Namen des aktiven Moduls an.
- hat am rechten Rand Schaltflächen zum Minimieren, Verkleinern / Maximieren oder Schließen der Anwendung.

Menüleiste ...

- befindet sich unterhalb der Titelleiste.
- sammelt alle Befehle der Anwendung
- bietet aufklappbare Menüs zu verschiedenen Themen an. In diesen Menüs werden Befehle und Funktionen der Anwendung angezeigt.
- zeigt auf der obersten Ebene Kategorien an. In Abhängigkeit dieser Kategorien werden die Befehle zusammengefasst. Der Name der Kategorie gibt einen ersten Hinweis auf die Benutzung der darin enthaltenen Befehle.

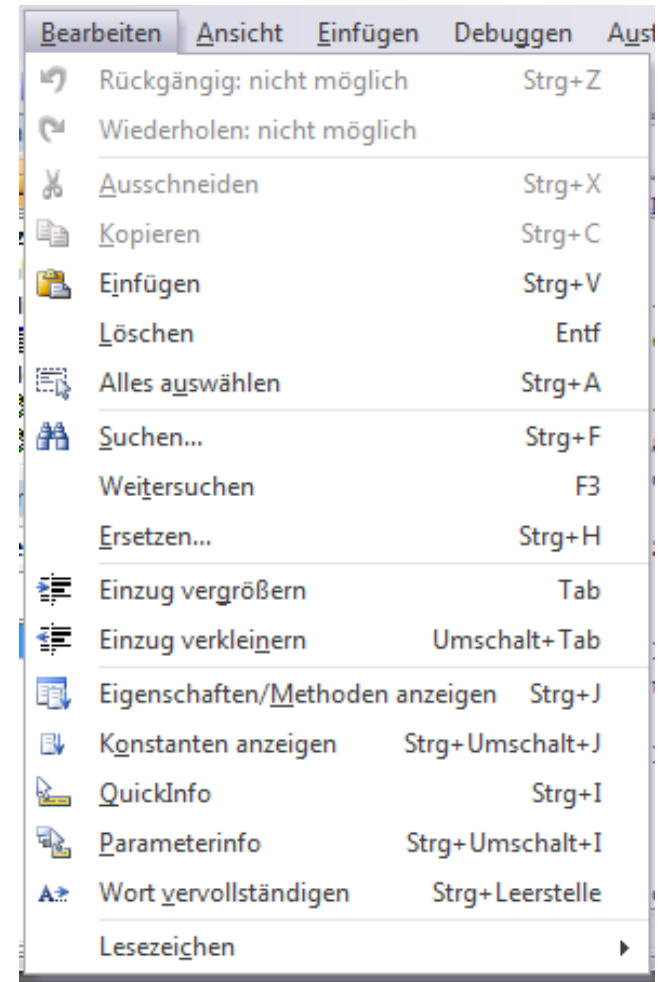
Menü öffnen

- Mit einem Mausklick auf eine Kategorie in der Menüleiste wird das passende Menü geöffnet.
- In dem Menü werden alle Befehle, die zu dem Thema gehören, aufgelistet.



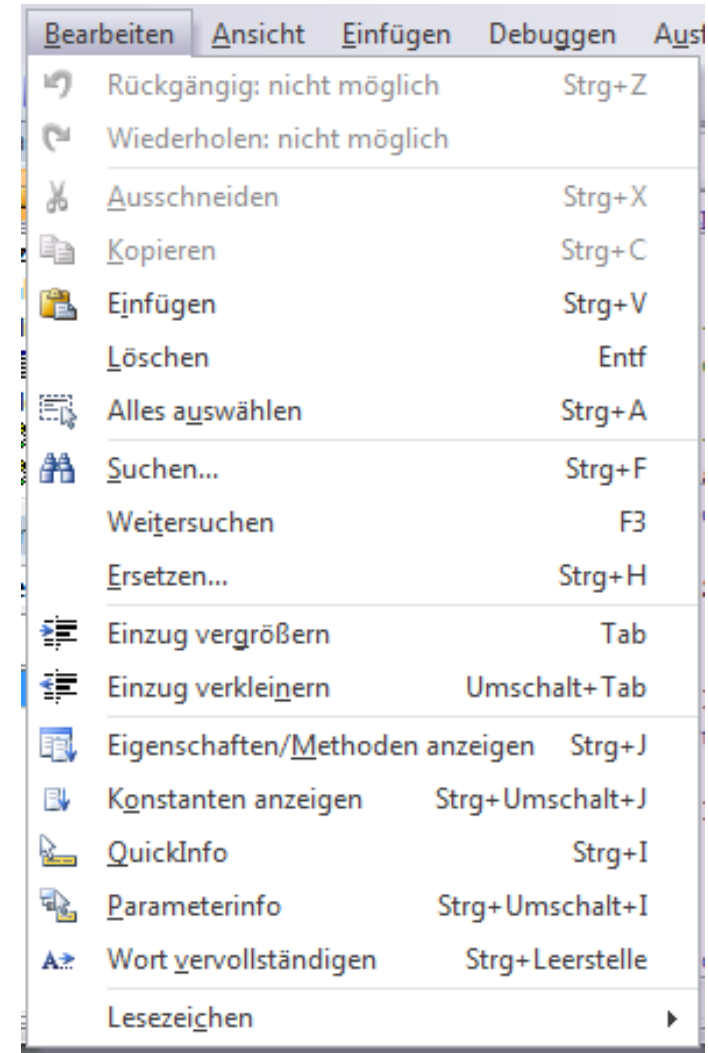
Befehle im Menü ...

- starten eine Aktion. Häufig werden diese durch ein Symbol am linken Rand und / oder einem Tastaturkürzel am rechten Rand gekennzeichnet
- öffnen Symbolleisten und Fenster. Ein Häkchen am linken Rand kennzeichnet ein eingeblendetes Element.
- öffnen ein Menü. Mit Hilfe des schwarzen Pfeils am rechten Rand werden die untergeordneten Menüs gekennzeichnet.



Aktionen im Menü ...

- werden mit Hilfe des Symbols am linken Rand in den Symbolleisten gekennzeichnet.
- können mit Hilfe des Tastaturkürzels am rechten Rand gestartet werden.
- können in Abhängigkeit einer Aktion oder Elements eingeblendet werden. Inaktive Aktionen sind grau hinterlegt.



Symbolleisten ...

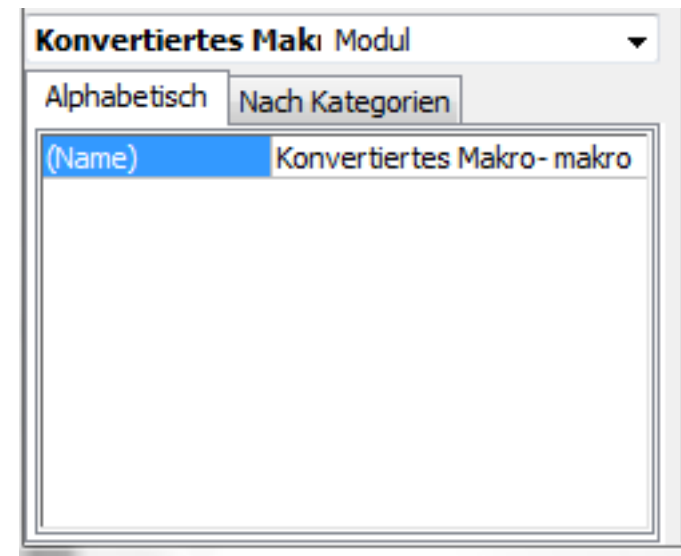
- zeigen häufig genutzte Befehle zu einem Thema mit Hilfe eines Icons an.
- beginnen mit der senkrechten gestrichelten Linie am linken Rand. Mit Hilfe der gedrückt gehaltenen Maustaste über diese Linie kann die Symbolleiste verschoben werden.
- werden über das Menü *Ansicht – Symbolleiste* ein- oder ausgeblendet.
- haben einen Pfeil nach unten am rechten Rand. Mit einem Klick auf den Pfeil wird ein Menü zum Ein- und Ausblenden von Icons in der Symbolleiste eingeblendet.

Befehle ausführen

- Über die Menüleiste:
 - Die passende Kategorie wird geöffnet.
 - Mit einem Mausklick auf die Bezeichnung des Befehls wird dieser gestartet.
- Über die Symbolleiste:
 - Mit einem Klick auf passende Icon wird der Befehl gestartet.
- Andere Möglichkeit:
 - Das passende Tastaturkürzel wird eingegeben.

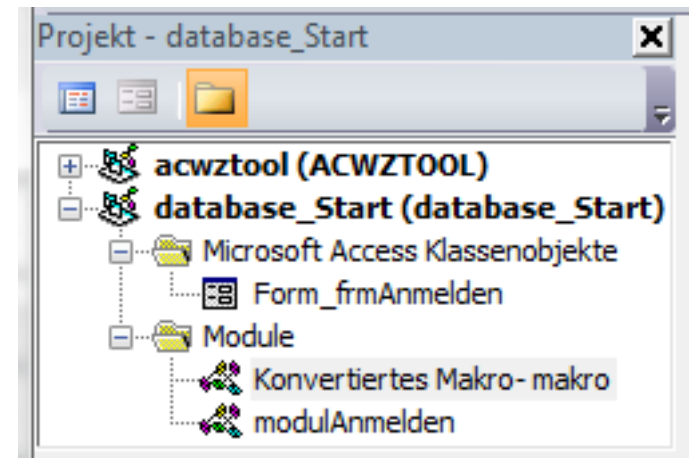
Eigenschaftfenster ...

- zeigt den Namen eines Moduls an.
- wird in Access selten benötigt.



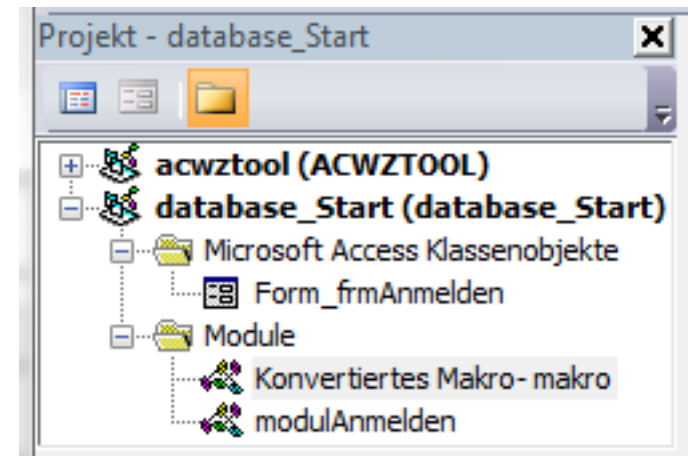
Projekt-Explorer ...

- zeigt die Codemodule einer Datenbank an.
- verwaltet die Komponenten einer Datenbank im Bereich „Programmierung“.
- ist die Schaltzentrale eines VBA-Projekts.
- ist frei platzierbar.
- kann über das Menü *Ansicht* eingeblendet werden.

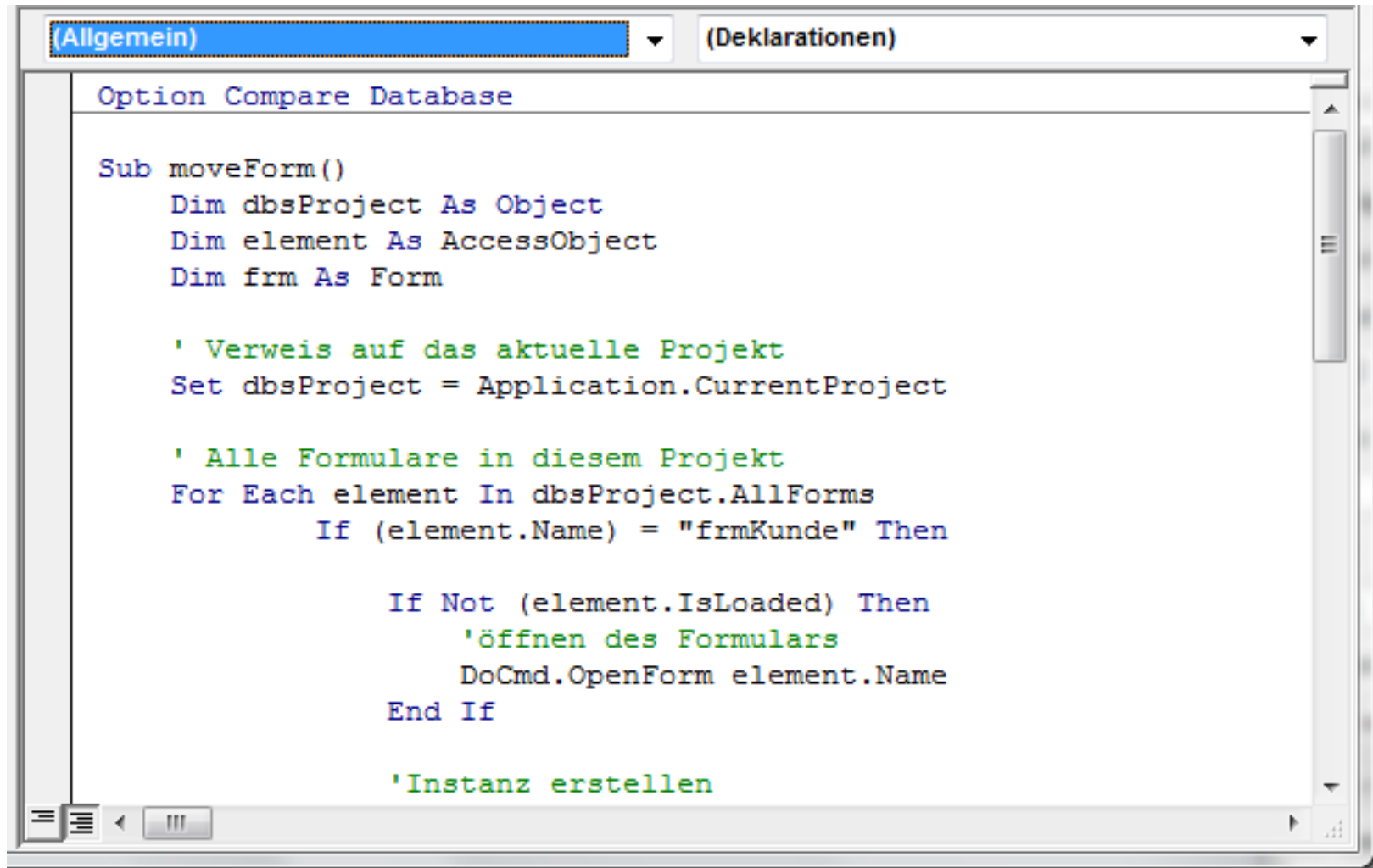


Aufbau

- In der Titelleiste wird der Name der Datenbank sowie die Schließen-Schaltfläche angezeigt.
- Darunter befindet sich die Symbolleiste mit den Icons
 - „Zeige zum gewählten Element den Code an“.
 - „Zeige das passende Objekt zum Code an“.
 - „Sortierung mit Hilfe von Ordnern“.
- Unterhalb der Symbolleiste wird der Code alphabetisch oder in Abhängigkeit von Ordnern sortiert angezeigt.



Code-Fenster



```
Option Compare Database

Sub moveForm()
    Dim dbsProject As Object
    Dim element As AccessObject
    Dim frm As Form

    ' Verweis auf das aktuelle Projekt
    Set dbsProject = Application.CurrentProject

    ' Alle Formulare in diesem Projekt
    For Each element In dbsProject.AllForms
        If (element.Name) = "frmKunde" Then

            If Not (element.IsLoaded) Then
                'öffnen des Formulars
                DoCmd.OpenForm element.Name
            End If

            'Instanz erstellen
        End If
    End For
End Sub
```

... einblenden

- *Ansicht – Codefenster.*
- Doppelklick auf eines der Code-Module im Projekt-Explorer.
Das Code-Fenster wird mit dem dazugehörigen Code geöffnet.

Objekt-Kombinationsfeld ...

- befindet sich am linken, oberen Rand des Codefensters.
- enthält das Element (Allgemein). Für Standardmodule ist dies der einzige Eintrag.
- listet den Eintrag Form für das dazugehörige Formular, Report für den dazugehörigen Bericht sowie die Namen, der in dem Formular oder Bericht enthaltenen Steuerelemente auf.

Prozedur-Kombinationsfeld ...

- befindet sich am rechten oberen Rand des Codefensters.
- zeigt alle Prozeduren zu den, im linken Kombinationsfeld gewählten Objekt an.
- listet alle Ereignisse zu dem gewählten Objekt auf.
- hat immer den Eintrag (Deklaration). Dieser Eintrag enthält Anweisungen, die das gesamte Modul betreffen. Die Anweisungen befinden sich meist am Anfang des Moduls im Codebereich.

Statuszeile des Code-Fensters

- Schaltfläche *Prozeduransicht* (links). Es wird immer nur eine Prozedur angezeigt.
- Schaltfläche *Vollständige Modulansicht* (rechts). Der Code in dem Modul wird vollständig angezeigt.

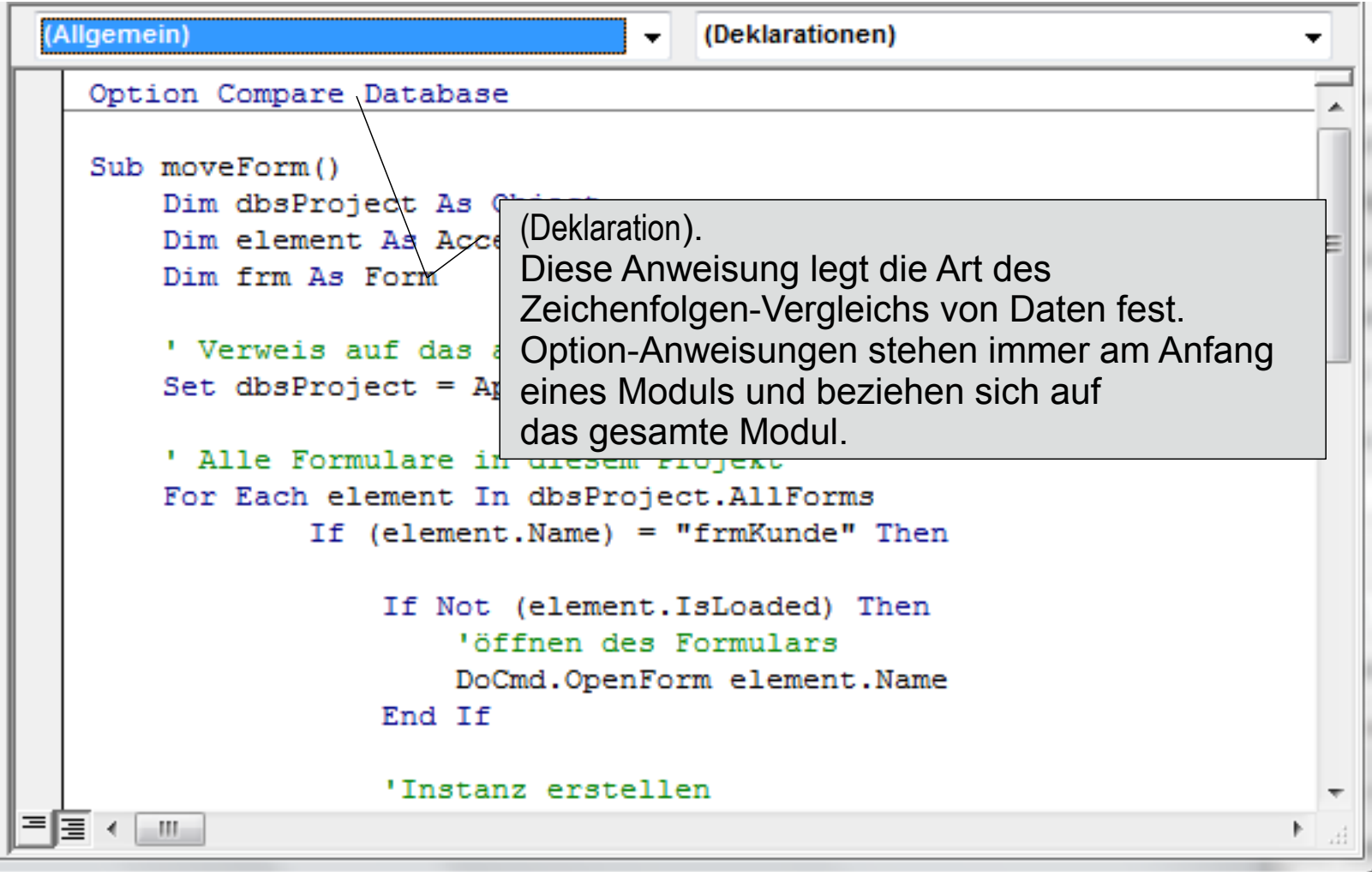
Module ...

- sind Container für Code zu einem Thema.
- entsprechen einem Makro aus dem Navigationsbereich.
- können vom Entwickler geschrieben werden.
- werden automatisch durch die Anlage einer Ereignisprozedur im Eigenschaftfenster eines Formulars, Berichts oder Steuerelements erzeugt.

Modul-Typen

- Formular- oder Berichtsmodule werden automatisch durch das Erstellen einer Ereignisprozedur erzeugt. In diesen Modulen wird alles gespeichert, was zur Benutzeroberfläche gehört. Das Modul wird automatisch durch das Entfernen des Formulars oder Berichts gelöscht.
- Standardmodule können von überall aufgerufen werden. Diese Art von Modulen enthält Code, der in der gesamten Datenbank genutzt wird. In diesen Modulen werden Berechnungen etc. gespeichert. Jede Subroutine in dem Modul ist eigenständig und kann für sich getestet werden.
- Klassenmodule enthalten benutzerdefinierte Objektbeschreibungen. In diesen Modulen können eigene Objekte wie zum Beispiel Person definiert werden.

Deklarationen in einem Modul



(Allgemein) (Deklarationen)

```
Option Compare Database

Sub moveForm()
    Dim dbsProject As Database
    Dim element As AccessForm
    Dim frm As Form

    ' Verweis auf das aktuelle Projekt
    Set dbsProject = CurrentDatabase

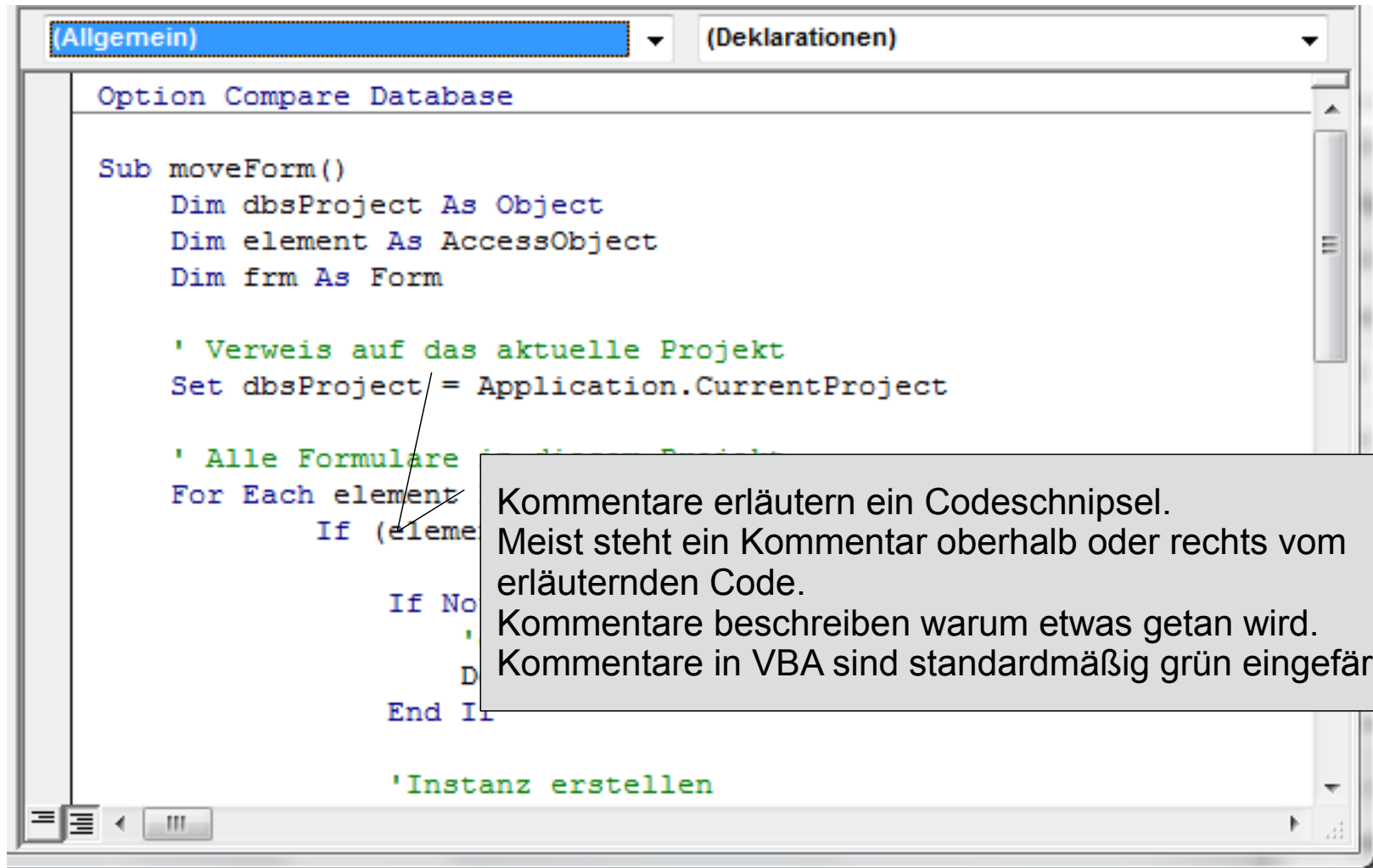
    ' Alle Formulare in diesem Projekt
    For Each element In dbsProject.AllForms
        If (element.Name) = "frmKunde" Then

            If Not (element.IsLoaded) Then
                'öffnen des Formulars
                DoCmd.OpenForm element.Name
            End If

            'Instanz erstellen
        End If
    End For
End Sub
```

(Deklaration).
Diese Anweisung legt die Art des Zeichenfolgen-Vergleichs von Daten fest. Option-Anweisungen stehen immer am Anfang eines Moduls und beziehen sich auf das gesamte Modul.

Kommentare



```
(Allgemein) (Deklarationen)
Option Compare Database

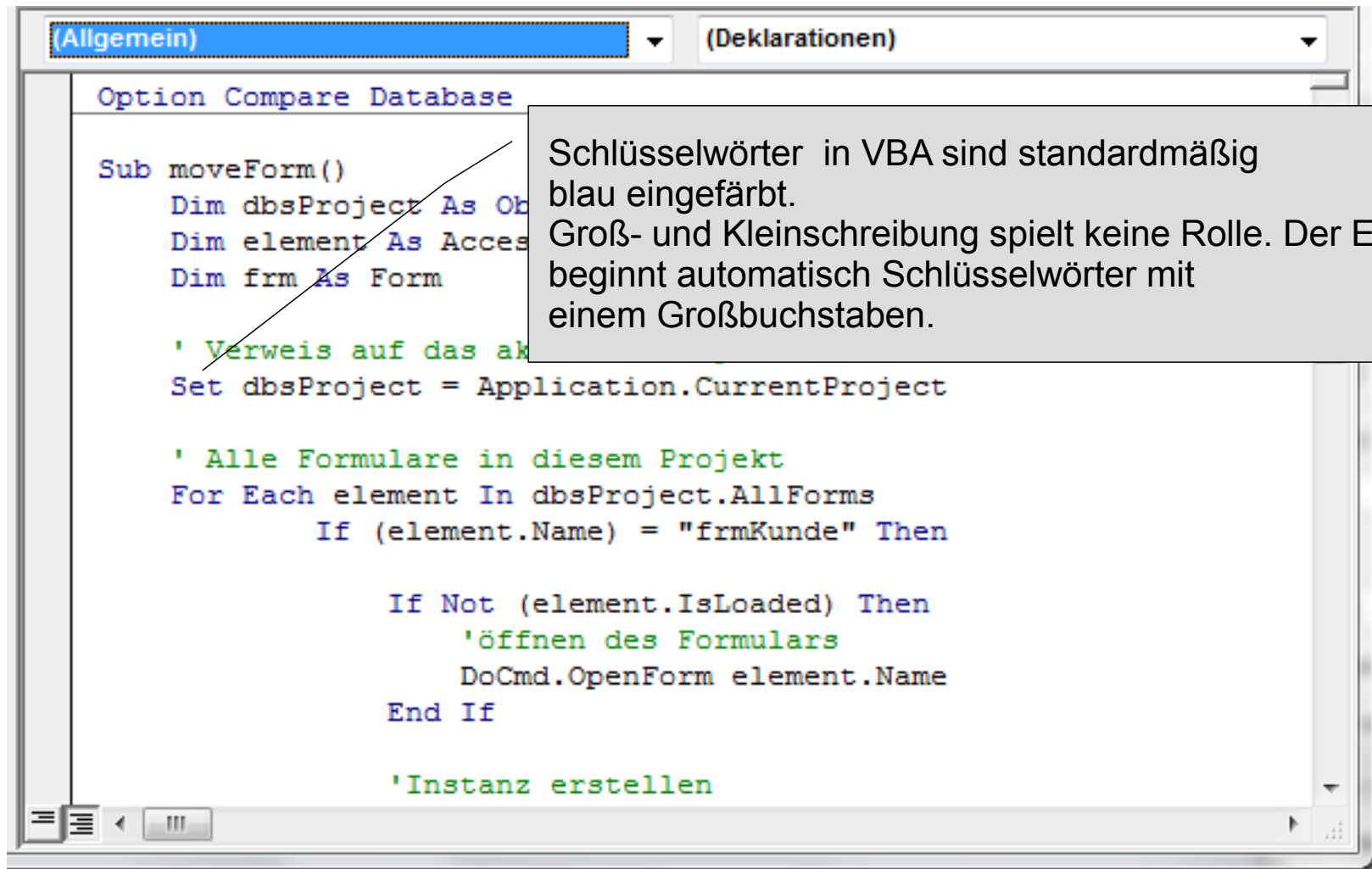
Sub moveForm()
  Dim dbsProject As Object
  Dim element As AccessObject
  Dim frm As Form

  ' Verweis auf das aktuelle Projekt
  Set dbsProject = Application.CurrentProject

  ' Alle Formulare im aktuellen Projekt
  For Each element In dbsProject.Forms
    If (element.Name = "Form1") Then
      ' Instanz erstellen
    End If
  End For
End Sub
```

Kommentare erläutern ein Codeschnipsel.
Meist steht ein Kommentar oberhalb oder rechts vom erläuternden Code.
Kommentare beschreiben warum etwas getan wird.
Kommentare in VBA sind standardmäßig grün eingefärbt.

Schlüsselwörter



```
(Allgemein) (Deklarationen)
Option Compare Database

Sub moveForm()
  Dim dbsProject As Database
  Dim element As AccessForm
  Dim frm As Form

  ' Verweis auf das aktuelle Projekt
  Set dbsProject = Application.CurrentProject

  ' Alle Formulare in diesem Projekt
  For Each element In dbsProject.AllForms
    If (element.Name) = "frmKunde" Then

      If Not (element.IsLoaded) Then
        'öffnen des Formulars
        DoCmd.OpenForm element.Name
      End If

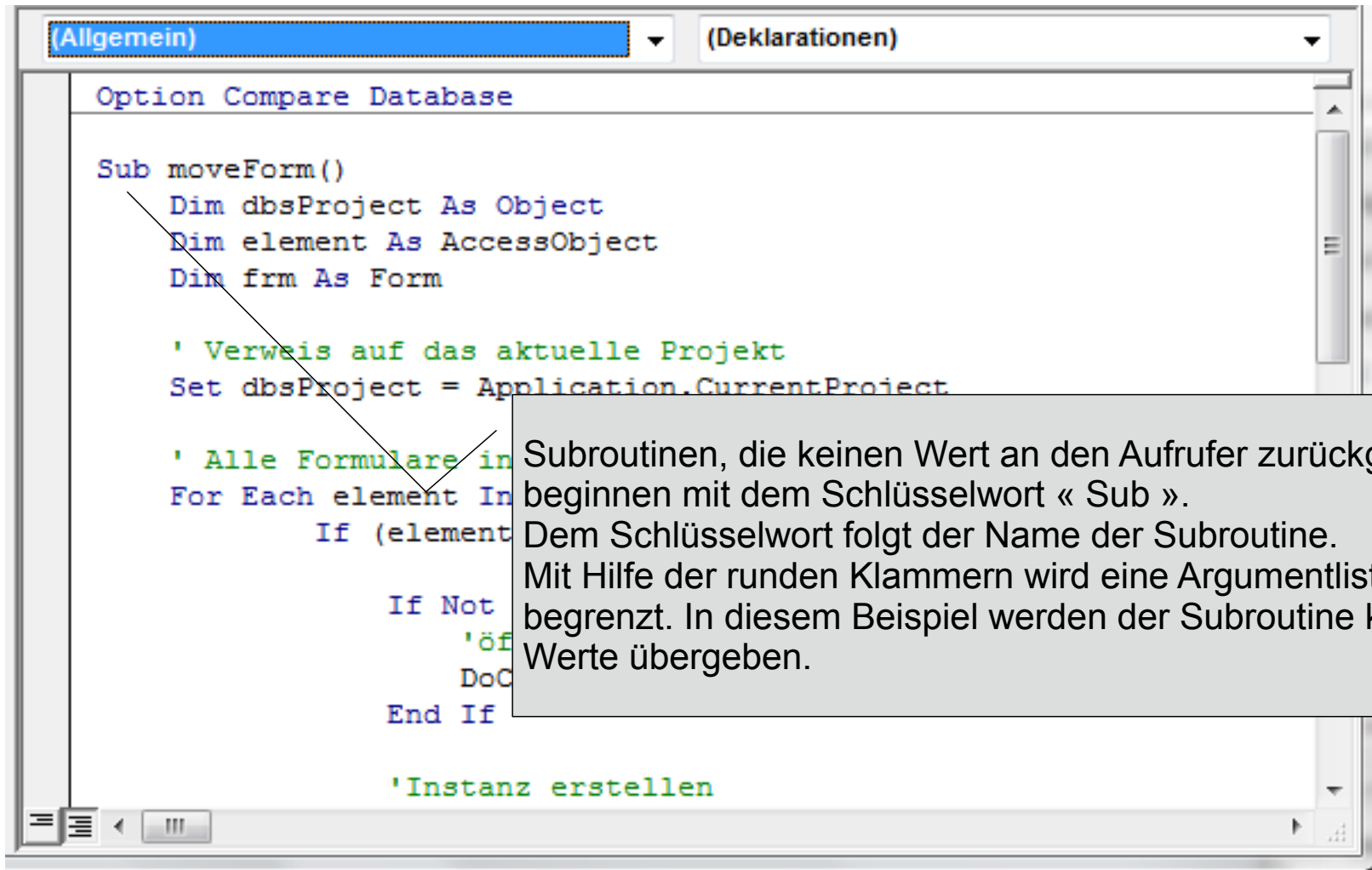
      'Instanz erstellen
    End If
  End For
End Sub
```

Schlüsselwörter in VBA sind standardmäßig blau eingefärbt. Groß- und Kleinschreibung spielt keine Rolle. Der Editor beginnt automatisch Schlüsselwörter mit einem Großbuchstaben.

Subroutinen ...

- unterteilen Module in verschiedene Kapitel.
- implementieren eine bestimmte Aufgabe.
- können Argumente haben. Argumente werden der Subroutine übergeben und in dieser verarbeitet.
- die keinen Wert an den Aufrufer zurückgeben, werden als Prozeduren bezeichnet.
- die einen Wert an den Aufrufer zurückgeben, werden als Funktionen bezeichnet.

Prozeduren



The screenshot shows the VBA editor window with the 'Allgemein' (General) tab selected. The code defines a subroutine 'moveForm()' with several declarations and comments. A callout box points to the 'Sub' keyword in the code.

```
Option Compare Database

Sub moveForm()
    Dim dbsProject As Object
    Dim element As AccessObject
    Dim frm As Form

    ' Verweis auf das aktuelle Projekt
    Set dbsProject = Application.CurrentProject

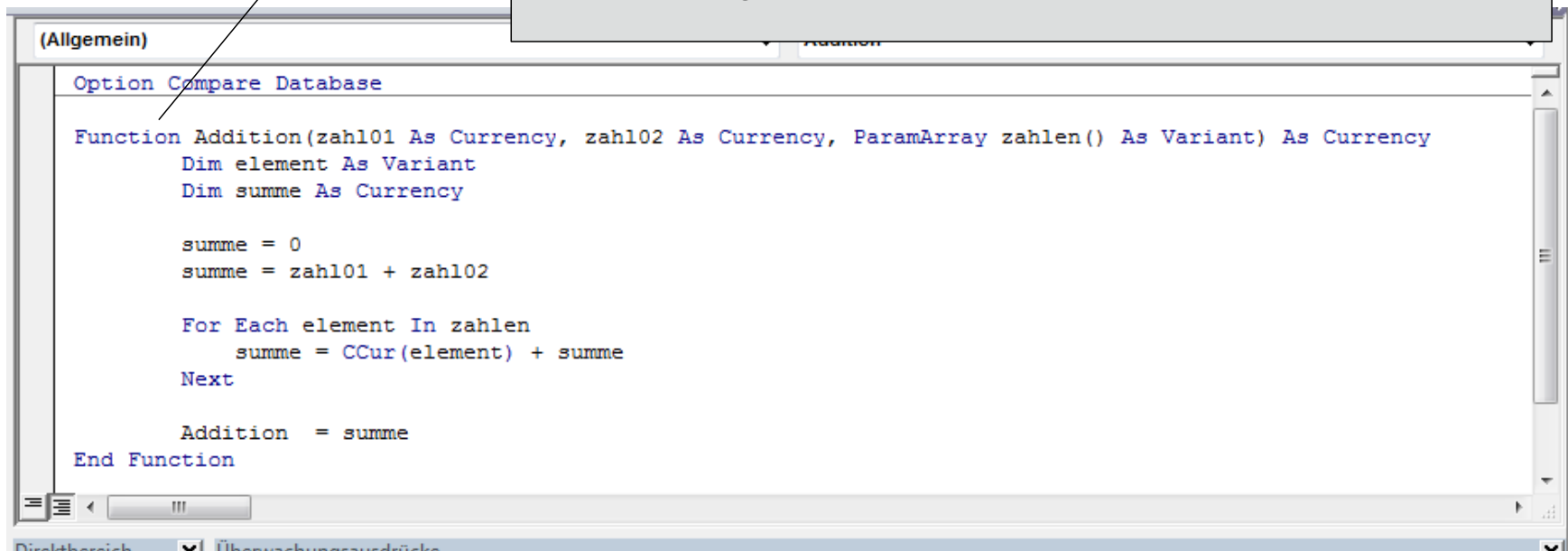
    ' Alle Formulare in
    For Each element In
        If (element
            If Not
                'öf
                DoC
            End If

    'Instanz erstellen
```

Subroutinen, die keinen Wert an den Aufrufer zurückgeben, beginnen mit dem Schlüsselwort « Sub ». Dem Schlüsselwort folgt der Name der Subroutine. Mit Hilfe der runden Klammern wird eine Argumentliste begrenzt. In diesem Beispiel werden der Subroutine keine Werte übergeben.

Funktionen

Subroutinen, die einen Wert an den Aufrufer zurückgeben, beginnen mit dem Schlüsselwort « Function ». Dem Schlüsselwort folgt der Name der Subroutine. Mit diesen Namen wird die Funktion aufgerufen. Dem Namen wird ein Rückgabewert mit Hilfe des Gleichheitszeichens zugewiesen.



```
(Allgemein)
Option Compare Database

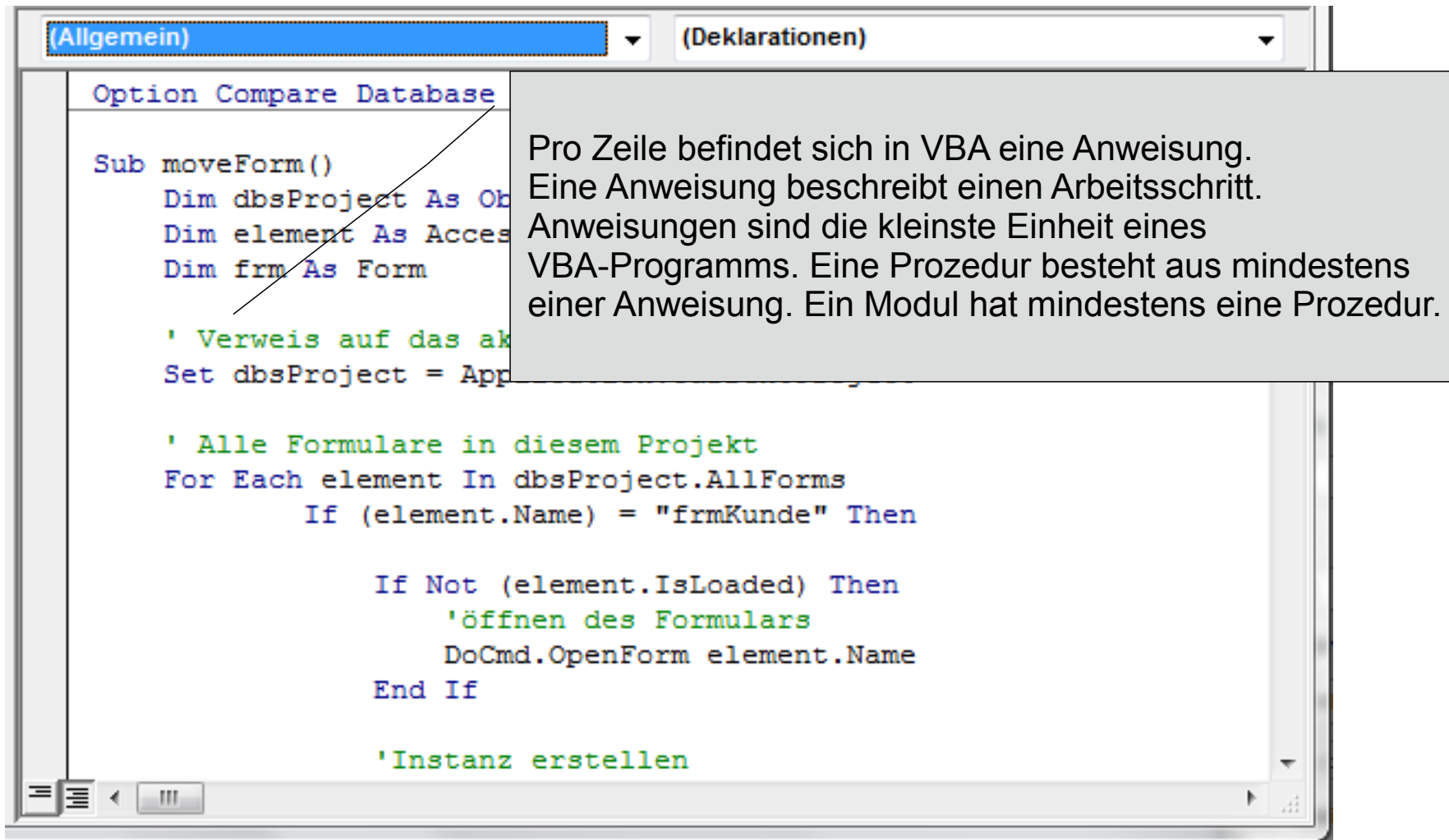
Function Addition(zahl01 As Currency, zahl02 As Currency, ParamArray zahlen() As Variant) As Currency
    Dim element As Variant
    Dim summe As Currency

    summe = 0
    summe = zahl01 + zahl02

    For Each element In zahlen
        summe = CCur(element) + summe
    Next

    Addition = summe
End Function
```

Anweisungen



```
(Allgemein) (Deklarationen)
Option Compare Database

Sub moveForm()
  Dim dbsProject As Object
  Dim element As AccessForm
  Dim frm As Form

  ' Verweis auf das aktuelle Projekt
  Set dbsProject = Application.CurrentDatabase

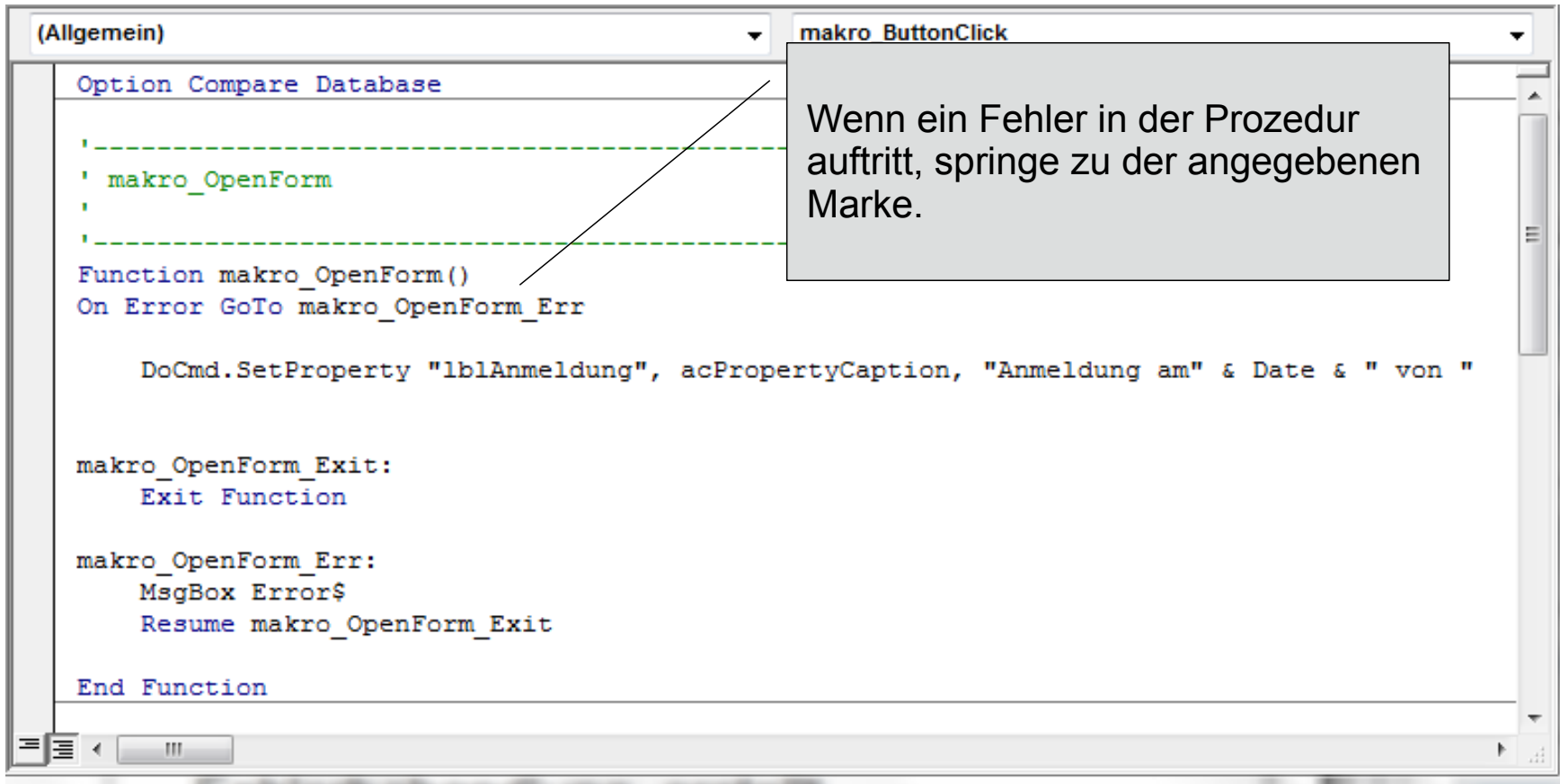
  ' Alle Formulare in diesem Projekt
  For Each element In dbsProject.AllForms
    If (element.Name) = "frmKunde" Then

      If Not (element.IsLoaded) Then
        'öffnen des Formulars
        DoCmd.OpenForm element.Name
      End If

      'Instanz erstellen
    End If
  End For
End Sub
```

Pro Zeile befindet sich in VBA eine Anweisung. Eine Anweisung beschreibt einen Arbeitsschritt. Anweisungen sind die kleinste Einheit eines VBA-Programms. Eine Prozedur besteht aus mindestens einer Anweisung. Ein Modul hat mindestens eine Prozedur.

Fehlerbehandlung



The screenshot shows the VBA editor for a macro named 'makro ButtonClick'. The code is as follows:

```
(Allgemein)
Option Compare Database

'-----
' makro_OpenForm
'-----

Function makro_OpenForm()
On Error GoTo makro_OpenForm_Err

    DoCmd.SetProperty "lblAnmeldung", acPropertyCaption, "Anmeldung am" & Date & " von "

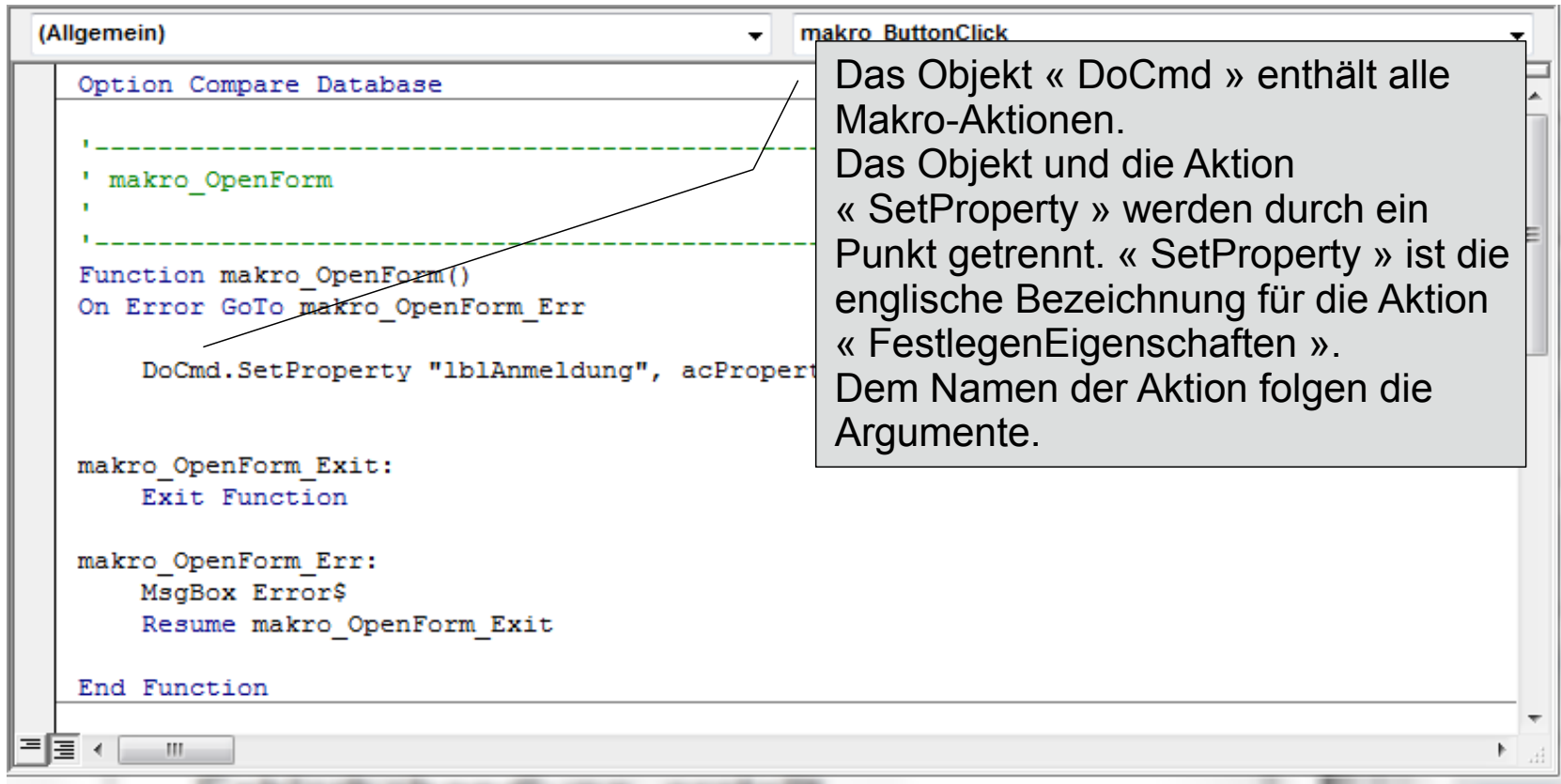
makro_OpenForm_Exit:
    Exit Function

makro_OpenForm_Err:
    MsgBox Error$
    Resume makro_OpenForm_Exit

End Function
```

A callout box with a grey background and black border contains the text: "Wenn ein Fehler in der Prozedur auftritt, springe zu der angegebenen Marke." A line points from this box to the 'On Error GoTo' statement in the code.

Makro-Aktionen



(Allgemein) makro ButtonClick

```
Option Compare Database

'-----
' makro_OpenForm
'-----

Function makro_OpenForm()
On Error GoTo makro_OpenForm_Err

    DoCmd.SetProperty "lblAnmeldung", acProperty

makro_OpenForm_Exit:
    Exit Function

makro_OpenForm_Err:
    MsgBox Error$
    Resume makro_OpenForm_Exit

End Function
```

Das Objekt « DoCmd » enthält alle Makro-Aktionen. Das Objekt und die Aktion « SetProperty » werden durch ein Punkt getrennt. « SetProperty » ist die englische Bezeichnung für die Aktion « FestlegenEigenschaften ». Dem Namen der Aktion folgen die Argumente.

Code verändern

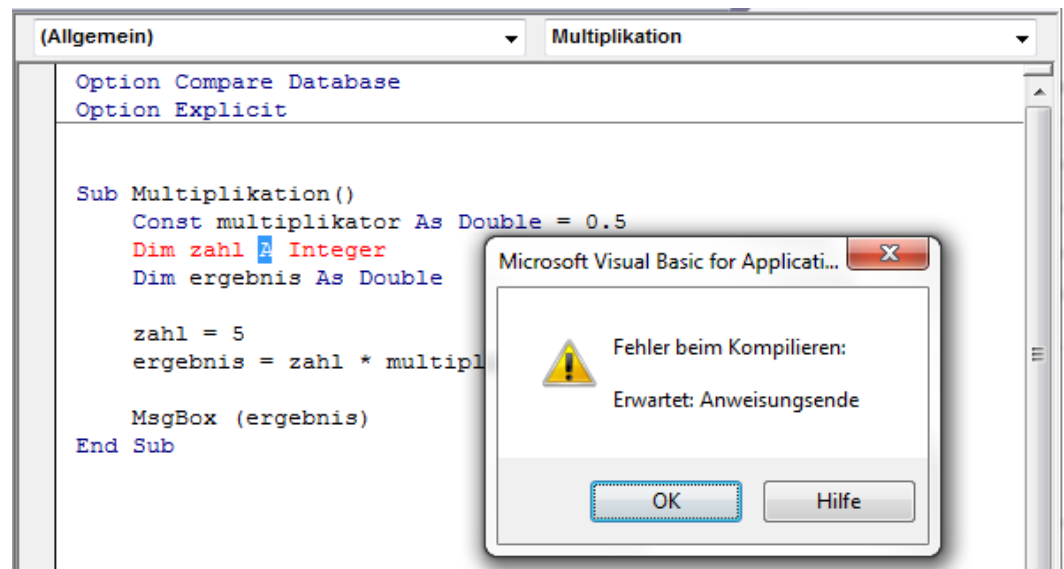
- Mit einem Mausklick wird die Einfügemarke an einer bestimmten Position eingeblendet. An dieser Position kann Code mit Hilfe der Tastatur eingegeben werden.
- Mit Hilfe der gedrückt gehaltenen Maustaste kann Code markiert werden. Der markierte Code kann mit <ENTF> gelöscht oder überschrieben werden.

... speichern

- Mit Hilfe von <STRG>+<S> wird der Code gespeichert.
- Bei einer erstmaligen Speicherung muss ein Name für das Modul eingegeben werden. Vergeben Sie einen Namen entsprechend der gewünschten Sammlung von Code.

Syntaxfehler ...

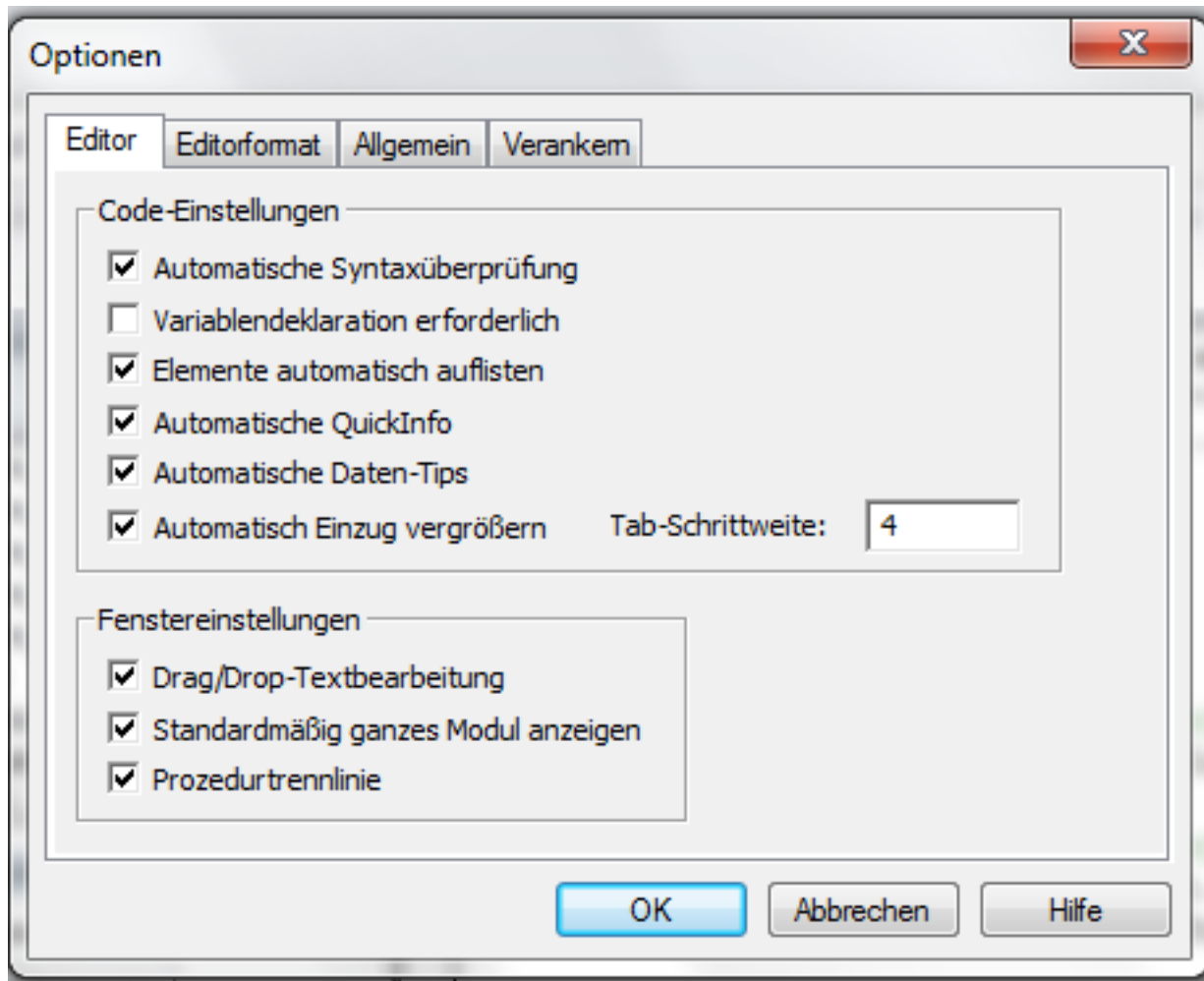
- sind zum Beispiel Schreibfehler in einem Schlüsselwort.
- verhindern das Ausführen von Code.
- werden rot im Code-Fenster markiert.
- wird durch eine Fehlermeldung erläutert.



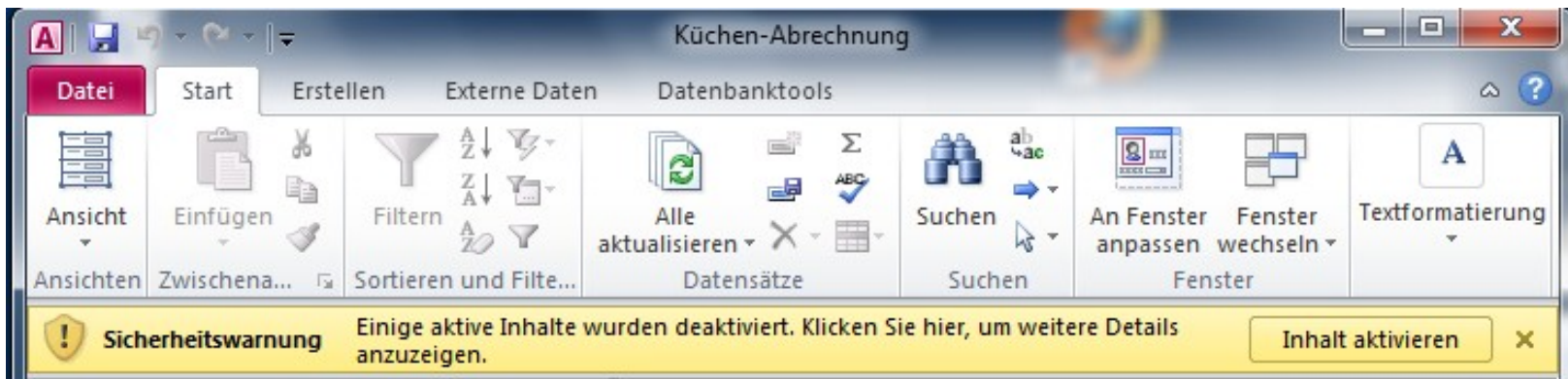
Einstellungsmöglichkeiten

- *Extras – Optionen.*
- Die Registerkarte *Editorformat* legt die die verschiedenen Schriftarten für den Text im Code-Fenster fest.
- *Verankern* legt die Ausrichtung der Fenster innerhalb des VBA-Editors untereinander fest.
- *Allgemein* legt das Verhalten bei Fehlern und die Rastergröße von UserForms fest. Die Registerkarte wird in Access selten genutzt.
- *Editor* legt das Aussehen und Verhalten des Editors fest.

Registerkarte „Editor“

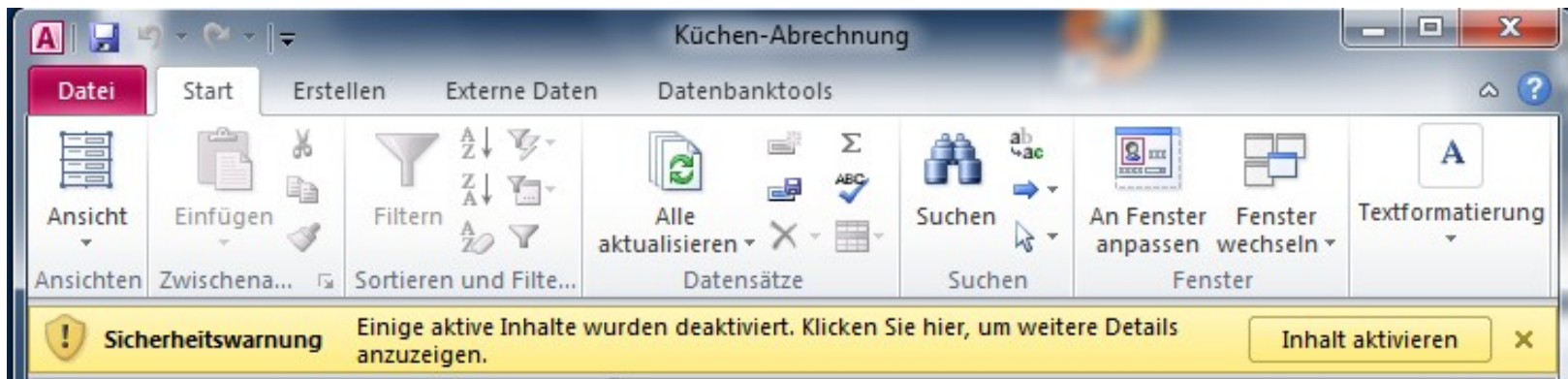


Sicherheitswarnung einer Datenbank ...



- wird durch Code / Programmelemente in der Datenbank ausgelöst.
- wird nach dem erstmaligen Öffnen der Datenbank angezeigt.
- wird unterhalb der Menüleiste angezeigt.
- kann deaktiviert werden. Die Datenbank und der darin enthaltene Code wird als sicher eingestuft.

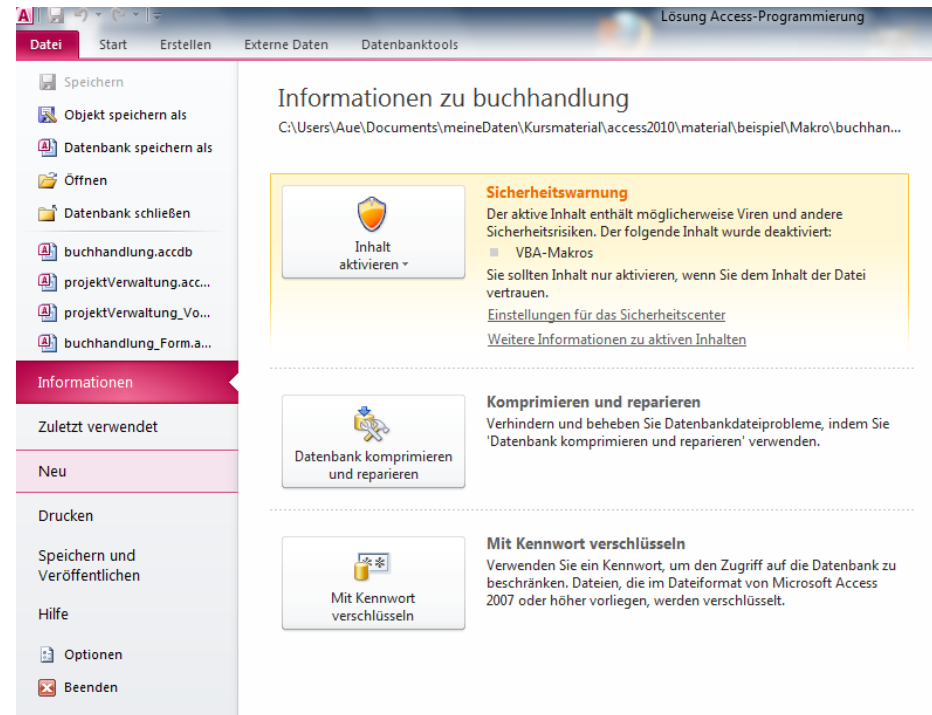
... deaktivieren



- Mit Hilfe der Schaltfläche *Inhalt aktivieren* wird die Warnung bestätigt und ausgeblendet.
- Die Datenbank und der darin enthaltene Code wird als sicher eingestuft.

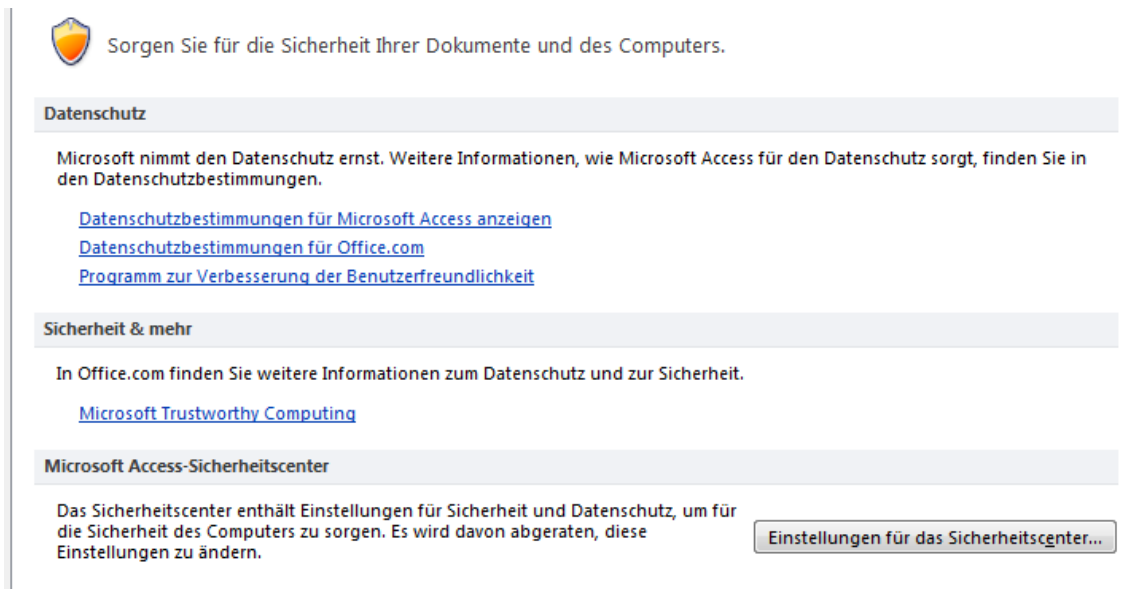
... in der Backstage-Ansicht ...


- *Datei – Informationen.*
- wird beim erstmaligen Öffnen eingeblendet.



Sicherheitscenter ...

- *Datei – Optionen – Sicherheitscenter.*
- zeigt Hinweise in Form von Textlinks an.
- bietet Einstellungsmöglichkeiten mit Hilfe der Schaltfläche *Einstellungen für das Sicherheitscenter* an.



 Sorgen Sie für die Sicherheit Ihrer Dokumente und des Computers.

Datenschutz

Microsoft nimmt den Datenschutz ernst. Weitere Informationen, wie Microsoft Access für den Datenschutz sorgt, finden Sie in den Datenschutzbestimmungen.

[Datenschutzbestimmungen für Microsoft Access anzeigen](#)
[Datenschutzbestimmungen für Office.com](#)
[Programm zur Verbesserung der Benutzerfreundlichkeit](#)

Sicherheit & mehr

In Office.com finden Sie weitere Informationen zum Datenschutz und zur Sicherheit.

[Microsoft Trustworthy Computing](#)

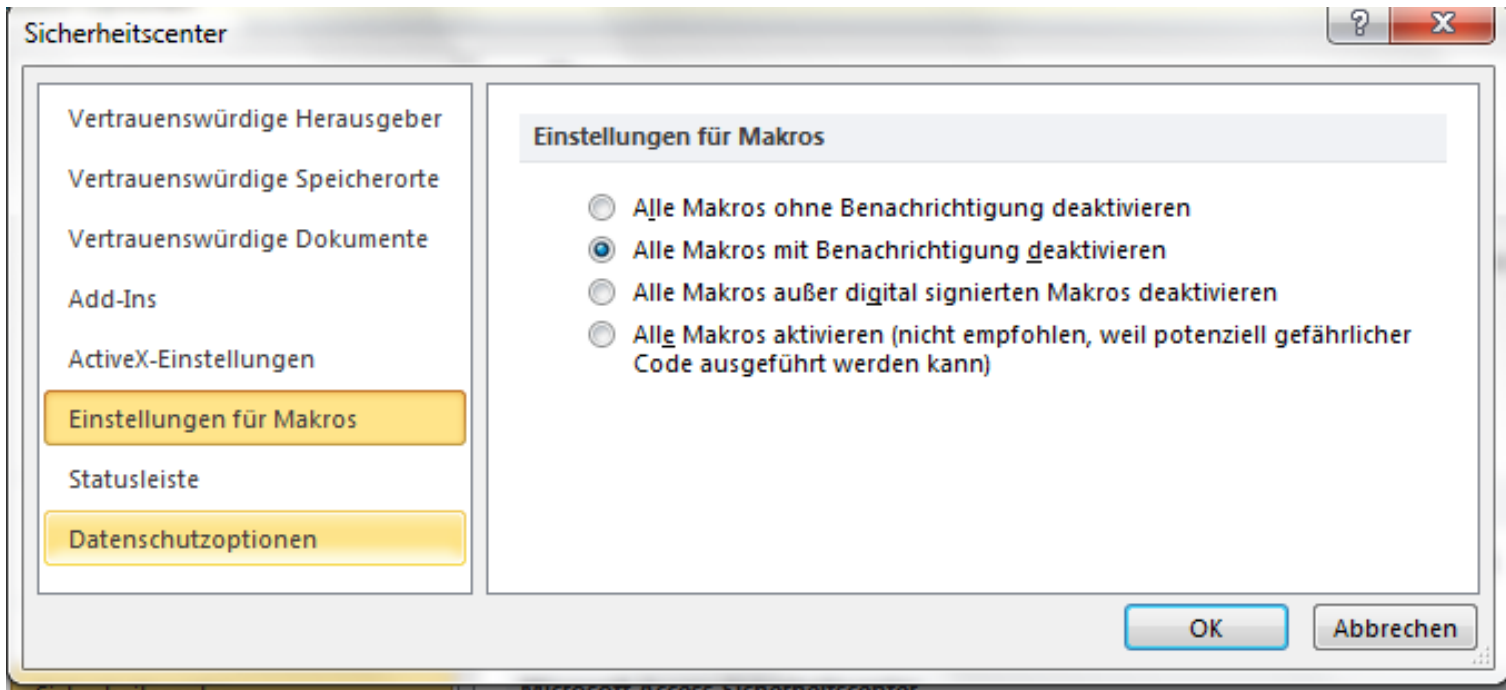
Microsoft Access-Sicherheitscenter

Das Sicherheitscenter enthält Einstellungen für Sicherheit und Datenschutz, um für die Sicherheit des Computers zu sorgen. Es wird davon abgeraten, diese Einstellungen zu ändern.

[Einstellungen für das Sicherheitscenter...](#)

Einstellungen im Sicherheitscenter

- Klicken Sie auf Schaltfläche *Einstellungen für das Sicherheitscenter*.



Einstellungsmöglichkeiten

- Einen zertifizierten Herausgeber benennen.
- Einen sicheren Speicherort im Netz oder auf dem eigenen Rechner für die Datenbank wählen.
- Sind Datenbanken im Netz vertrauenswürdig?
- Welche Makros sind standardmäßig aktiviert?
- Einstellungen im Zusammenhang mit der Internet-Anbindung an Microsoft.