



BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Wirtschaftsinformatik 1

LE 08 – Oberflächen und Ereignisse

Prof. Dr. Thomas Off

<http://www.ThomasOff.de/lehre/beuth/wi1>

Einordnung



06 – Zusammenfassung

05 – Debugger und Testen

04 – Fortgeschrittene Konzepte

04.A
Oberflächen (Teil 1)
Elemente und
Eigenschaften

04.B
Oberflächen (Teil 2)
Ereignisverarbeitung

04.C
Zugriff auf
Dateisystem und
Anwendungen

03 – Grundkonzepte

03.A
Wert
Ausdruck
Variable
Konstante
Datentyp

03.B
Bedingte
Ausführung/
Verzwei-
gungen

03.C
Schleifen

03.D
Felder
Mengen

03.E
Prozedur
Funktion
Modul

02 – Grundlagen der Programmierung

01 – Grundbegriffe der Wirtschaftsinformatik





Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick



Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick

Rückblick



Rückblick



Prozedur

- Form eines Unterprogramms, das keinen Ergebniswert zurückliefert
- Aufruf einer Prozedur (einfache Form)

```
Call <BezeichnerDerProzdeur>
```

- Deklaration einer Prozedur (einfache Form)

```
Sub <BezeichnerDerProzdeur> ()  
<Anweisung(en)>  
End Sub
```

Konvention für Bezeichner von Prozeduren

- Bezeichner von Prozeduren zusammengesetzt aus Verb + ggf. Objekt
- Beispiele

Rückblick



Prozedur mit Parametern

- Aufruf einer Prozedur mit Parametern

```
Call <BezProzdeur> (<BezParam1>, <BezParam2>, ...)
```

- Deklaration einer Prozedur mit Parametern

```
Sub <BezProzdeur> (<BezParam1> As <DTyp>, ...)  
<Anweisung(en)>  
End Sub
```

Konvention

- Parameterbezeichner mit
"p" + Präfix des Datentyps + Name
 - Vorname → **pstrVorname**
 - Geburtsdatum → **pdatGebDatum**



Rückblick



Funktion mit Parametern und Rückgabewert

- ist eine Form des Unterprogramms und liefert einen Ergebniswert zurück
- Aufruf einer Funktion mit Parametern und Rückgabewert sollte innerhalb einer Zuweisung erfolgen

```
Let <Var> = <BezFnkt>(<BezParam1>, <BezParam2>, ...)
```

- Deklaration einer Funktion mit Parametern und Rückgabewert

```
Function <BezFnkt>(<BezParam1> As <DTyp>, ...) As <DTyp>  
  <Anweisung(en)>  
  Let <BezFnkt> = <RückgabeWertOderAusdruck>  
End Function
```

Rückblick



Modul

- dient der Gliederung großer Programme in einzelne Teile
 - fachliche Komponenten (z.B. Bestellungen, Kunden, Produkte)
 - in Schichten (z.B. für Präsentation, Verarbeitung und Speicherung)
- kann anderen Modulen Prozeduren, Funktionen und Variablen zur Verfügung stellen
- Namenskonvention
 - "mdl" + Bezeichnung im Plural (ggf. mit Postfix zur Zuordnung zu einer Schicht)



Rückblick



Syntax für den Zugriff auf Modulbestandteile

- des eigenen Moduls direkt durch Verwendung des Bezeichners
- anderer Module durch Verwendung der Punkt Notation

' Generelle Syntax

```
<BezeichnerAnderesModul>.<BezeichnerDesModulbestandteils>
```

' Beispiele

' Zugriff auf Variable/Feld in anderem Modul

```
Debug.Print mdlKunden.intLetzteKundeNr  
Let knDKunde42 = mdlKunde.kndKundenliste(42)
```

' Funktions- und Prozeduraufruf in anderem Modul

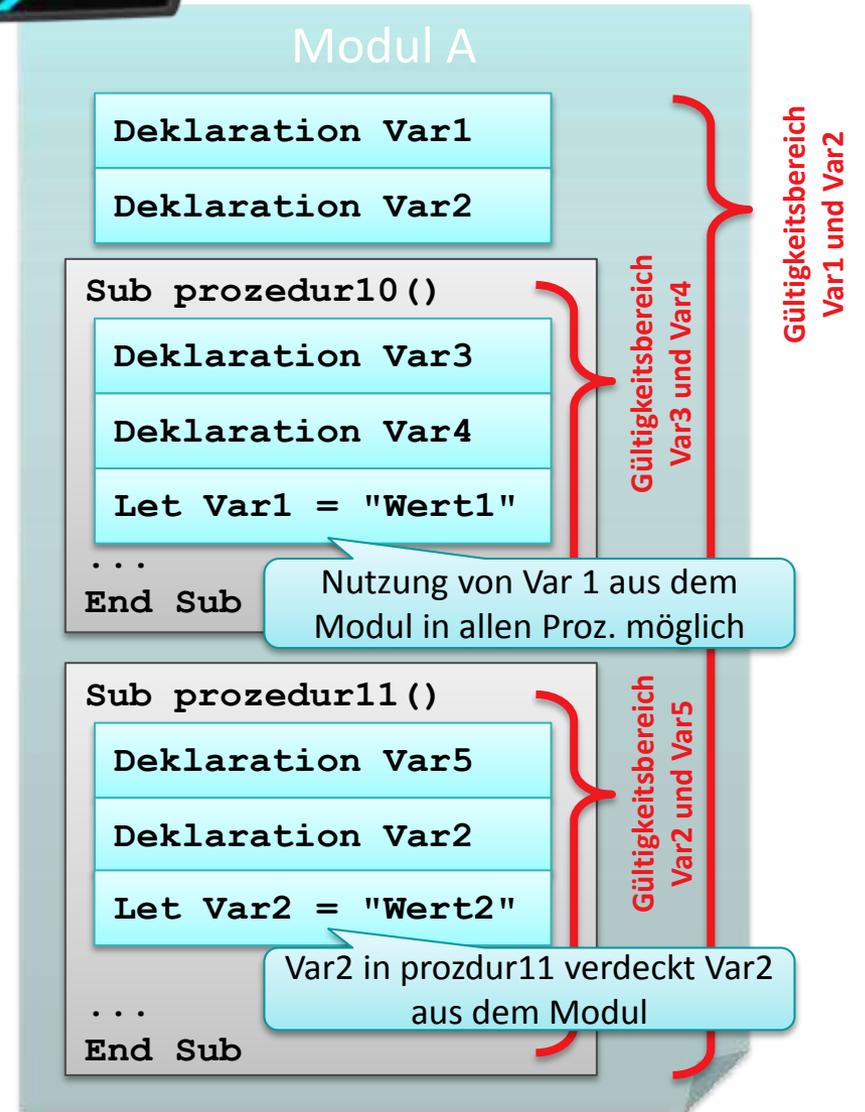
```
Let knDKunde42 = mdlKunden.gibKunde(42)  
Call mdlProdukte.zeigeAlleProdukte
```

Rückblick



Gültigkeitsbereiche

- innerhalb der Bereiche sind Variablen/Konstanten deklariert und verwendbar
- Variablen/Konstanten übergeordneter Gültigkeitsbereiche in untergeordneten Gültigkeitsbereichen verwendbar
- Sonderfall des "Verdeckens" durch eine Variable mit gleichem Bezeichner im einem untergeordnetem Gültigkeitsbereich



Rückblick



Sichtbarkeit

- Elemente eines Moduls ein in anderen Modulen sichtbar, wenn das Element als **Public** deklariert wurde
- Elemente sind nur innerhalb ihres Moduls sichtbar, wenn das Element als **Private** deklariert wurde

Geheimnisprinzip

- dient dem Verbergen der internen Realisierung von Funktionen/Prozeduren und Modulen
- durch Einschränkungen der Sichtbarkeit und eine definierte Schnittstelle nach außen



Rückblick



Syntax: Schlüsselwort **Private** oder **Public** in Verbindung mit

- Deklaration von Variablen auf Modulebene (anstelle von Dim)

```
Private | Public <Variable> As <Datentyp>
```

- Deklaration von Konstanten auf Modulebene

```
Private | Public Const <Konstante> As <DTyp> = <WertAusd>
```

- Zusammengesetzten Datentypen

```
Private | Public Type <Typbezeichner>  
  <Eigenschaft> As <Datentyp>  
End Type
```

- Prozeduren und Funktionen

```
Private | Public Sub <BezProzedur> (<Param> As <DTyp>)  
Private | Public Function <BezFnkt> (<Param> As <DTyp>) As <DTyp>
```

Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick





Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick



Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick

Genereller Aufbau einer Anwendung



Benutzer arbeitet über *Heute* Benutzeroberfläche mit Anwendung

- optimiert für Endgeräte, z.B. Desktop, Web-Anwendung, Mobilgeräte

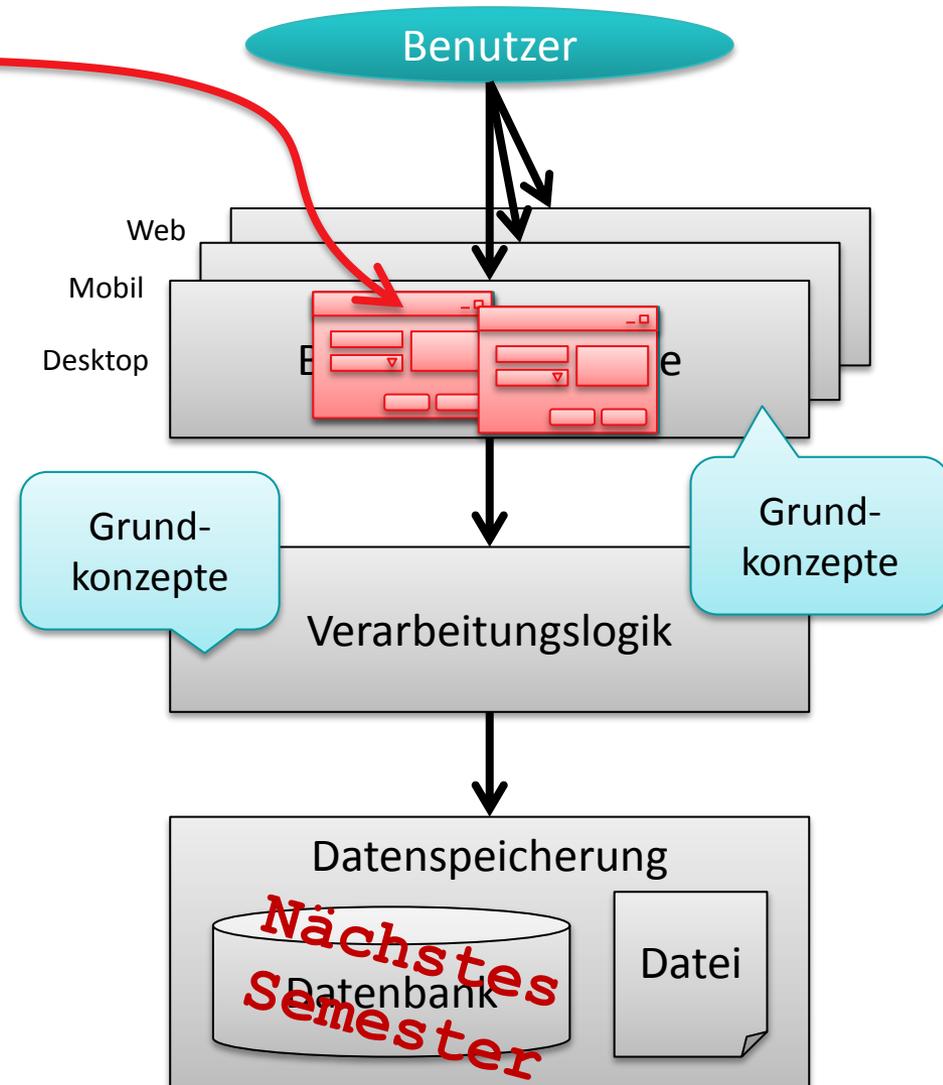
Benutzeroberfläche

- bietet Funktionen, zeigt Ausgabe und nimmt Eingabe entgegen
- nutzt Verarbeitungslogik außerhalb der Oberfläche

Verarbeitungslogik

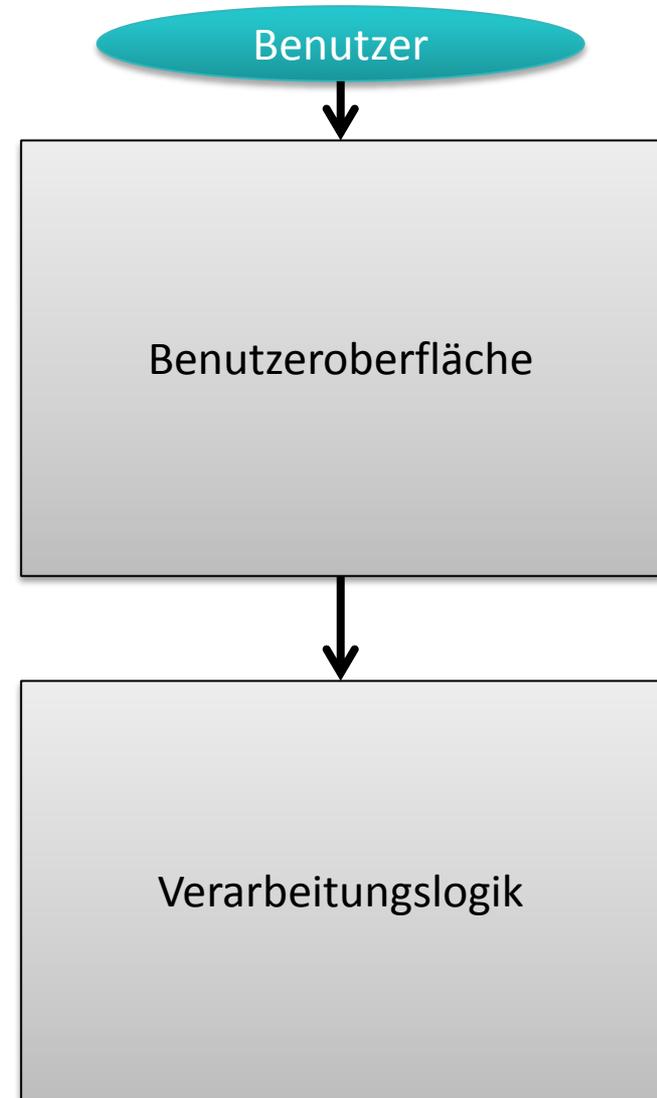
- fachliche Algorithmen zur Verarbeitung der eingegebenen und Aufbereitung der auszugebenden Daten
- nutzt von der Datenspeicherung bereitgestellte Daten

Datenspeicherung bietet Zugriff auf die gespeicherten Daten (i.d.R. in einer Datenbank oder im Dateisystem gespeichert)



Genereller Aufbau einer Anwendung

Benutzeroberfläche



Verarbeitungslogik

- Module
 - Prozeduren
 - Funktionen

Genereller Aufbau einer Anwendung

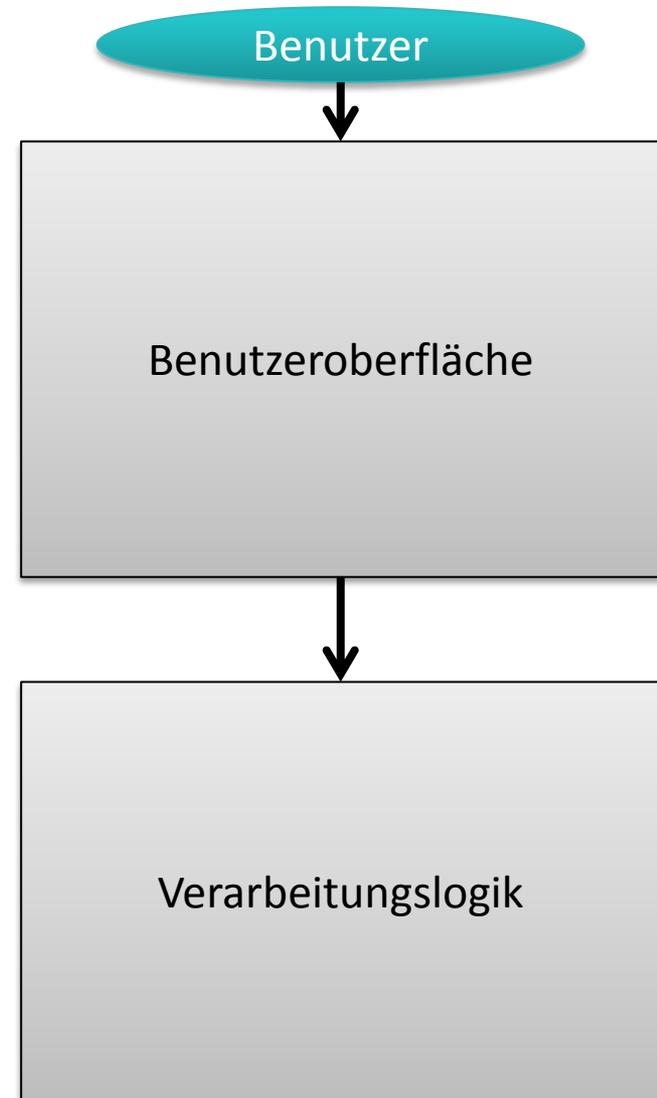


Benutzeroberfläche

- Oberflächenelemente

Verarbeitungslogik

- Module
 - Prozeduren
 - Funktionen



Genereller Aufbau einer Anwendung

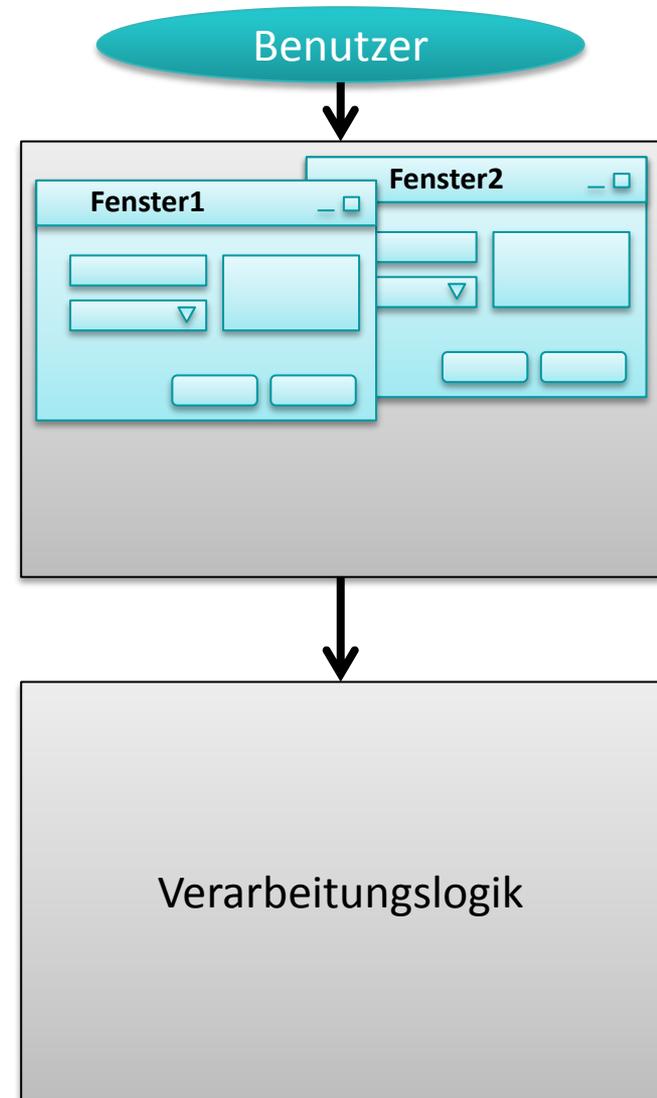


Benutzeroberfläche

- Oberflächenelemente
 - Felder
 - Schaltflächen
 - Gliederungselemente
 - Fenster/Dialoge
 - ...

Verarbeitungslogik

- Module
 - Prozeduren
 - Funktionen



Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick





Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick



Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

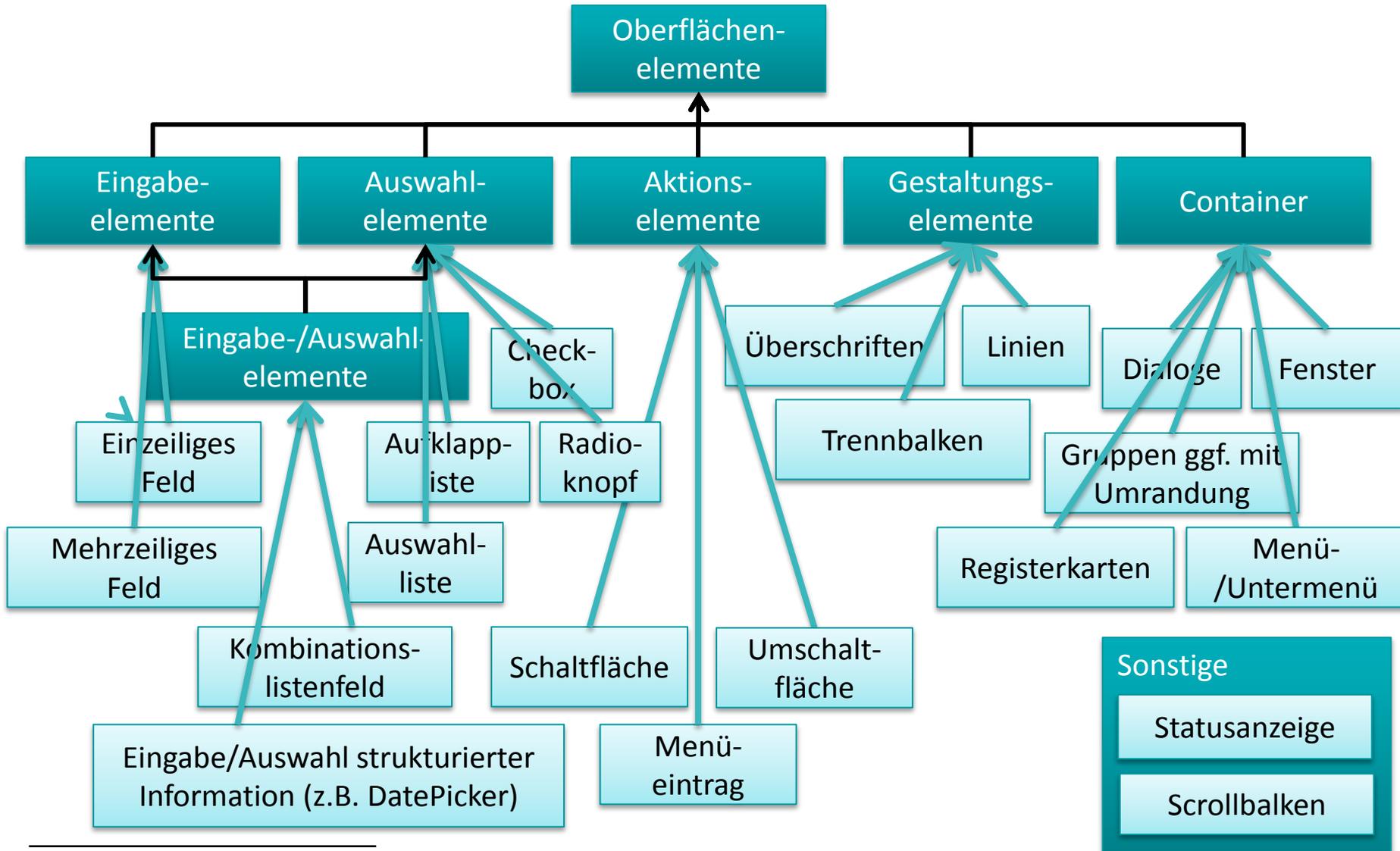
- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick

Wichtige Oberflächenelemente



Eigene Erweiterung der Darstellung aus [Balzert, 1996], S. 552ff.

Wichtige Oberflächenelemente in MS Access



Eingabe und Auswahl

Aktionselemente

Container

Wichtige Oberflächenelemente in MS Access



Eingabe und Auswahl

- Felder (ggf. mit Eingabeunterstützung)
- Aufklappliste/Kombinationsfeld
- Einfach-/Mehrfachauswahllisten
- Radioknöpfe (Optionsfeld)
 - zusammengefasst in Gruppen (mit Rahmen)
- Kontrollkästchen (Checkbox)
 - häufig zusammengefasst in Gruppen (mit Rechteck anstelle von Rahmen)

Aktionselemente

Container

Feld

Mehrzeiliges Feld

Datum

Calendar control showing the month of May 2013. The date 14 is highlighted in yellow. A mouse cursor is pointing at the 14th. The calendar includes a 'Heute' button at the bottom.

Kombinationslistenfeld

Dropdown list with four alternatives: Alternative 1, Alternative 2, Alternative 3, and Alternative 4. A mouse cursor is pointing at Alternative 2.

Mehrfachauswahlliste

Möglichkeit 1	12,50 €
Möglichkeit 2	24,80 €
Möglichkeit 3	37,50 €

A multi-select list with three options: Möglichkeit 1 (12,50 €), Möglichkeit 2 (24,80 €), and Möglichkeit 3 (37,50 €). A mouse cursor is pointing at Möglichkeit 3.

Radioknöpfe

- Alternative 1
- Alternative 2
- Alternative 3

Möglichkeiten

- Möglichkeit 1
- Möglichkeit 2
- Möglichkeit 3

Wichtige Oberflächenelemente in MS Access



Eingabe und Auswahl

Aktionselemente

Container

Wichtige Oberflächenelemente in MS Access



Eingabe und Auswahl

Aktionselemente

- Schaltfläche (einfach)
- Umschaltfläche (Toggle)

Container



Wichtige Oberflächenelemente in MS Access



Eingabe und Auswahl

Aktionselemente

Container

Wichtige Oberflächenelemente in MS Access



Eingabe und Auswahl

Aktionselemente

Container

- Rahmen/Gruppen
- Registerkartensatz mit Registerkarten
- Fenster/Dialoge (in Access als Formulare)
 - Inhalt
 - Datensatzmarkierer
 - Navigationsleiste

Anschrift

Straße Nr.

PLZ Ort

Namen Adressen

Vorname

Name

Benutzername

Namen Adressen

Straße Nr.

PLZ Ort

Formular1

Datensatz: 1 von 1

Kein Filter

Suchen

Wichtige Oberflächenelemente in MS Access



Eingabe und Auswahl

Aktionselemente

Container

Wichtige Oberflächenelemente: Beispiel 08.01



Ziel

- Verwenden der Textfelder als Oberflächenelemente
- Kennenlernen der Layoutansicht

Aufgabe

- Erstellen einer Oberfläche für die Anzeige eines Kunden mit Textfeldern für
 - Name
 - Vorname
 - Straße
 - Hausnummer
 - PLZ
 - Ort(jeweils in getrennten Feldern)

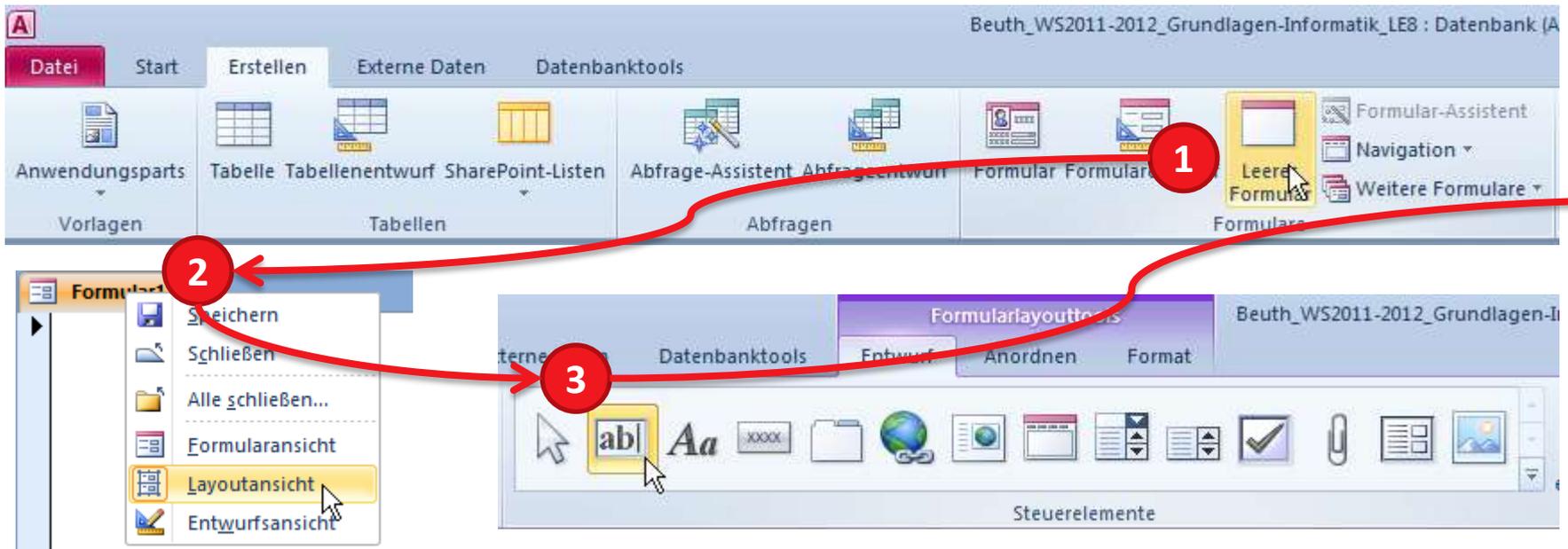


Wichtige Oberflächenelemente: Beispiel 08.01



Schritt 1

- Erstellen-Menü > Leeres Formular > Rechtsklick auf Kopf > Layoutansicht
- Formularlayouttools > Entwurf > Gruppe "Steuerelemente" > "Textfeld" auswählen

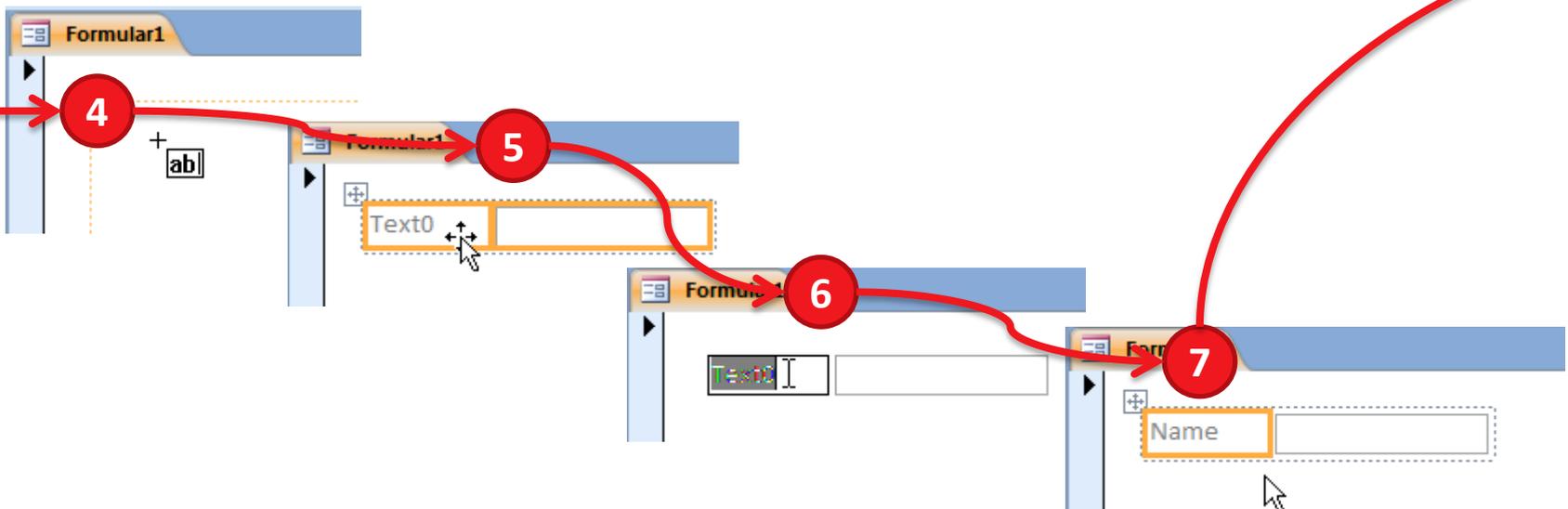


Wichtige Oberflächenelemente: Beispiel 08.01



Schritt 2

- Textfeld innerhalb des Formular positionieren, Doppelklick auf Bezeichnungsfeld (links vom Feld), Änderung der Bezeichnung und Bestätigung mit Enter
- Auswahl des nächsten Textfelds (unter Entwurf>Steuerelemente)

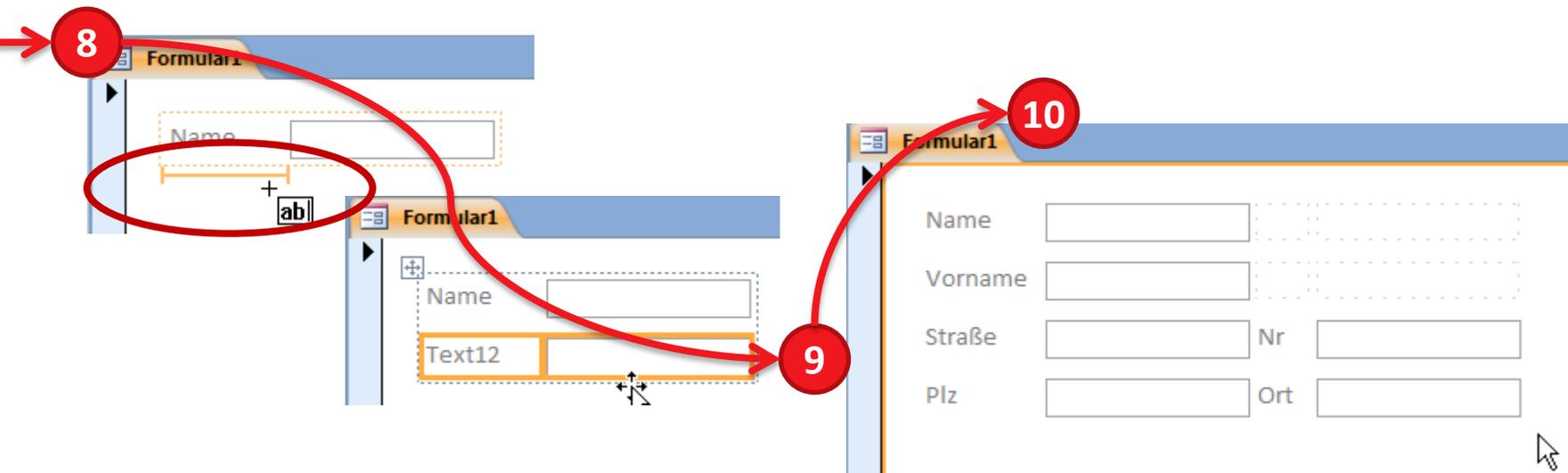


Wichtige Oberflächenelemente: Beispiel 08.01



Weitere Schritte

- weiteres Textfeld innerhalb des Formular unterhalb der bisherigen Felder positionieren, Doppelklick auf Bezeichnungsfeld (links vom Feld), Änderung der Bezeichnung und Bestätigung mit Enter



Wichtige Oberflächenelemente: Beispiel 08.01



Ziel

- Verwenden einer Textfelder als Oberflächenelemente
- Kennenlernen der Layoutansicht

Aufgabe

- Erstellen einer Oberfläche für die Anzeige eines Kunden mit Textfeldern für Name, Vorname, Straße, Hausnummer, PLZ und Ort (jeweils in getrennten Feldern)

Ziel

Kunde			
Name	<input type="text" value="Müller"/>	Vorname	<input type="text" value="Mike"/>
Straße	<input type="text" value="Rosenweg"/>	HausNr	<input type="text" value="8"/>
Plz	<input type="text" value="12345"/>	Ort	<input type="text" value="Berlin"/>



Inhalt

Einordnung

Rückblick

Ausgangspunkt



Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick



Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick



Eigenschaften von Oberflächenelementen

Oberflächenelemente haben Eigenschaften,

- die ihr Aussehen festlegen
- die Interaktionsmöglichkeiten des Benutzers beeinflussen
- Auswirkung auf die Programmierung haben (siehe Teil 2)
- ...

Beispielsweise haben die meisten Oberflächenelemente

- Beschriftung
- Farben
- Änderbarkeit
- Sichtbarkeit
- Name (im Formular nicht sichtbar)

The screenshot shows a user account form titled 'Benutzerkonto'. The form contains several input fields and a checkbox. Red lines with circular endpoints point from the list on the left to specific elements in the form:

- A red line points from 'Beschriftung' to the 'Anrede' label.
- A red line points from 'Farben' to the 'Herr' dropdown menu.
- A red line points from 'Änderbarkeit' to the 'Stammkunde' checkbox.
- A red line points from 'Sichtbarkeit' to the 'Name' label.
- A red line points from 'Name (im Formular nicht sichtbar)' to the 'Verstecktes Feld' label.

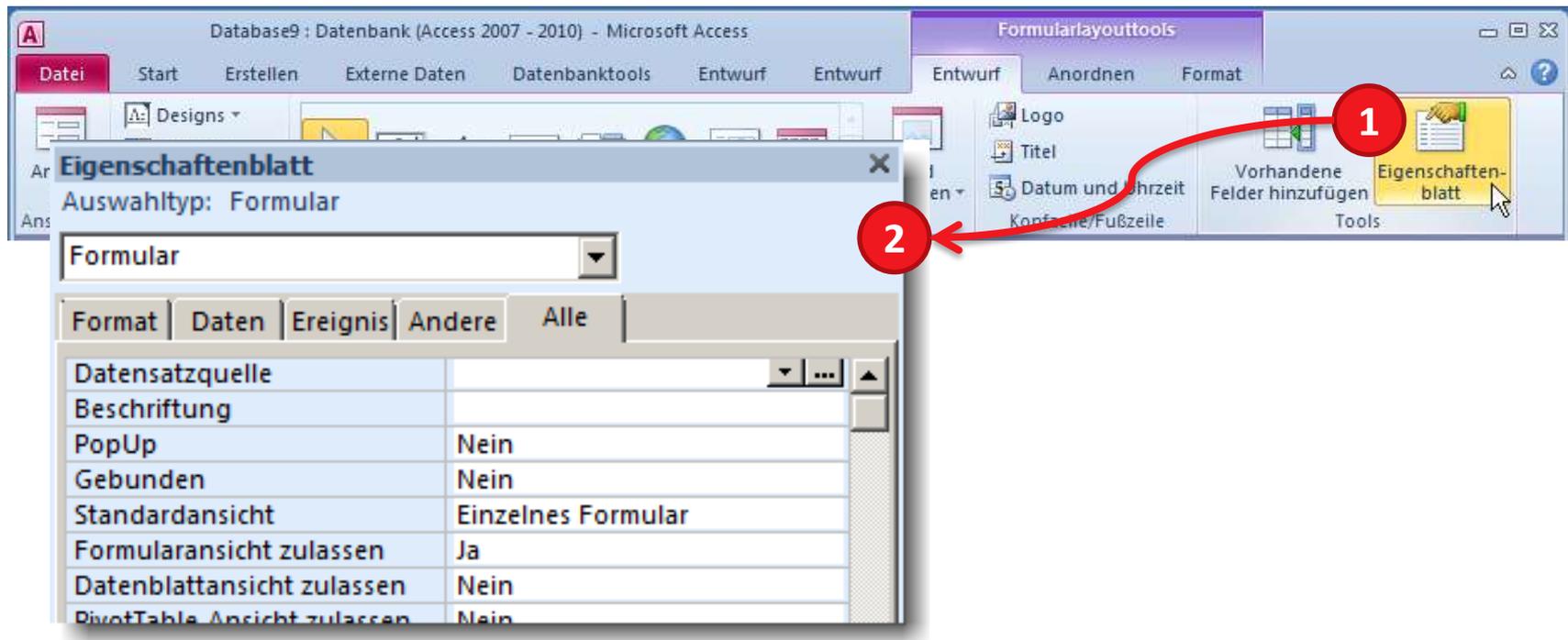
Anrede	Herr	Stammkunde	<input checked="" type="checkbox"/>
Name	Mustermann	Vorname	Mike
Telefon	0123/456 789	Mailadresse	mike@live.de
Benutzername	mike.mustermann	Passwort	*****
Verstecktes Feld			

Eigenschaften von Oberflächenelementen



Eigenschaftenblatt dient zur Festlegung der Eigenschaften von Elementen

- in Layout- oder Entwurfsansicht eines Formulars
- einblenden über die Registerkarte "Entwurf" der Formularlayouttools im Menüband



Erster Schritt: Festlegung des Namens!



Nach Anlegen des Elementes stets den Namen vergeben

- Beispiel: Textfeld "Beispielfeld1" hat
 - Bezeichnung (links) mit Namen "lblBeispielfeld1"
 - Feld (rechts) mit Namen "txtBeispielfeld1"

The image displays two screenshots of the Microsoft Access interface, illustrating the naming convention for a text field. The top screenshot shows the 'Detailbereich' (Detail Area) with a text field named 'Beispielfeld1' selected. The 'Eigenschaftenblatt' (Property Sheet) for this field shows the 'Name' property set to 'lblBeispielfeld1'. The bottom screenshot shows the 'Detailbereich' with the 'Ungebunden' (Unbound) property selected for the same text field. The 'Eigenschaftenblatt' shows the 'Name' property set to 'txtBeispielfeld1'. Red circles and arrows highlight the connections between the field names and their respective property values.





Erster Schritt: Festlegung des Namens!

Denn: Zugriff auf Oberflächenelemente über Namen

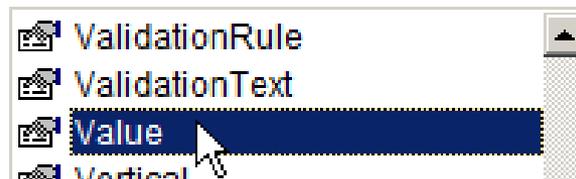
- Nutzung von Me (Referenz auf aktuelles Fenster)

```
Me.tx
```



- gefolgt vom Namen des Feldes "txtBezeichnungsfeld1"

```
Me.txtBeispielfeld1.V
```



- gefolgt von der Eigenschaft, auf die Zugriffen werden soll (hier Value)

```
Me.txtBeispielfeld1.Value = "Hallo Welt!"
```

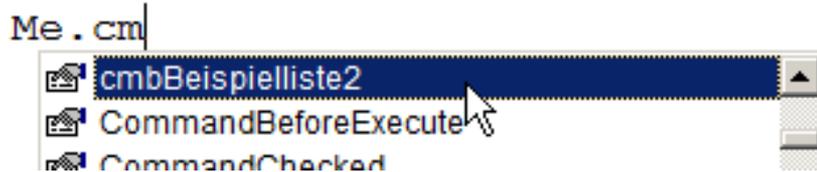
- jeweils getrennt durch Punkt "." (Punkt-Notation)



Erster Schritt: Festlegung des Namens!

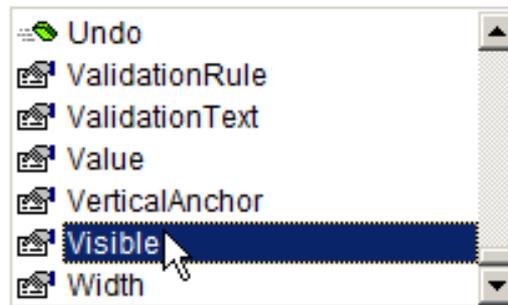
Denn: Zugriff auf Oberflächenelemente über Namen

- Nutzung von Me (Referenz auf aktuelles Fenster)



- gefolgt vom Namen des Feldes "cmbBeispielliste2" und

```
Me.cmbBeispielliste2.V
```



- der Eigenschaft, auf die Zugegriffen werden soll (hier Visible)

```
Me.cmbBeispielliste2.Visible = False
```

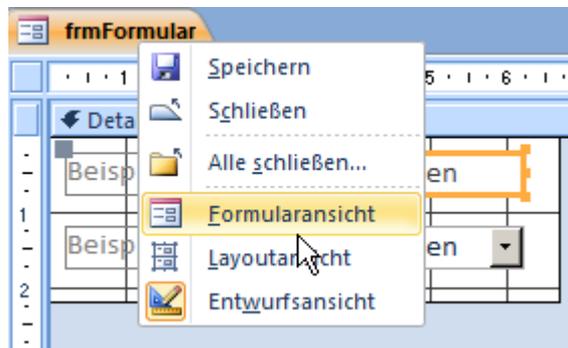
- jeweils getrennt durch Punkt "." (Punkt-Notation)

Erster Schritt: Festlegung des Namens!



Ergebnis: Wird das Formular von der Entwurfsansicht in die Formularansicht geschaltet,

- steht im Textfeld "Hallo Welt"
- ist die Aufklappliste unsichtbar



Hinweis: Details, wie auf das Umschalten reagiert werden kann, folgen auf den nächsten Folien (in Teil 2).



Namenskonvention

Namen von Oberflächenelementen

- werden in VBA (meist) direkt im Programmcode verwendet
- sind anschließend nur noch umständlich änderbar

Konventionen besonders sinnvoll

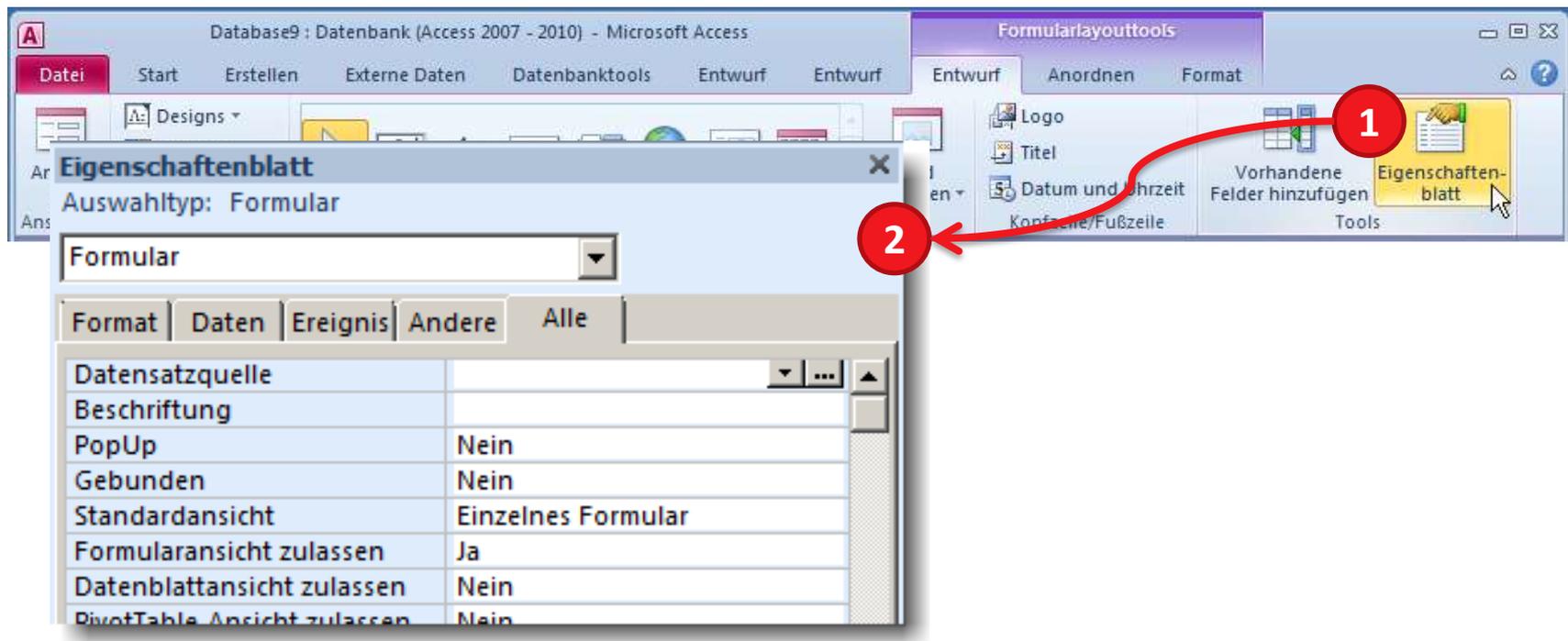
Element	Namenskonvention	Beispiel
Textfeld	txt	txtName
Aufklappliste (Combo)	cmb	cmbAnrede
Liste (Mehrfachauswahl)	lst	lstProduktart
Radioknöpfe (Option)	opt	optAdresstyp
Ankreuzfelder (Checkbox)	chk	chkIstGültig
Gruppen	grp	grpAdresse
Registerkarte (Tab)	tab	tabName
Formular	frm	frmPersonen
Schaltfläche (Button)	btn	btnZurueck
Bezeichnungsfeld (Label)	lbl	lblVorname
...		

Eigenschaften von Oberflächenelementen



Eigenschaftenblatt dient zur Festlegung der Eigenschaften von Elementen

- in Layout- oder Entwurfsansicht eines Formulars
- einblenden über die Registerkarte "Entwurf" der Formularlayouttools im Menüband



Wichtige Oberflächenelemente: Beispiel 08.02



Ziel

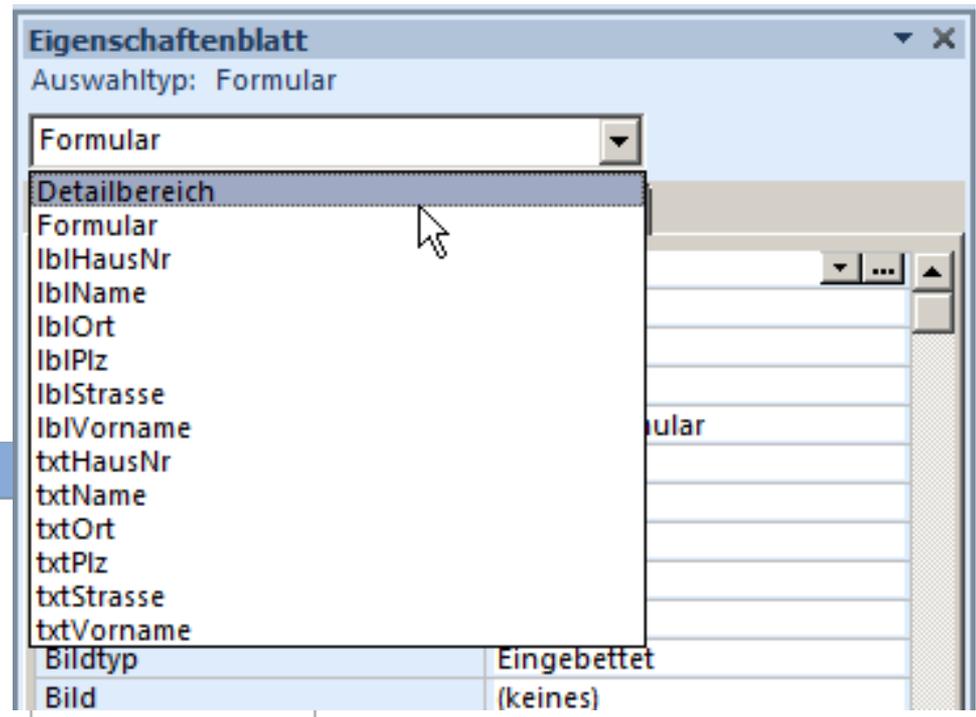
- Festlegen der Namen von Oberflächenelementen

Aufgabe

- Vergeben Sie Namen für die vorhandenen Oberflächenelemente
 - Textfelder
 - Bezeichnungsfelder
- Orientieren Sie sich an der Namenskonvention

Ziel

Kunde	
Name	<input type="text" value="Müller"/> Vorname
Straße	<input type="text" value="Rosenweg"/> HausNr
Plz	<input type="text" value="12345"/> Ort



Wichtige Eigenschaften von Textfeldern



Name

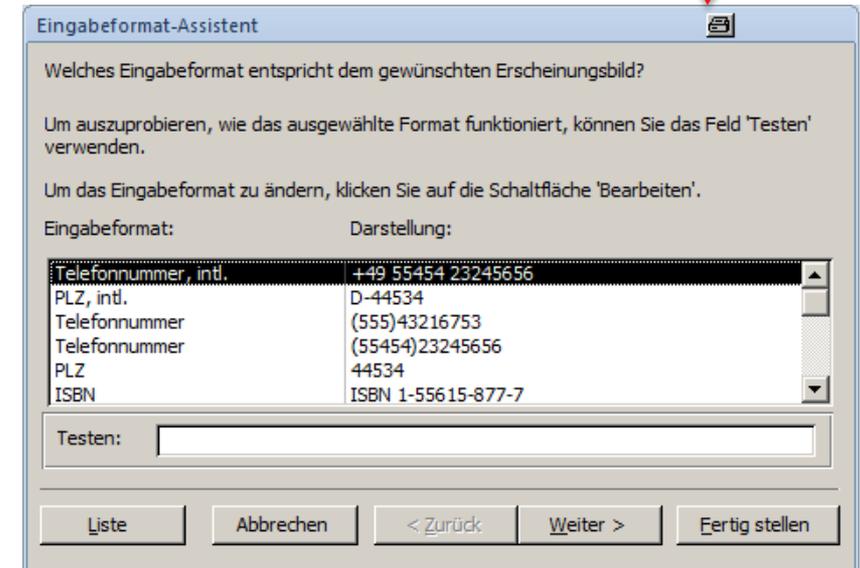
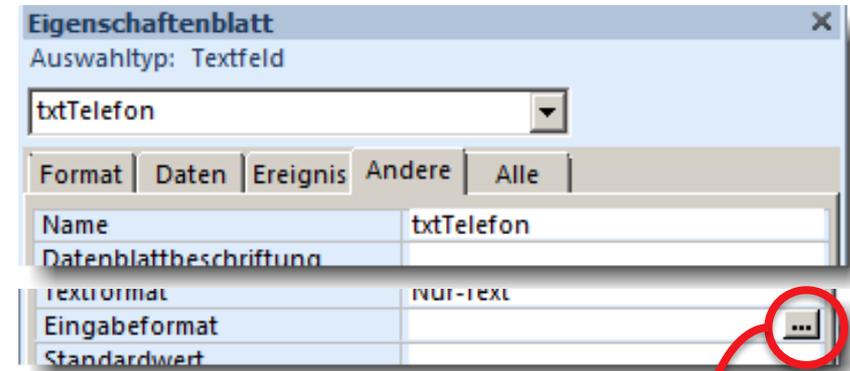
- für Programmierung benötigt, muss eindeutig sein

Eingabeformat

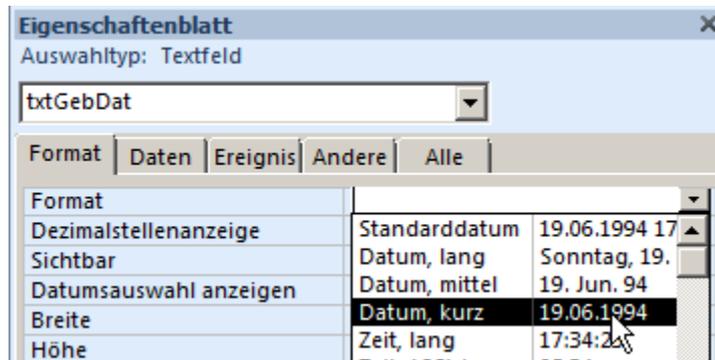
- wie Daten vom Benutzer einzugeben sind

Anzeigeformat

- wie eingegebene Daten angezeigt werden



...



Wichtige Eigenschaften von Textfeldern (Forts.)



...

Standardwert

- Inhalt des Feldes, wenn nichts eingegeben wurde

Eingabeformat	
Standardwert	"01.01.1990"
Gültigkeitsregel	

GebDat

Leeres Feld bei Verlassen gefüllt

GebDat

Änderbarkeit

- Angabe, ob Änderung durch Benutzer zulässig

Filter anwenden	Datenbankstandard
Aktiviert	Nein
Gesperrt	Ja
Smarttags	Nein

Sichtbarkeit

- Feld im Formular sichtbar ist

Benutzername

mike.mustermann

Hinweis: Beschriftung ist eigenständiges Element

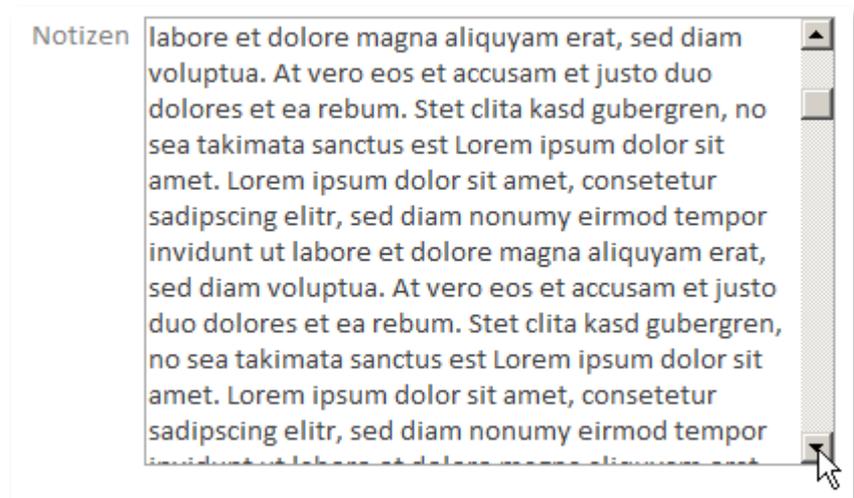
Eigenschaftenblatt	
Auswahltyp: Textfeld	
txtVerstecktesFeld	
Format Daten Ereignis Andere Alle	
Format	
Dezimalstellenanzeige	Automatisch
Sichtbar	Nein
Datumsauswahl anzeigen	Ja
Breite	Nein
Höhe	0,556cm

Wichtige Eigenschaften von Textfeldern (Forts.)



Mehrzeilige Textfelder

- verwendbar wie einzeilige Textfelder
- Bildlaufleiste (Scrollbar) sinnvoll
 - Standard: Keine
 - Sinnvoll: Vertikal





Wichtige Eigenschaften: Beispiel 08.03

Ziel

- Textfelder verwenden und ihre Eigenschaften festlegen

Aufgabe

- Mehrzeiliges Textfeld für Notizen hinzufügen
- Eingabeformat für PLZ festlegen

Ziel

Kunde

Name	<input type="text" value="Müller"/>	Vorname	<input type="text" value="Mike"/>
Straße	<input type="text" value="Rosenweg"/>	HausNr	<input type="text" value="8"/>
Plz	<input type="text" value="12345"/>	Ort	<input type="text" value="Berlin"/>
Notizen	<input type="text" value="Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor"/>		



Wichtige Eigenschaften von Aufklapplisten



Spalten

- Anzahl und Breite

Auswahlmöglichkeiten

- Herkunftstyp: Werteliste
- Datensatzherkunft: durch Semikolon getrennte Werte

Spaltenanzahl	2
Spaltenbreiten	2,54 cm; 2,54 cm
Spaltenüberschriften	Multi XY Maxi AB Mega Drive

Produkt	
Multi XY	19,95
Maxi AB	14,95
Mega Drive	24,95

Eigenschaftenblatt				
Auswahltyp: Kombinationsfeld				
cmbProduktZweiSpalten				
Format	Daten	Ereignis	Andere	Alle
Steuerelementinhalt				
Datensatzherkunft		"Multi XY";19,95;"Maxi AB";14,95;"Mega Drive";24,95		
Herkunftstyp		Werteliste		
Gebundene Spalte		1		
Nur Listeneinträge		Nein		

Datenblattbeschriftung	
Spaltenanzahl	1
Spaltenbreiten	2,54 cm

Produkt	
Multi XY (19,95)	
Maxi AB (14,95)	
Mega Drive (24,95)	

Eigenschaftenblatt				
Auswahltyp: Kombinationsfeld				
cmbProduktEineSpalte				
Format	Daten	Ereignis	Andere	Alle
Steuerelementinhalt				
Datensatzherkunft		"Multi XY (19,95)";"Maxi AB (14,95)";"Mega Drive (24,95)"		
Herkunftstyp		Werteliste		
Gebundene Spalte		1		

Wichtige Eigenschaften von Aufklapplisten (Forts.)



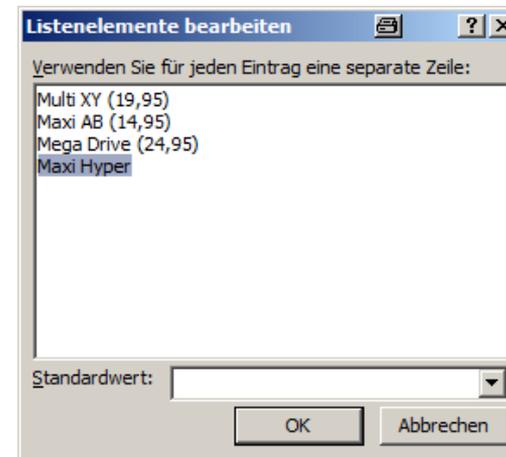
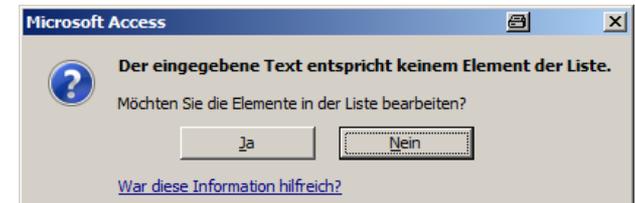
Auswahl oder Eingabe

- Nur Listeneinträge legt fest, ob direkte Eingaben möglich sind
- Wertelistenbearbeitung legt fest, ob Benutzer zusätzliche Elemente hinzufügen kann
 - Standard: Ja
 - Empfehlung: Nein

Produkt

- Multi XY (19,95)
- Maxi AB (14,95)
- Mega Drive (24,95)

Gebundene Spalte	1
Nur Listeneinträge	Ja
Wertlistenbearbeitung zulassen	Ja
Bearbeitungsformular für Listen	



Gebundene Spalte	1
Nur Listeneinträge	Ja
Wertlistenbearbeitung zulassen	Nein
Bearbeitungsformular für Listen	

Wichtige Eigenschaften von Aufklapplisten (Forts.)

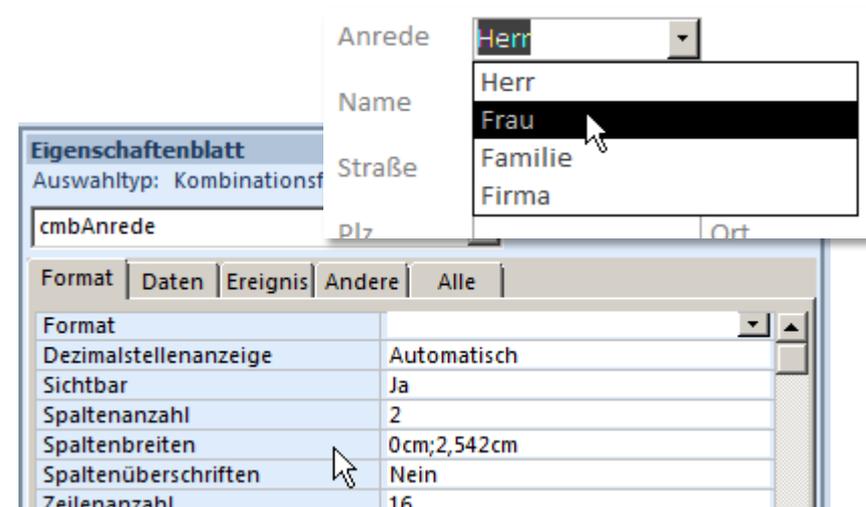
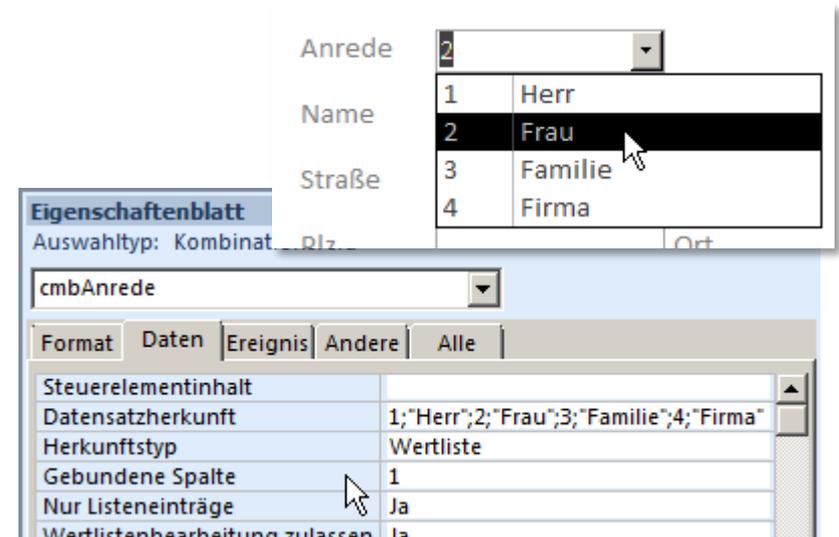


Wert der Aufklappliste

- Empfehlung: mind. 2 Spalten
 - Nr. des Eintrags
 - Text des Eintrags
- Programmierung soll Nr. des Eintrags verwenden
 - gebundene Spalte mit Nr. versehen
- Anzeige der Spalte mit Text
 - Breite der Spalte mit der Nr. auf 0 cm setzen

Weitere

- Name, Sichtbarkeit, Änderbarkeit wie bei Textfeldern



Wichtige Eigenschaften von Auswahllisten

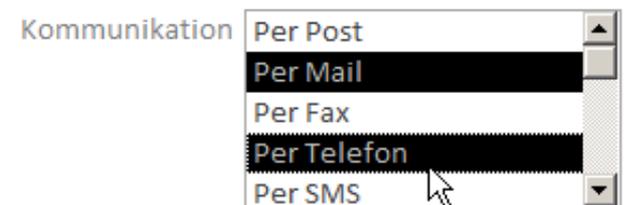
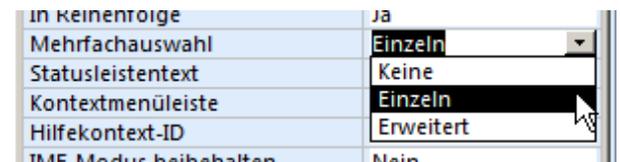
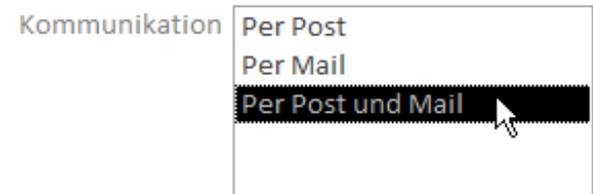
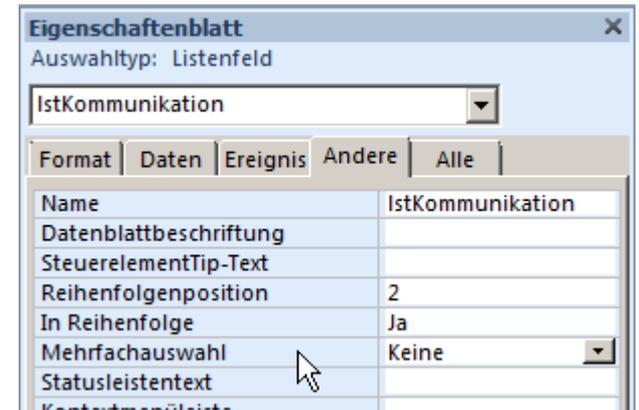


Mehrfachauswahl

- Kann der Benutzer nur eine oder mehrere Auswahlen treffen?
- Kann nur in der Entwurfsansicht geändert werden (vermutl. Fehler von Access)

Weitere

- Auswahlmöglichkeiten, Spalten wie bei Aufklapplisten
- Name, Sichtbarkeit, Änderbarkeit wie bei Textfeldern



Wichtige Eigenschaften: Beispiel 08.04



Ziel

- Listen verwenden und ihre Eigenschaften festlegen

Aufgabe

- Aufklappliste für die Anrede (Herr, Frau, Familie, Firma)
- Mehrfachauswahlliste für Branchen (z.B. Handel, Industrie)

Ziel

Kunde

Anrede

Name

Straße

Plz Ort

Branchen

Notizen

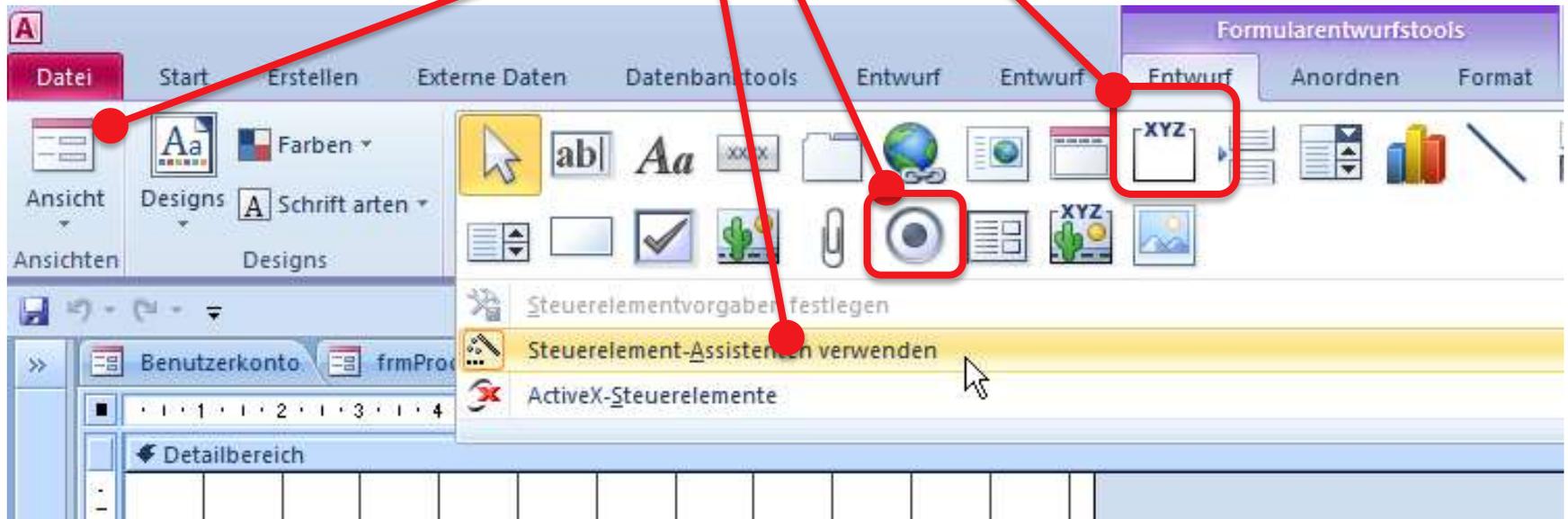


Wichtige Eigenschaften von Radioknöpfen



Hinweise zu Radioknöpfen

- als Gruppe bequem einzufügen
- oder einzeln einfügen
- komfortable Nutzung des Assistenten
- nur in Entwurfsansicht verfügbar!



Wichtige Eigenschaften von Radioknöpfen (Forts.)



Sinnvoll in Gruppe/Rahmen mit Assistent

Optionsgruppen-Assistent

Eine Optionsgruppe enthält eine Gruppe von Optionsfeldern, Kontrollkästchen oder Umschaltflächen, von denen nur eines bzw. eine ausgewählt werden kann.

Welche Beschriftung soll jede Option erhalten?

Beschriftungen	Werte
Per Mail	
Per Brief	
Per Fax	
Per SMS	
*	

Abbrechen < Zurück Weiter > Fertig stellen

Optionsgruppen-Assistent

Wenn Sie auf eine Option einer Optionsgruppe klicken, wird der Optionsgruppe der Wert der ausgewählten Option zugewiesen.

Welchen Wert möchten Sie jeder Option zuweisen?

Beschriftungen	Werte
Per Mail	1
Per Brief	2
Per Fax	3
Per SMS	4

Abbrechen < Zurück Weiter > Fertig stellen

Optionsgruppen-Assistent

Möchten Sie eine Option als Standardauswahl in Ihrer Optionsgruppe vorgeben?

Ja, und zwar folgende: Per Mail

Nein, ich möchte keine Standardauswahl treffen

Abbrechen < Zurück Weiter > Fertig stellen

Optionsgruppen-Assistent

Welche Steuerelemente möchten Sie in Ihrer Optionsgruppe verwenden?

Optionsfelder

Kontrollkästchen

Umschaltflächen

Welchen Stil soll Ihre Optionsgruppe haben?

Graviert

Schattiert

Flach

Vertieft

Erhöht

< Zurück Weiter > Fertig stellen

Optionsgruppen-Assistent

Welche Beschriftung soll Ihre Optionsgruppe haben?

Kommunikation

Dies sind alle Informationen, die der Assistent benötigt, um Ihre Optionsgruppe zu erstellen.

Abbrechen < Zurück Weiter > Fertig stellen

Kommunikation

Per Mail

Per Brief

Per Fax

Per SMS

Wichtige Eigenschaften von Radioknöpfen (Forts.)



Standardwert

- wird über die Gruppe/den Rahmen festgelegt

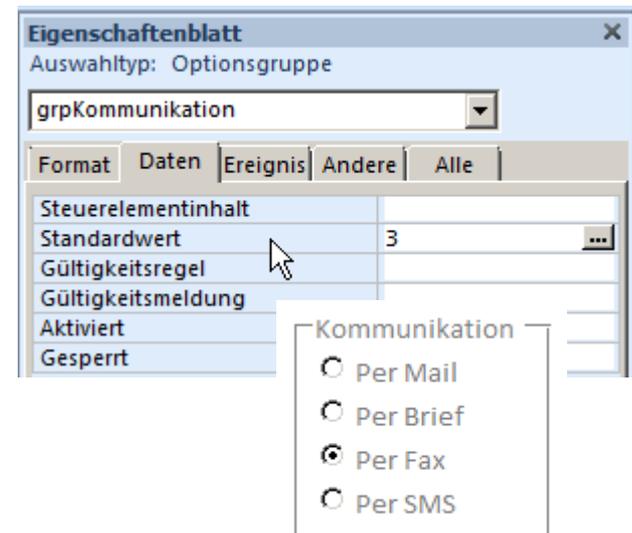
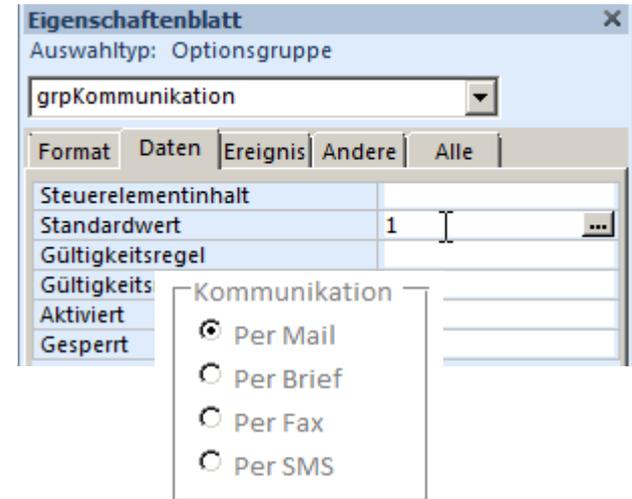
Aktueller Wert

- kann über Gruppe ermittelt werden

Beachten, dass Radioknopf und Bezeichnung getrennte Elemente sind

Weitere Eigenschaften

- Name, Sichtbarkeit, Änderbarkeit wie bei Textfeldern



Wichtige Eigenschaften: Beispiel 08.05



Ziel

- Radioknöpfe verwenden und ihre Eigenschaften festlegen

Aufgabe

- Radioknöpfe für den Status des Kunden im Bonusprogramm
 - Alternative: Platin, Gold, Silber, Basic
 - Basic soll vorgelegt sein



Kunde

Anrede: Herr

Name: Müller Vorname: Mike

Straße: Roseweg HausNr: 8

Plz: 12345 Ort: Berlin

Kundenstatus

Platin

Gold

Silber

Basic

Notizen: Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet elita kasd gubergren, no sea

Wichtige Eigenschaften von Checkboxes

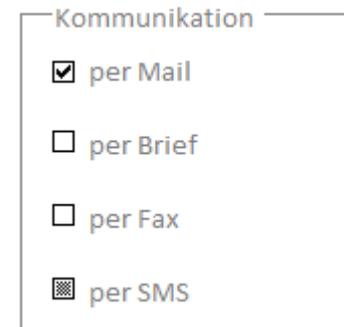


Standardwert

- Festlegung sinnvoll, da sonst "dritter" Zustand als Standard
 - Standardwerte im Beispiel
 - per Mail: True
 - per Brief: False
 - per Fax: False
 - per SMS: ohne Festlegung

Hinweis

- Assistent für Radioknöpfe ermöglicht scheinbar auch Checkboxes aber keine Mehrfachauswahl (Fehler in Access?)



Wichtige Eigenschaften von Checkboxes (Forts.)



Layout von Checkboxes

- Rechteck als Rahmen
- Bezeichnung frei positioniert mit "normalem" Hintergrund

The diagram illustrates the layout and properties of checkboxes in a software application. It shows a toolbar with a 'Rechteck' (Rectangle) button (1) and a 'Bezeichnung' (Label) button (3). A grid layout (2) shows checkboxes for 'per Mail', 'per Brief', 'per Fax', and 'per SMS'. A label 'Kommunikation' (4) is positioned to the left of the checkboxes. The 'Eigenschaftenblatt' (Properties) window (5) shows the 'Bezeichnungsfeld' (Label Field) properties, including 'Beschriftung' (Label) set to 'Kommunikation', 'Sichtbar' (Visible) checked, and 'Hintergrundart' (Background Type) set to 'Normal'. The final layout (6) shows the checkboxes with the label 'Kommunikation' positioned to the left of the checkboxes.

Wichtige Eigenschaften: Beispiel 08.06



Ziel

- Checkboxes verwenden und ihre Eigenschaften festlegen

Aufgabe

- Checkboxes für Produktinformation hinzufügen
 - Möglichkeiten: per Mail, per Briefpost, per Fax, per SMS
 - per Briefpost soll vorgelegt sein



Kunde

Anrede:

Name: Vorname:

Straße: HausNr:

Plz: Ort:

Kundenstatus

Platin
 Gold
 Silber
 Basic

Produktinformation

per Briefpost
 per Mail
 per Fax
 per SMS

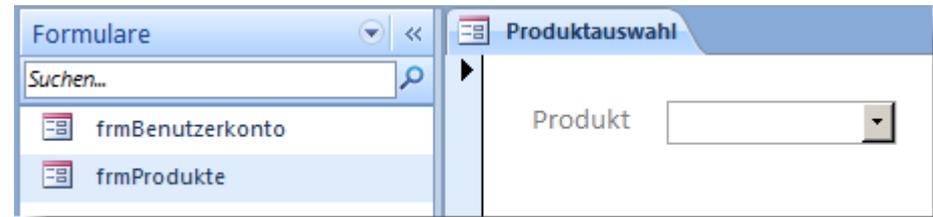
Notizen: Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores

Wichtige Eigenschaften von Formularen



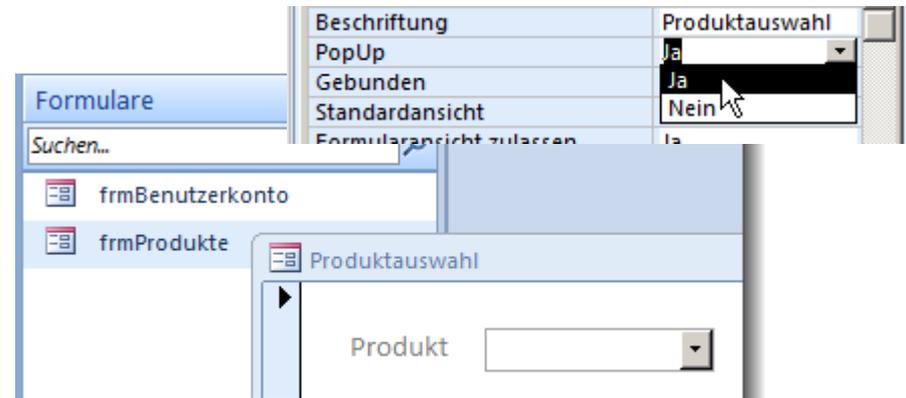
Beschriftung

- Wird im Tab bzw. in der Titelleiste des Formulars angezeigt



Darstellung

- Registerblatt oder Dialog/Fenster über Eigenschaft "Popup"



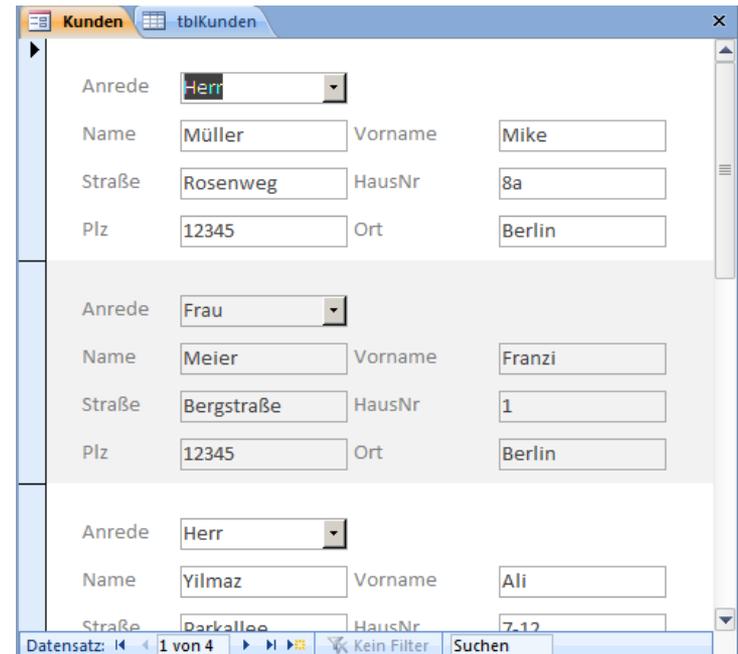
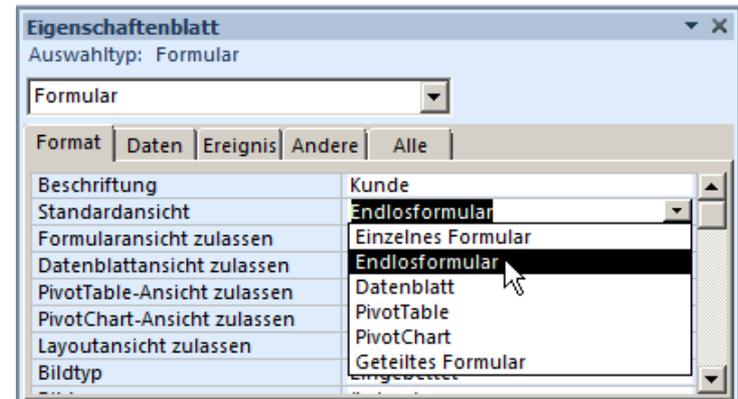
Wichtige Eigenschaften von Formularen (Forts.)



Layout kann angepasst werden, z.B.

- Darstellung (Standardansicht): z.B. Endlosformular (siehe Kurs Datenbanken)
- Datensatzmarkierer und Navigationsschaltflächen nur bei "vielen" Datensätzen aus Datenbank (siehe Kurs Datenbanken) sinnvoll
 - Empfehlung: Abschalten

Rahmenart	Veränderbar
Datensatzmarkierer	Nein
Navigationsschaltflächen	Nein
Navigationsschaltflächen	Ja
Trennlinien	Nein

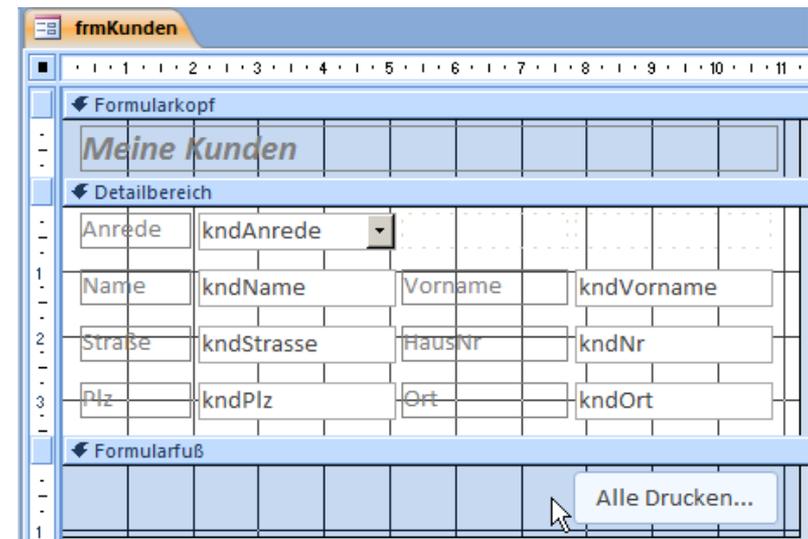
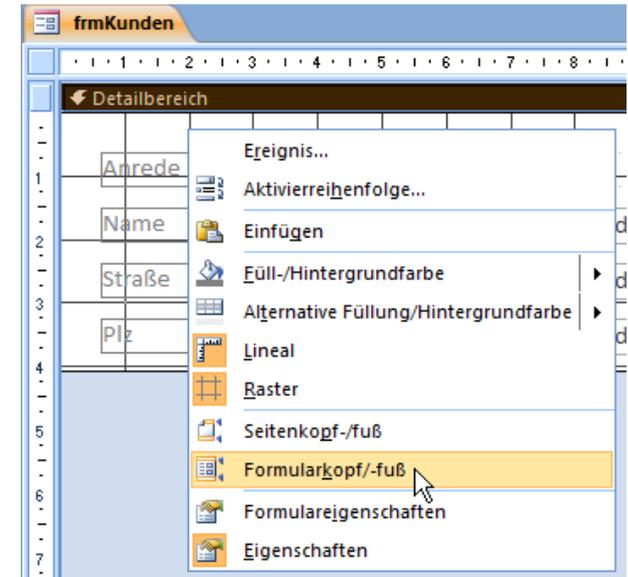


Wichtige Eigenschaften von Formularen (Forts.)



Layout kann angepasst werden, z.B.

- Formulkopf und Fuß sinnvoll für Konsistenz und Darstellung großer Formulare
- Kopf- und Fuß immer sichtbar



Wichtige Eigenschaften von Formularen (Forts.)



Layout kann angepasst werden, z.B.

- Formularkopf und Fuß sinnvoll für Konsistenz und Darstellung großer Formulare
- Kopf- und Fuß immer sichtbar
- z.B. für Aktionen auf Elementen, Navigationsmöglichkeiten

Kunden

Meine Kunden

Anrede Herr

Name Müller Vorname Mike

Straße Rosenweg HausNr 8a

Plz 12345 Ort Berlin

Anrede Frau

Name Meier Vorname Franzi

Straße Bergstraße HausNr 1 Ort Berlin

Alle Drucken...

Datensatz: 1 von 4 Kein Filter Suchen

Kunden

Neuer Kunde

Anrede Herr

Name Müller Vorname Mike

Straße Bergstraße HausNr 8

Plz 12345 Ort Berlin

Notizen sadipscing elit, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren

Speichern

Wichtige Eigenschaften: Beispiel 08.07



Ziel

- Eigenschaften des Formulars festlegen

Aufgabe

- Datensatzmarkierer und Navigationsleiste ausblenden
- Kopf- und Fußzeile hinzufügen, wie im folgenden Beispiel
 - Bezeichnungsfeld "Neuer Kunde" im Kopfbereich
 - Schaltflächen "Abbrechen" und "Speichern" im Fußbereich

The screenshot shows a web form titled "Kunden" with a sub-header "Neuer Kunde". The form is contained within a blue header and footer. The header contains the title "Neuer Kunde". The footer contains two buttons: "Abbrechen" and "Speichern". The form fields are as follows:

Anrede	Herr		
Name	Müller	Vorname	Mike
Straße	Rosenweg	HausNr	8
Plz	12345	Ort	Berlin
Kundenstatus		Produktinformation	
<input type="radio"/> Platin		<input type="checkbox"/> perMail	
<input type="radio"/> Gold		<input checked="" type="checkbox"/> per Briefpost	
<input type="radio"/> Silber		<input type="checkbox"/> per Fax	
<input checked="" type="radio"/> Basic		<input type="checkbox"/> per SMS	
Notizen	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores		



Verwendung der Elemente

Auswahl von Alternativen (Einfachauswahl)

- Aufklapplisten, wenn
 - die Alternativen vom Programm ermittelt werden oder
 - zu erwarten ist, dass sich die Alternativen (später) ändern
 - und es eine größere Anzahl Alternativen gibt
- Radioknöpfe, wenn
 - sich die möglichen Alternativen nicht mehr ändern
 - und wenige Alternativen existieren

Auswahl einer oder mehrerer Möglichkeiten (Mehrfachauswahl)

- Listen, wenn
 - die Möglichkeiten vom Programm ermittelt werden oder
 - zu erwarten ist, dass sich die Möglichkeiten (später) ändern
 - und es eine größere Anzahl Möglichkeiten gibt
- Checkboxen, wenn
 - sich die möglichen Möglichkeiten nicht mehr ändern
 - und wenige Möglichkeiten existieren

Auswahl sehr vieler Elemente über spezielle Such-/Filterdialoge

Inhalt

Einordnung

Rückblick

Ausgangspunkt



Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick



Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick

Grundlegende Gestaltung



Inhalt

- Ausgewählte Gestaltungsprinzipien
- Wichtige Gestaltungsrichtlinien
- GUI Style Guides
 - Bedeutung
 - Inhalt



Wichtige Gestaltungsprinzipien

Aufmerksamkeitssteuerung

Gesetz der guten Gestalt

Figur-Hintergrund-Unterscheidung

Gestaltungs- und Gliederungsprinzipien

- Gleichartigkeit
- Nähe
- Geschlossenheit
- Gute Fortsetzung
- ...

Gruppierungsregeln nach

- Größe
- Anzahl
- Anordnung

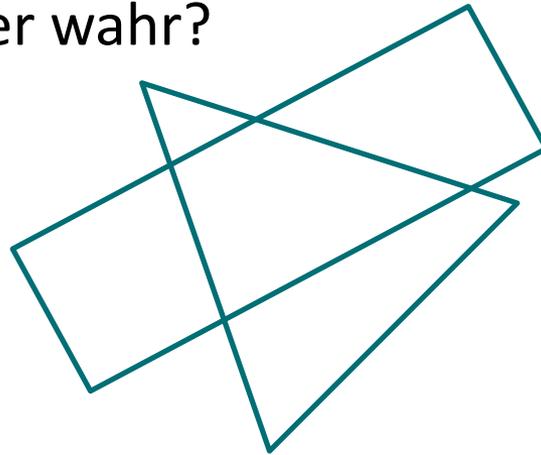
Farben und Hervorhebungen

Gesetz der guten Gestalt

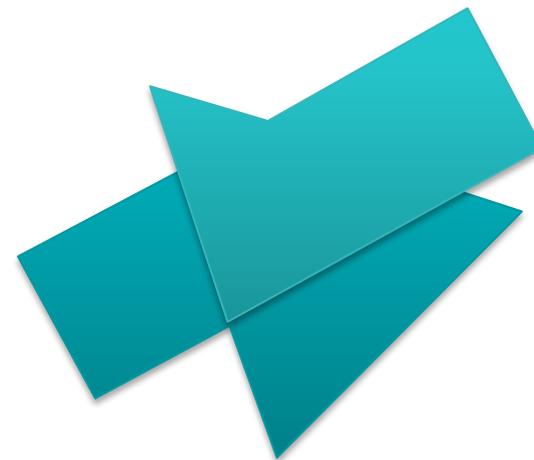
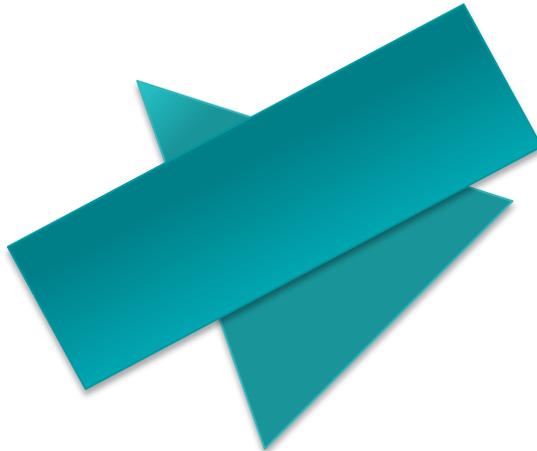


Bildung von Gruppen und Zusammenhängen im Sehfeld bei mehreren Alternativen immer in der einfachsten Form

– Was nehmen Sie hier wahr?



– Variante A oder B?



Figur-Hintergrund-Unterscheidung



Gruppe von Elementen soll als Figur wahrgenommen werden und übrige Elemente sollen Hintergrund bilden

– Problemeispiele



– Lösungsansätze

- kleine Elemente als Figur auf größerem Hintergrund
- dunkle Elemente als Figur auf hellem Hintergrund
- zentral angeordnete Elemente
- ...

Figur-Hintergrund-Unterscheidung



– Lösungsansätze

- ...
- Elementanordnung mit vertikaler oder horizontaler Hauptachse eher als Figur wahrgenommen, als mit schräger Hauptachse
- Symmetrisches Elementanordnung eher als Figur erkennbar, als asymmetrische
- nach außen gewölbte Ränder begrenzen Figur besser, als nach innen gewölbte



Binnengliederung



Gleichartigkeit: gleiche oder ähnliche Elemente werden als Gruppe wahrgenommen

- Ungleichmäßige Verteilung lässt keine Gruppen erkennen



- Gleichartigkeit macht zusammengehörige Gruppen erkennbar





Binnengliederung

Nähe: räumlich nahe Elemente werden als Gruppe wahrgenommen

– gleiche Abstände lassen keine Gruppen erkennen



– räumliche Nähe macht Gruppen deutlich





Binnengliederung

Gute Fortsetzung: Elemente, die räumlich oder zeitlich in einfacher, harmonischer, gesetzmäßiger Weise angeordnet sind, werden als zusammengehörig wahrgenommen

– Eher so...

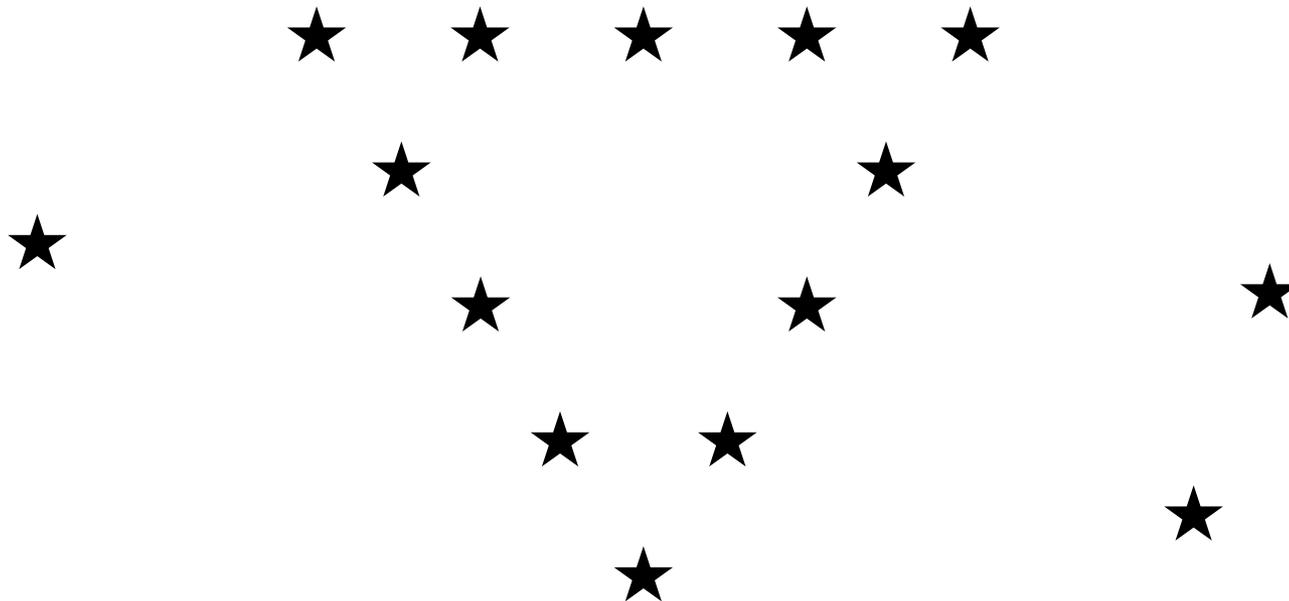
als so...



Binnengliederung



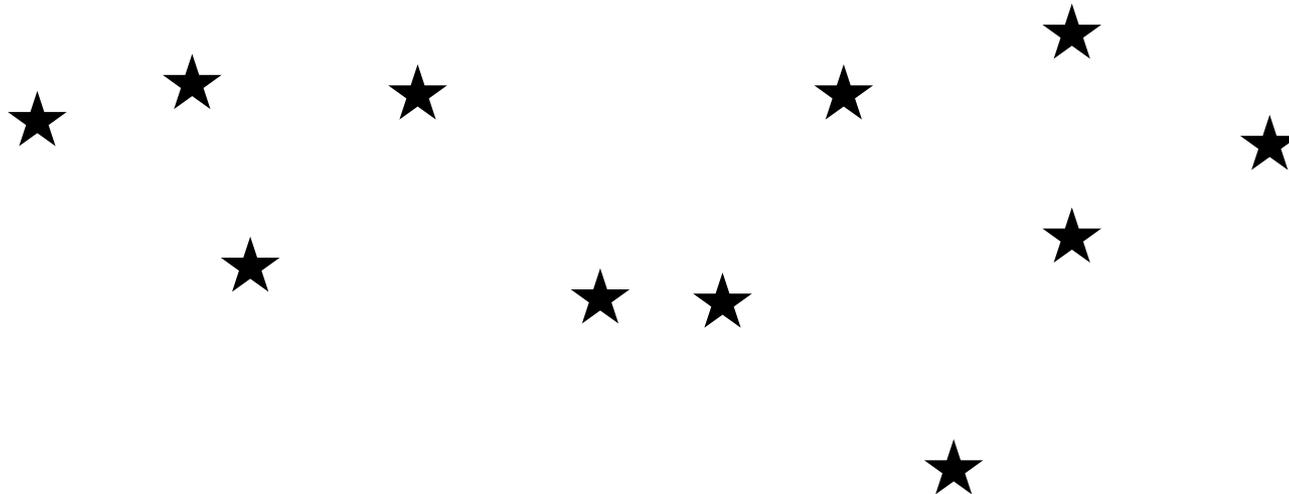
Geschlossenheit: Elemente, die geschlossene Figur bilden, werden als zusammengehörig wahrgenommen



Binnengliederung



Gemeinsames Schicksal: Elemente mit gleicher Veränderung, Bewegungsrichtung, Entwicklung usw. als zusammengehörig wahrgenommen



Wichtige Gestaltungsprinzipien: Beispiel 08.02



Ziel

- Erweiterung der Oberfläche um weitere Elemente und Einhaltung von Gestaltungsprinzipien

Aufgabenstellung

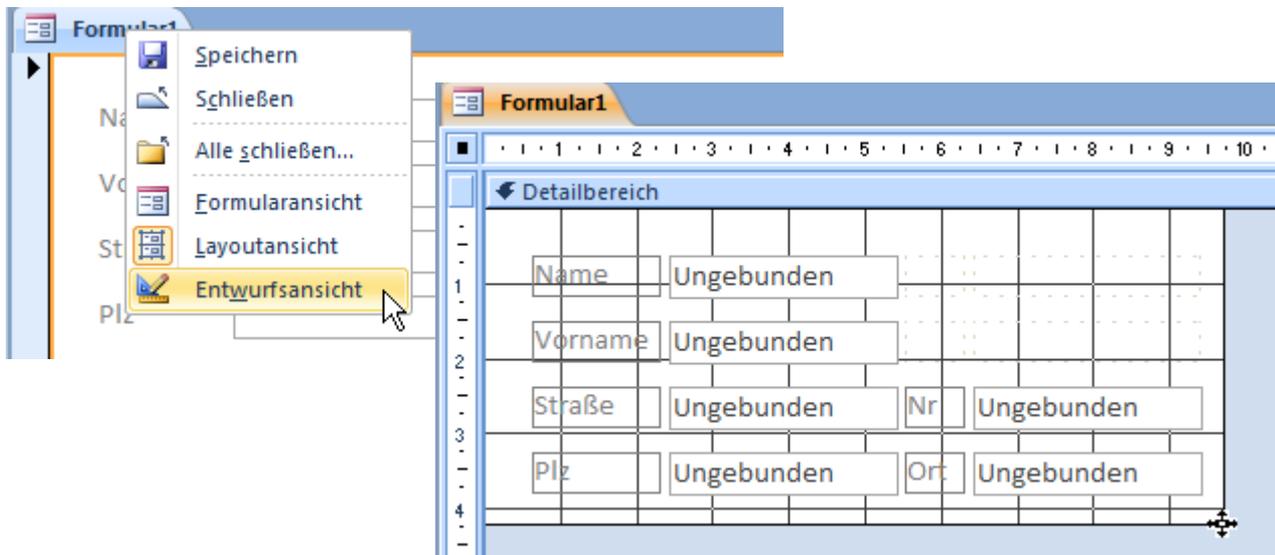
- Erweitern Sie die Oberfläche aus dem vorherigen Beispiel um Schaltflächen
 - zum Löschen
 - zum Speichern von Änderungen
 - zum Anzeigen des nächsten Kunden
 - zum Anzeigen des vorherigen Kunden
- Achten Sie insbesondere auf die Steuerung der Aufmerksamkeit des Benutzers, auf die Einhaltung der Gliederungsprinzipien

Wichtige Gestaltungsprinzipien: Beispiel 08.02



Schritt 1

- Wechsel in den Entwurfsansicht: Rechtsklick auf Formular und Auswahl und vergrößern durch Ziehen

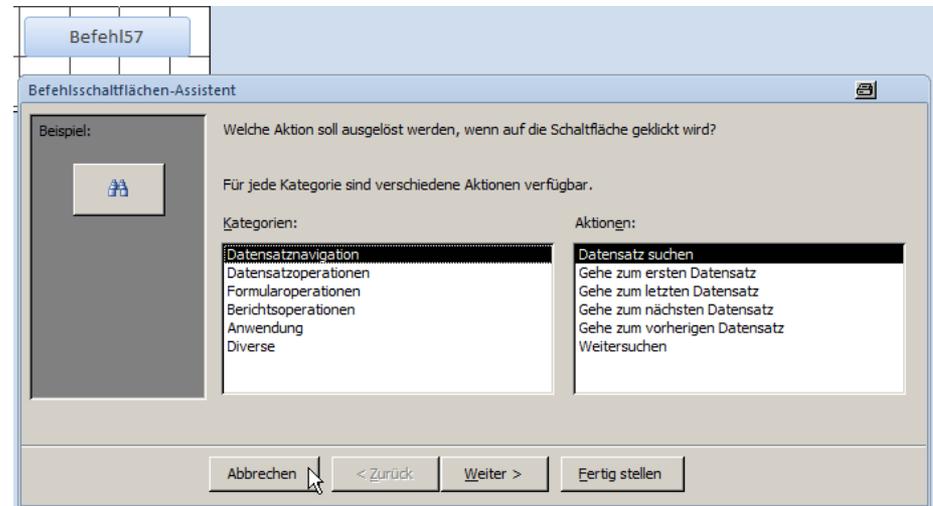
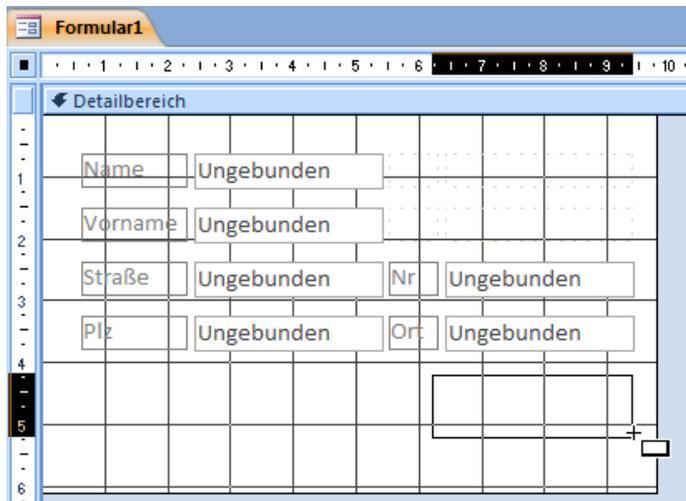


Wichtige Gestaltungsprinzipien: Beispiel 08.02



Schritt 2

- Schaltfläche auswählen und frei positionieren
- Befehlsschaltfläche-Assistent mit Abbrechen beenden
- Bezeichnung durch Doppelklick vergeben

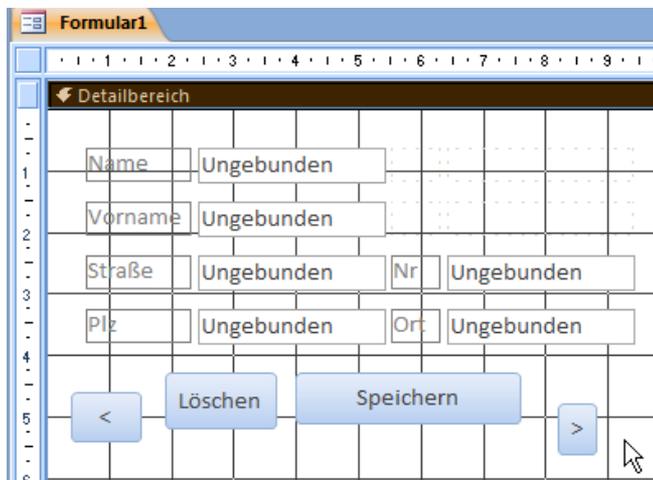


Wichtige Gestaltungsprinzipien: Beispiel 08.02



Weitere Schritte

- analog zu Schritt 2 weitere Schaltflächen hinzufügen
- zur Layoutverbesserung Formularentwurfstools > Anordnen > Größe/Abstand bzw. Ausrichten





Wichtige Gestaltungsrichtlinien

Softwareergonomie nach DIN EN ISO 9241

- Gebrauchstauglichkeit
- Grundsätze der Dialoggestaltung
- Grundsätze der Informationsdarstellung

Weitere

- Bildschirmarbeitsverordnung (BildscharbV)
- ...



Wichtige Gestaltungsrichtlinien

Softwareergonomie nach DIN EN ISO 9241

- Gebrauchstauglichkeit
- Grundsätze der Dialoggestaltung
- Grundsätze der Informationsdarstellung

Weitere

- Bildschirmarbeitsverordnung (BildscharbV)
- ...

Maß, wie effektiv, effizient und insgesamt zufriedenstellend die Software genutzt werden kann.



Wichtige Gestaltungsrichtlinien

Softwareergonomie nach DIN EN ISO 9241

- Gebrauchstauglichkeit
- Grundsätze der Dialoggestaltung
- Grundsätze der Informationsdarstellung

Weitere

- Bildschirmarbeitsverordnung (BildscharbV)
- ...

Aufgabenangemessenheit
Selbstbeschreibungsfähigkeit
Steuerbarkeit
Erwartungskonformität
Fehlertoleranz
Lernförderlichkeit
Individualisierbarkeit



Wichtige Gestaltungsrichtlinien

Softwareergonomie nach DIN EN ISO 9241

- Gebrauchstauglichkeit
- Grundsätze der Dialoggestaltung
- Grundsätze der Informationsdarstellung

Weitere

- Bildschirmarbeitsverordnung (BildscharbV)
- ...

Aufgabenangemessenheit

- Arbeitsaufgabe ist effektiv und effizient zu erledigen
- ausreichende Informationsdarstellung
- geeignete Eingabemöglichkeiten in passender Reihenfolge
- Vorbelegungen und Positionierung der Eingabemarkierung
- keine systembedingten Zusatzaufgaben

Selbstbeschreibungsfähigkeit

Steuerbarkeit

Erwartungskonformität

Fehlertoleranz

Lernförderlichkeit

Individualisierbarkeit



Wichtige Gestaltungsrichtlinien

Softwareergonomie nach DIN EN ISO 9241

- Gebrauchstauglichkeit
- Grundsätze der Dialoggestaltung
- Grundsätze der Informationsdarstellung

Weitere

- Bildschirmarbeitsverordnung (BildscharbV)
- ...

Aufgabenangemessenheit

Selbstbeschreibungsfähigkeit

- jeder Schritt durch Rückmeldung des Systems unmittelbar verständlich oder bei Bedarf erklärt
- Begriffe/Symbole den Fachbegriffen der Arbeitsaufgabe angepasst
- aktueller Bearbeitungsstand sichtbar
- Hilfe verfügbar und auf Kontext (z.B. aktueller Arbeitsschritt, einzelne Felder) bezogen

Steuerbarkeit

Erwartungskonformität

Fehlertoleranz

Lernförderlichkeit

Individualisierbarkeit



Wichtige Gestaltungsrichtlinien

Softwareergonomie nach DIN EN ISO 9241

- Gebrauchstauglichkeit
- Grundsätze der Dialoggestaltung
- Grundsätze der Informationsdarstellung

Weitere

- Bildschirmarbeitsverordnung (BildscharbV)
- ...

Aufgabenangemessenheit

Selbstbeschreibungsfähigkeit

Steuerbarkeit

- Benutzer kann Dialogablauf bis zum Erreichen des Ziels beeinflussen
- Art der Interaktion (z.B. Menü, Abkürzungstaste, Maus) wählbar
- Abfolge von Teilaufgaben (z.B. Reihenfolge, Abkürzung) beeinflussbar
- Unterbrechung und Wiederaufnahme ohne Datenverlust
- Rückgängig-Funktion

Erwartungskonformität

Fehlertoleranz

Lernförderlichkeit

Individualisierbarkeit



Wichtige Gestaltungsrichtlinien

Softwareergonomie nach DIN EN ISO 9241

- Gebrauchstauglichkeit
- Grundsätze der Dialoggestaltung
- Grundsätze der Informationsdarstellung

Weitere

- Bildschirmarbeitsverordnung (BildscharbV)
- ...

Aufgabenangemessenheit
Selbstbeschreibungsfähigkeit
Steuerbarkeit

Erwartungskonformität

- bisherige Kenntnisse/ Erfahrungen anwendbar
- wiederkehrende Interaktionsmuster
- einheitliche Layout-/Dialoggestaltung
- Informationen, Objekte und Meldungen werden an einheitlichen Orten präsentiert
- einheitliche Antwortzeiten

Fehlertoleranz

Lernförderlichkeit

Individualisierbarkeit



Wichtige Gestaltungsrichtlinien

Softwareergonomie nach DIN EN ISO 9241

- Gebrauchstauglichkeit
- Grundsätze der Dialoggestaltung
- Grundsätze der Informationsdarstellung

Weitere

- Bildschirmarbeitsverordnung (BildscharbV)
- ...

Aufgabenangemessenheit
Selbstbeschreibungsfähigkeit
Steuerbarkeit
Erwartungskonformität

Fehlertoleranz

- Ergebnis trotz erkennbar fehlerhafter Eingaben mit minimalem/ohne Korrekturaufwand erreichbar
- Plausibilitätsprüfungen und Warnung vor Aktionen mit weitreichenden Konsequenzen
- Lokalisierung von Fehlerquellen und Aufmerksamkeitssteuerung
- Verständlichkeit der Fehlermeldung und Korrekturvorschlag

Lernförderlichkeit

Individualisierbarkeit



Wichtige Gestaltungsrichtlinien

Softwareergonomie nach DIN EN ISO 9241

- Gebrauchstauglichkeit
- Grundsätze der Dialoggestaltung
- Grundsätze der Informationsdarstellung

Weitere

- Bildschirmarbeitsverordnung (BildscharbV)
- ...

Aufgabenangemessenheit
Selbstbeschreibungsfähigkeit
Steuerbarkeit
Erwartungskonformität
Fehlertoleranz

Lernförderlichkeit

- Benutzer wird während des Erlernens unterstützt
- Vorhandensein Lernprogramm und weitere Zusatzinformationen (z.B. Verweise Fachinformation)
- Berücksichtigung verschiedener Lernstrategien und Wiedererlernen
- zusätzliche Informationen während der Bearbeitung (z.B. Hinweise auf Abkürzungstasten in Tooltips)

Individualisierbarkeit



Wichtige Gestaltungsrichtlinien

Softwareergonomie nach DIN EN ISO 9241

- Gebrauchstauglichkeit
- Grundsätze der Dialoggestaltung
- Grundsätze der Informationsdarstellung

Weitere

- Bildschirmarbeitsverordnung (BildscharbV)
- ...

Aufgabenangemessenheit
Selbstbeschreibungsfähigkeit
Steuerbarkeit
Erwartungskonformität
Fehlertoleranz
Lernförderlichkeit

Individualisierbarkeit

- Anpassung an individuelle Belange und Fähigkeiten bezüglich der Arbeitsaufgabe
- Einstellung zu Hinweis- und Warnmeldungen
- Erfahrungsniveau (z.B. Anfänger und Experte) wird unterstützt
- Informationsmenge einstellbar
- Anpassung von Icons, Bezeichnungen und Einstellungen
- Makros zur Automatisierung



Wichtige Gestaltungsrichtlinien

Softwareergonomie nach DIN EN ISO 9241

- Gebrauchstauglichkeit
- Grundsätze der Dialoggestaltung
- Grundsätze der Informationsdarstellung

Weitere

- Bildschirmarbeitsverordnung (BildscharbV)
- ...

Aufgabenangemessenheit
Selbstbeschreibungsfähigkeit
Steuerbarkeit
Erwartungskonformität
Fehlertoleranz
Lernförderlichkeit
Individualisierbarkeit



Wichtige Gestaltungsrichtlinien

Softwareergonomie nach DIN EN ISO 9241

- Gebrauchstauglichkeit
- Grundsätze der Dialoggestaltung
- Grundsätze der Informationsdarstellung

Weitere

- Bildschirmarbeitsverordnung (BildscharbV)
- ...

Erkennbarkeit

- Aufmerksamkeit auf benötigte Information gelenkt

Unterscheidbarkeit

- angezeigte Information von anderen Daten unterscheidbar

Lesbarkeit

- Information leicht zu lesen

Verständlichkeit

- Bedeutung leicht verständlich, eindeutig, vermittelbar und erkennbar

Klarheit

- Informationsgehalt schnell und genau vermittelt

Kompaktheit/Prägnanz

- nur Information geben, die für Erledigung der Aufgabe notwendig

Konsistenz

- gleiche Information innerhalb der Anwendung stets auf die gleiche Art dargestellt

GUI Style Guides



Bedeutung

- regeln über die Gebrauchstauglichkeit hinausgehende Aspekte der Benutzerfreundlichkeit und des Benutzererlebnisses
- unterstützen Erwartungskonformität über Anwendungsgrenzen hinweg
- sorgen für einheitliches Erscheinungsbildes insbesondere in Verbindung mit einer Plattform, einem Produkt oder einer Marke

Inhalte, u.a.

- Gliederung von Fenstern
- Anordnung von Oberflächenelementen
- Gestaltung von Informationen und Meldungen
- Spezifische Einsatzmöglichkeiten
- ...



GUI Style Guides

Beispiele

- Mac OS X Human Interface Guidelines (siehe [7])
- Windows User Experience Interaction Guidelines for
 - Windows XP and Windows 2000 (siehe [1])
 - Windows[®] 7 and Windows Vista[®] (siehe [8])
 - Windows 8 (?)
- NOKIA Design and User Experience Library (siehe [2])
 - Design for Windows Phone 8 (siehe [3])
 - Design for Nokia Asha (siehe [4])
- Android User Interface Guide (siehe [6])
- iOS Human Interface Guidelines (siehe [5])

Beispiel: Oberflächen für Windows Phone 8



scroll through long lists. The spark jump grid can be an alphabetical list, a grid overlays the list, an item is selected, and the user is then taken back to



QUICK AMP GRID
Young attributed list



QUICK AMP GRID
Young grid header list



QUICK AMP GRID
Young grid list

Quelle: [3]

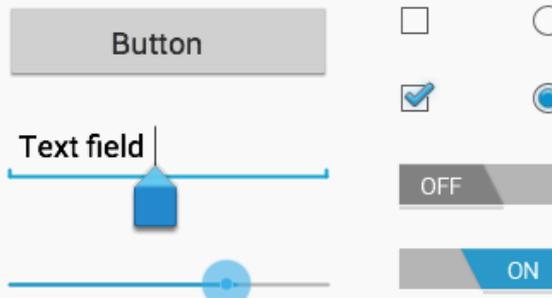
Beispiel: Oberflächen für Android



Input Controls

Input controls are the interactive components in your app's user interface. Android provides a wide variety of controls you can use in your UI, such as buttons, text fields, seek bars, checkboxes, zoom buttons, toggle buttons, and many more.

Adding an input control to your UI is as simple as adding an XML element to your *XML layout*. For example, here's a layout with a text field and button:



Normal view

A notification in normal view appears in an area that's up to 64 dp tall. Even if you create a notification with a big view style, it will appear in normal view until it's expanded. This is an example of a normal view:

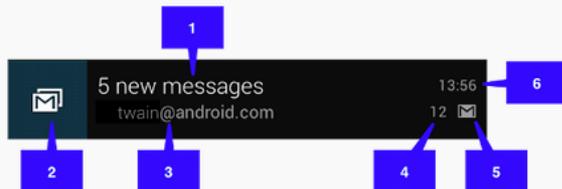


Figure 3. Notification in normal view.

The callouts in the illustration refer to the following:

1. Content title
2. Large icon
3. Content text
4. Content info
5. Small icon
6. Time that the notification was issued. You can set an explicit value with `setWhen()`; if you don't it defaults to the time that the system received the notification.

Quelle: [6]

of dialog designs and is often the only dialog class you'll use for an alert dialog:

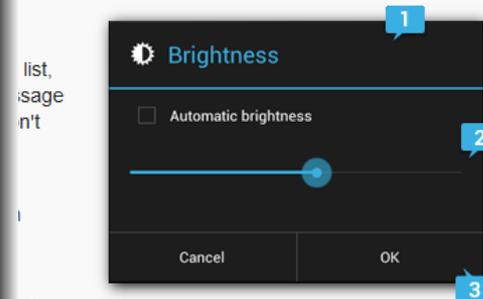


Figure 2. The layout of a dialog.

Contextual Menus

Contextual menus provide actions that are specific to the context frame in which they are used. They are most often used in a `ListView`, `GridView`, or `RecyclerView`, in which the actions on each item are different. Contextual menus provide contextual actions for each item in a list.

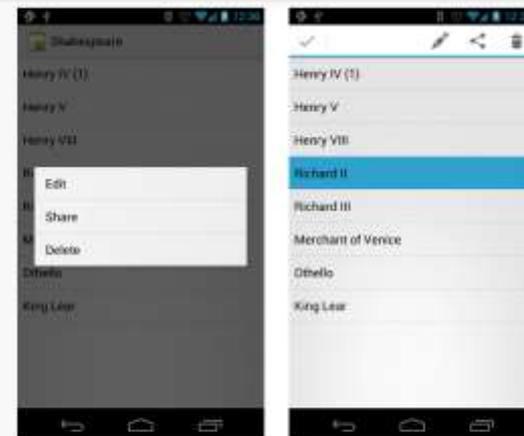


Figure 3. Screenshots of a floating context menu (left) and the contextual action bar (right).

Contextual menus are used in two modes. In *floating mode*, this mode is active, users can perform an action on multiple items at once (if your app allows it).

Beispiel: Oberflächen für iOS 6



Distinct, highly focused screens. Each screen displays one aspect of the Mail experience: account list, mailbox list, message list, message view, and composition view. Within a screen, the active content is



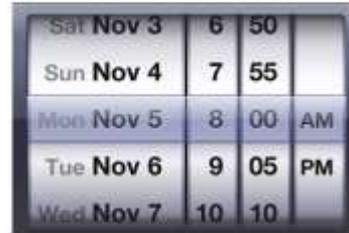
Easy, predictable navigation. Making one tap per screen, people do one thing at a time, and a back button that makes it easy for them to retrace their steps.

Simple, tappable controls, available when needed. Because not all controls are visible on every screen, when people are viewing a message



Different types of feedback for different tasks. When people do one thing at a time, they can hear a distinctive sound. By looking at the subtle text in the message

Date Picker
A **date picker** displays components of date and time, such as hours, minutes, days, and years.



To learn how to define a date picker in your code, see [UIDatePicker Class Reference](#).

Be sure to choose a status bar appearance that coordinates with the rest of your iPhone app. For example, avoid using a translucent status bar if the navigation bar is opaque.

On iPhone, set whether the change in status bar color should be animated. Note that the animation causes the old status bar to slide up until it disappears off the screen while the new status bar slides into place.

Navigation Bar

A **navigation bar** enables navigation through an information hierarchy and, often, provides a visual cue for the current screen's position in the hierarchy.



Modal View

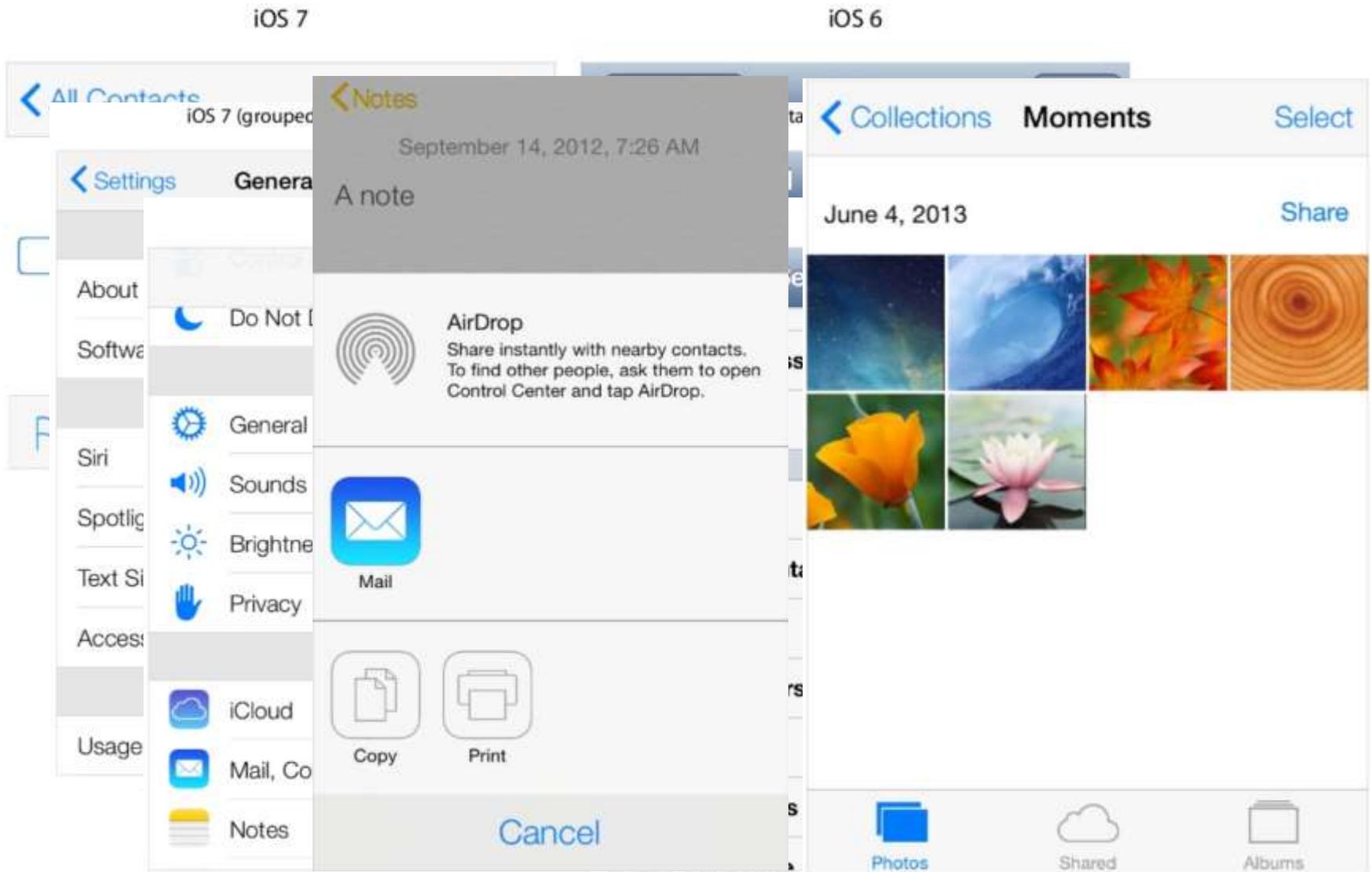
A **modal view** (that is, a view presented modally) provides self-contained functionality in the context of the current task or workflow.



To learn more about defining a modal view in your code, see [UIViewController Class Reference](#).

Quelle: [5]

Beispiel: Oberflächen für iOS 7 im Vergleich zu iOS 6



Quelle: [9]

Beispiel: Oberflächen für Nokia Asha



Activity screen icon:

Glyphs, shapes or logos can be placed (centred) within the 30 x 30 px focal zone

The coloured background and the bounding box have a size of 50 x 50 px.

The device software crops the square shape of the activity screen icon and applies the highlight rim. Size and appearance of it is necessary to check that the created glyph fits nicely into both activity screen and the surround launcher icon shape. The activity layer to preview the appearance of the final launcher icon. This is a custom graphic.



Launcher icon:

The background is cropped to a 42 x 42 px surround shape

Activity screen icon templates for Nokia Asha

The Nokia Asha Icon Toolkit offers templates to help you create your icons using Adobe Illustrator, Inkscape, or Adobe Photoshop.

You can simply place your graphic on top of a preset background, add shadows, and you have an activity screen icon ready for saving.

Device software will create a launcher icon automatically based on it. The following example shows how a piece of graphic is turned into an icon. [Nokia_Asha_Activity_Screen_illustrator_Template.aic](#)



original graphic



activity screen icon created with the help of the template



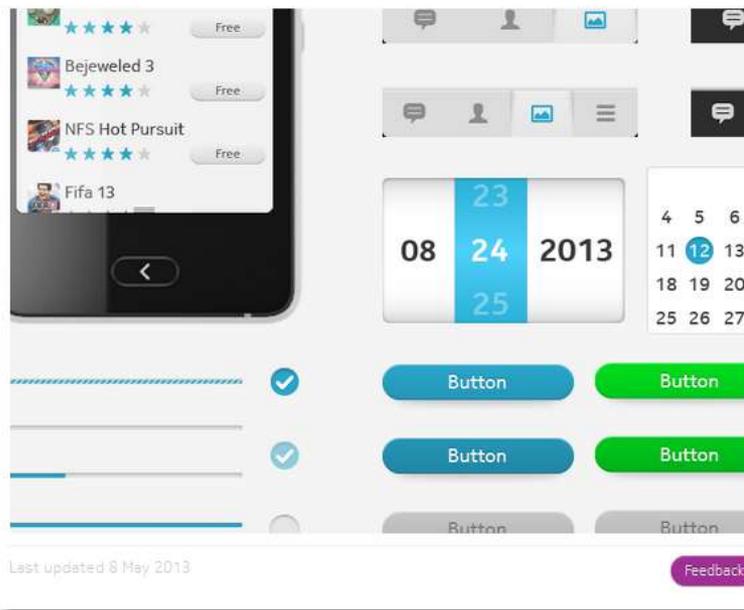
launcher icon created automatically by the device software

Grid view

Grid view examples:

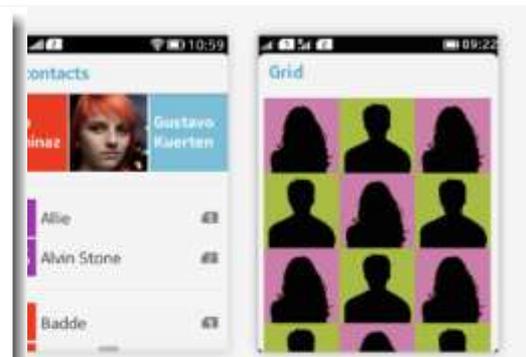
Building blocks - LWUIT

This component library presents all the basic building blocks available to build applications for the Nokia Asha UI with LWUIT. The guidelines of each component describe the key functionalities and interactions of the component. See [Developing applications with Java](#) section for more information about implementation alternatives.



Last updated 6 May 2013

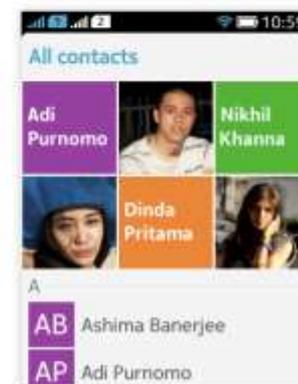
Feedback



LWUIT: For LCDUI, please refer to [Lists and grids](#).

normally locate

contacts are shown
contacts are shown
is not limited.



Quelle: [4]

Inhalt

Einordnung

Rückblick

Ausgangspunkt



Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick



Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick



BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Wirtschaftsinformatik 1

LE 08 – Oberflächen und Ereignisse

Prof. Dr. Thomas Off

<http://www.ThomasOff.de/lehre/beuth/wi1>

Einordnung



06 – Zusammenfassung

05 – Debugger und Testen

04 – Fortgeschrittene Konzepte

04.A
Oberflächen (Teil 1)
Elemente und
Eigenschaften

04.B
Oberflächen (Teil 2)
Ereignisverarbeitung

04.C
Zugriff auf
Dateisystem und
Anwendungen

03 – Grundkonzepte

03.A
Wert
Ausdruck
Variable
Konstante
Datentyp

03.B
Bedingte
Ausführung/
Verzwei-
gungen

03.C
Schleifen

03.D
Felder
Mengen

03.E
Prozedur
Funktion
Modul

02 – Grundlagen der Programmierung

01 – Grundbegriffe der Wirtschaftsinformatik



Rückblick



Rückblick



Wichtige Oberflächenelemente

- Eingabe und Auswahl
 - Textfelder
 - Aufklappliste/Kombinationsfeld
 - Mehrfachauswahllisten
 - Radioknöpfe (Optionsfeld)
 - Kontrollkästchen (Checkbox)
- Aktionselemente
 - Schaltfläche (einfach)
 - Umschaltfläche (Toggle)
 - Menüeinträge
- Container
 - Rahmen/Gruppen
 - Registerkartensatz mit Registerkarten
 - Menüs
 - Fenster/Dialoge (in Access als Formulare)

Feld

Mehrzeiliges Feld

Kombinationslistenfeld

▼
Alternative 1
Alternative 2
Alternative 3
Alternative 4

Radioknöpfe

Alternative 1
 Alternative 2
 Alternative 3

Möglichkeiten

Möglichkeit 1
 Möglichkeit 2
 Möglichkeit 3

Mehrfachauswahlliste

Möglichkeit 1	12, 50 €
Möglichkeit 2	24,80 €
Möglichkeit 3	37,50 €

Anschrift

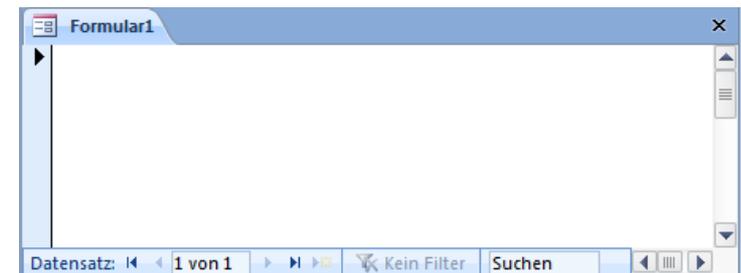
Straße Nr.

PLZ Ort

Namen Adressen

Straße Nr.

PLZ Ort





Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick

Genereller Aufbau einer Anwendung

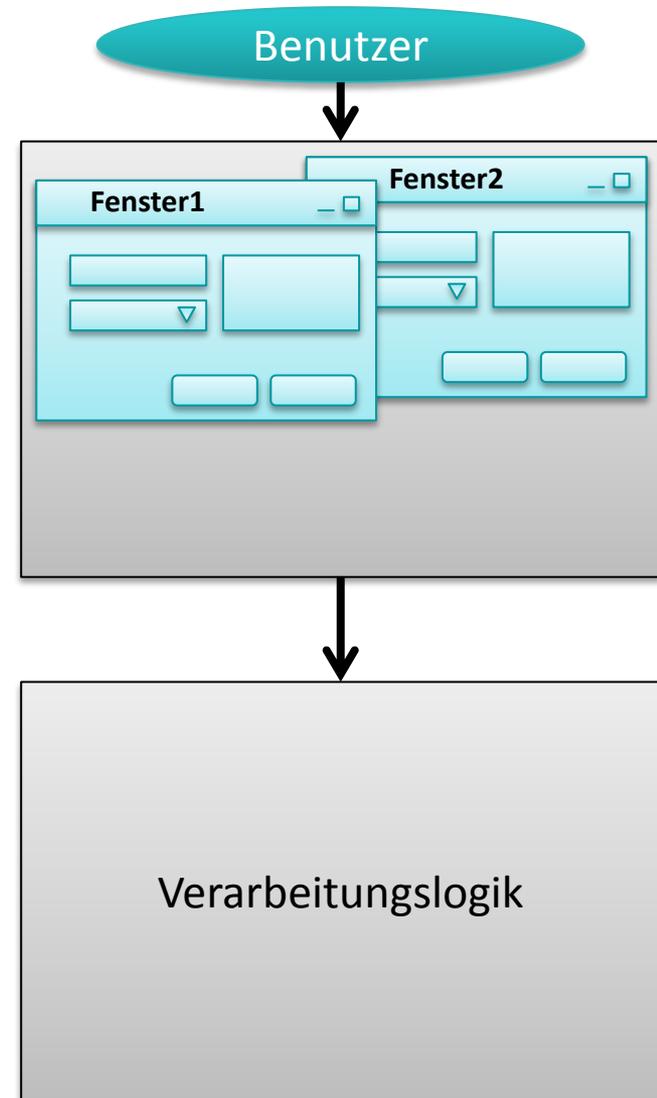


Benutzeroberfläche

- Oberflächenelemente
 - Felder
 - Schaltflächen
 - Gliederungselemente
 - Fenster/Dialoge
 - ...

Verarbeitungslogik

- Module
 - Prozeduren
 - Funktionen



Genereller Aufbau einer Anwendung

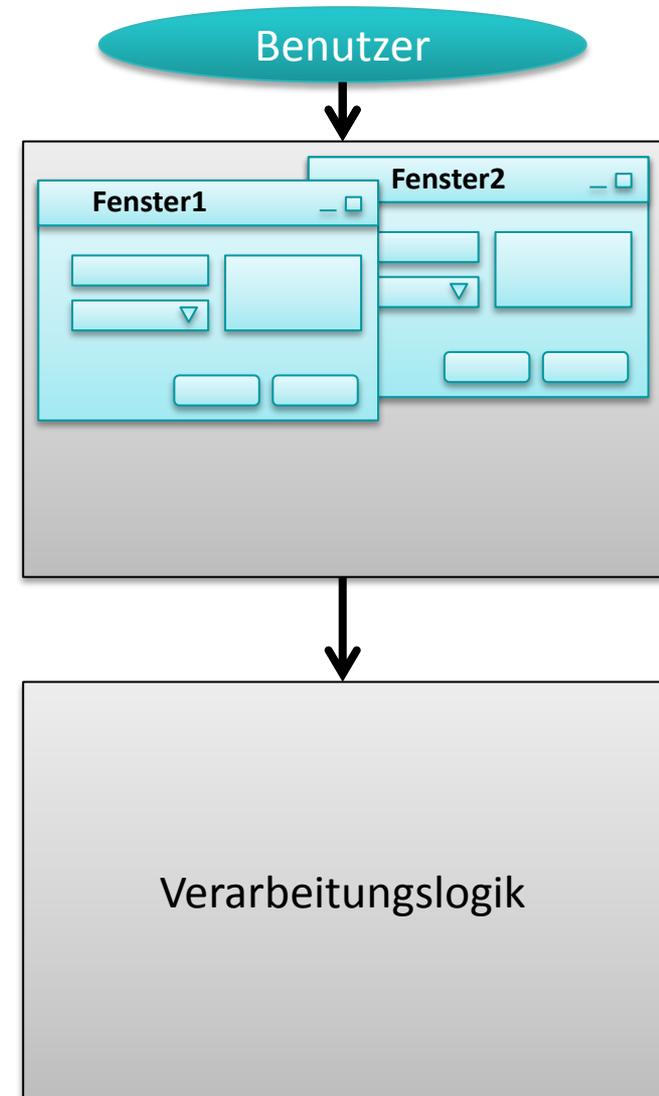


Benutzeroberfläche

- Oberflächenelemente
 - Felder
 - Schaltflächen
 - Gliederungselemente
 - Fenster/Dialoge
 - ...
- Reaktion auf Benutzeraktionen

Verarbeitungslogik

- Module
 - Prozeduren
 - Funktionen



Genereller Aufbau einer Anwendung

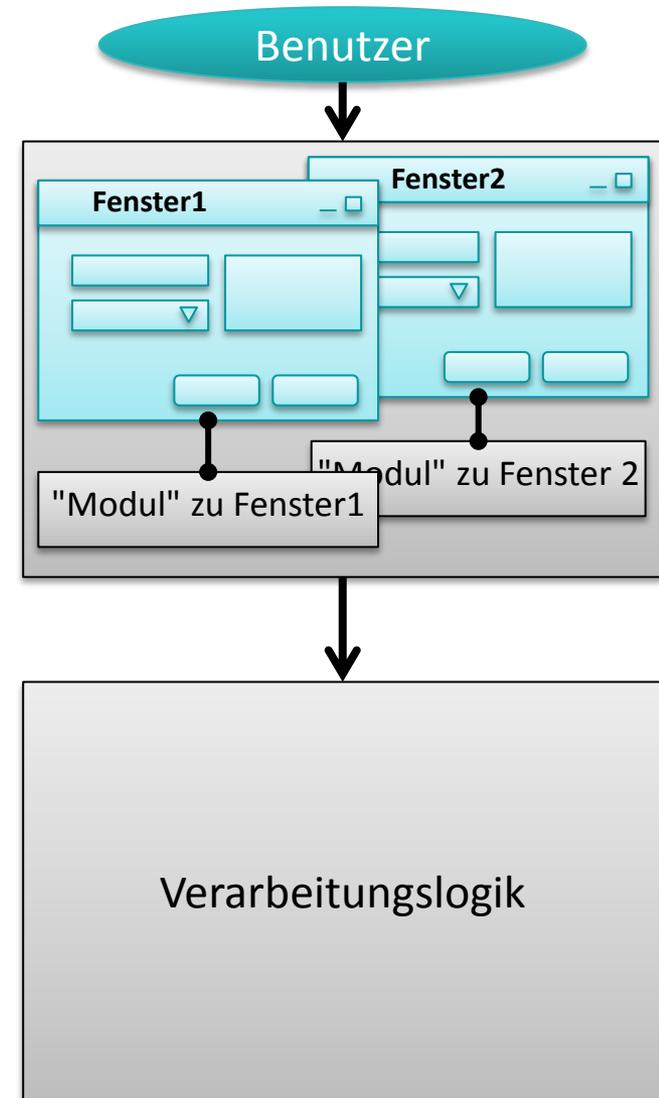


Benutzeroberfläche

- Oberflächenelemente
 - Felder
 - Schaltflächen
 - Gliederungselemente
 - Fenster/Dialoge
 - ...
- Reaktion auf Benutzeraktionen
 - Module (Klassenobjekte)

Verarbeitungslogik

- Module
 - Prozeduren
 - Funktionen



Genereller Aufbau einer Anwendung

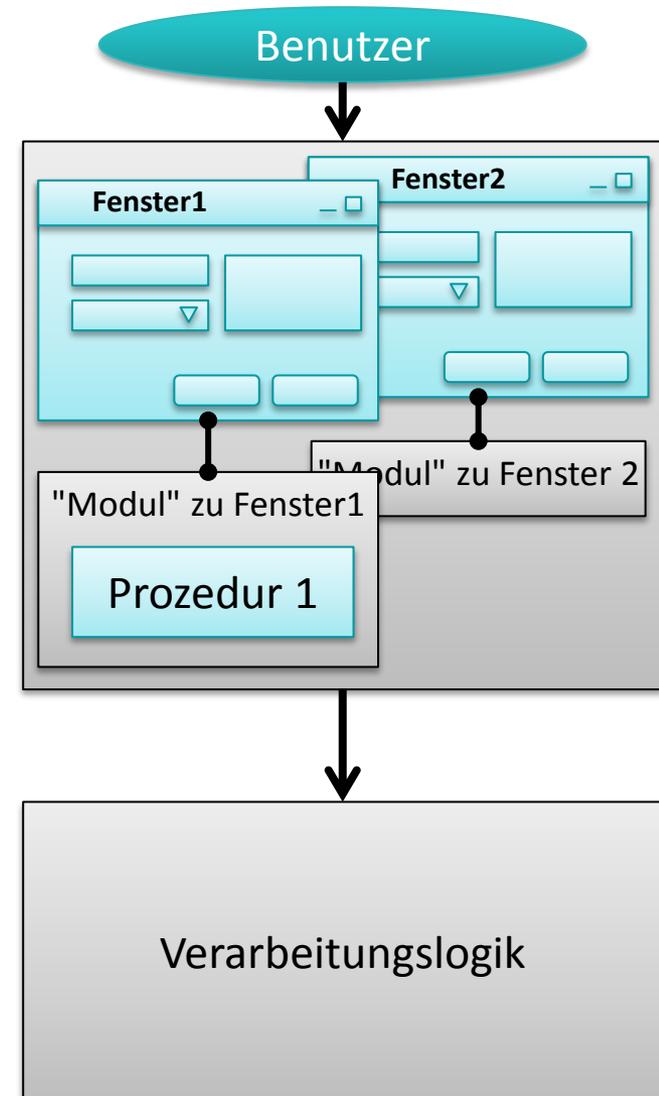


Benutzeroberfläche

- Oberflächenelemente
 - Felder
 - Schaltflächen
 - Gliederungselemente
 - Fenster/Dialoge
 - ...
- Reaktion auf Benutzeraktionen
 - Module
 - Steuerung der Oberfläche
 - Aufruf der Verarbeitungslogik
 - Prozeduren

Verarbeitungslogik

- Module
 - Prozeduren
 - Funktionen





Ereignisse als Reaktion auf Benutzeraktionen

Benutzer führt Aktionen auf der Oberfläche aus

- Klicken auf Schaltflächen
- Eingeben von Daten in Felder
- Auswahl von Listeneinträgen
- Verlassen von Feldern
- Mouse bewegen

Als Reaktion werden Ereignisprozeduren aufgerufen

- btnSchalflaeche_Click()
- txtTextfeld_KeyPress()
- cmbAuswahl_Change()
- txtTextfeld_Exit()
- Form_MouseMove()

The screenshot shows the Microsoft Access interface. On the left, the 'Formulare' (Forms) pane lists 'frmBeispielFormular1' and 'frmBeispielFormular2'. The main window displays 'frmBeispielFormular1' with a 'Detailbereich' (Detail Area) containing a 'Kombinationsfeld' (Combobox) control. The 'Eigenschaftenblatt' (Properties Sheet) for this control is open, showing 'Auswahltyp: Kombinationsfeld' and 'cmbAuswahl'. The 'Ereignis' (Events) tab is active, and the 'Bei Änderung' (On Change) event is set to '[Ereignisprozedur]' (Event Procedure). Other events listed include 'Bei Geändert', 'Bei Nicht in Liste', and 'Bei Fokuserhalt'.

Ereignisse als Reaktion auf Benutzeraktionen



Benutzer führt Aktionen auf der Oberfläche aus

- Klicken auf Schaltflächen
- Eingeben von Daten in Felder
- Auswahl von Listeneinträgen
- Verlassen von Feldern
- Mouse bewegen

Als Reaktion werden Ereignisprozeduren aufgerufen

- btnSchalflaeche_Click()
- txtTextfeld_KeyPress()
- cmbAuswahl_Change()
- txtTextfeld_Exit()

The screenshot displays the Microsoft Access interface. On the left, the 'Formulare' (Forms) pane shows 'frmBeispielFormular1' and 'frmBeispielFormular2'. The 'Eigenschaftenblatt' (Property Sheet) for 'cmbAuswahl' is open, showing 'Auswahltyp: Kombi...'. The 'Ereignisblatt' (Event Sheet) is also open, showing a list of events: 'Bei Geändert', 'Bei Änderung', 'Bei Nicht in Liste', and 'Bei Fokuserhalt'. The 'Bei Änderung' event is selected, and its procedure is '[Ereignisprozedur]'. The 'Code' window on the right shows the VBA code for the 'Change' event of 'cmbAuswahl':

```
Option Compare Database
Option Explicit

Private Sub cmbAuswahl_Change ()
|
End Sub
```



Ereignisse als Reaktion auf Benutzeraktionen

Benutzer führt Aktionen auf der Oberfläche aus

- Klicken auf Schaltflächen
- Eingeben von Daten in Felder
- Auswahl von Listeneinträgen
- Verlassen von Feldern
- Mouse bewegen

Als Reaktion werden Ereignisprozeduren aufgerufen

- btnSchalflaeche_Click()
- txtTextfeld_KeyPress()
- cmbAuswahl_Change()
- txtTextfeld_Exit()

The screenshot illustrates the VBA development environment for Microsoft Access. It shows the following components:

- Object Explorer:** Displays the project structure, including the form `frmBeispielFormular1` and its controls.
- Properties Window:** Shows the properties for the selected control, `cmbAuswahl`, with the `Change` event selected.
- VBA Code Window:** Displays the event procedure code for `cmbAuswahl_Change`. The code includes options for comparing the database and explicit options, followed by a private sub procedure definition.

```
Option Compare Database
Option Explicit

Private Sub cmbAuswahl_Change ()
End Sub
```

Beispiele für Ereignisse und deren Reihenfolge



Reihenfolge der Ereignisse

- beim Öffnen und Schließen eines Formulars
 - Open, Load, Resize, Activate
 - Unload, Deactivate, Close
- bei Betreten und Verlassen von Oberflächenelementen (allgemein)
 - Enter, GotFocus
 - Exit, LostFocus

Weitere Ereignisse

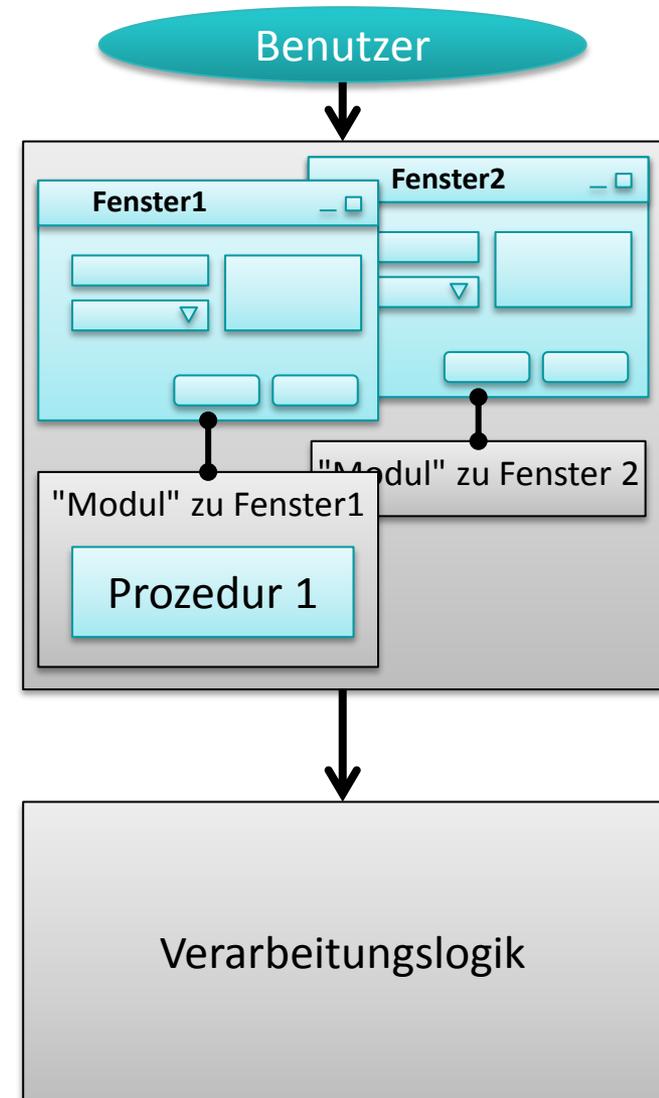
- KeyDown, KeyPress, Change, KeyUp
- Click, Mouse_Down, Mouse_Up



Reaktion auf Benutzeraktionen

Ereignisprozeduren bieten Zugriffsmöglichkeit auf die Oberflächenelemente und dienen zum

- Steuern der Elemente auf der Oberfläche
 - Aktivieren/Dekativieren bzw. Einblenden/Ausblenden von Elementen
 - Navigation zwischen Fenstern
 - ...
- Aufruf der Verarbeitungslogik
 - Übergabe der eingegebenen Daten zur Verarbeitung
 - Ermitteln der anzuzeigenden Daten
 - Ausführen von komplexen Berechnungen
 - ...





Reaktion auf Benutzeraktionen

Ereignisprozeduren bieten Zugriffsmöglichkeit auf die Oberflächenelemente und dienen zum

- Steuern der Elemente auf der Oberfläche
 - Aktivieren/Dekativieren bzw. Einblenden/Ausblenden von Elementen
 - Navigation zwischen Fenstern
 - ...
- Aufruf der Verarbeitungslogik
 - Übergabe der eingegebenen Daten zur Verarbeitung
 - Ermitteln der anzuzeigenden Daten
 - Ausführen von komplexen Berechnungen
 - ...

frmKunden

The screenshot shows a Windows form titled "Kunden". It contains a text box labeled "txtKndName", a dropdown menu, and a button labeled "btnOk". A red arrow points from the text box to the code block below.

"Modul" zu Fenster1

```
' Anwendung kennt alle Formulare
' Jedes Formular kennt seine
' Elemente
Application.Forms("frmKunden")._
Controls("txtKndName")
Me.Controls("txtKndName")
Me.Controls(2)

' Favorit
Me.txtKndName
```

Reaktion auf Benutzeraktionen



Ereignisprozeduren bieten Zugriffsmöglichkeit auf die Oberflächenelemente und dienen zum

- Steuern der Elemente auf der Oberfläche
 - Aktivieren/Dekativieren bzw. Einblenden/Ausblenden von Elementen
 - Navigation zwischen Fenstern
 - ...
- Aufruf der Verarbeitungslogik
 - Übergabe der eingegebenen Daten zur Verarbeitung
 - Ermitteln der anzuzeigenden Daten
 - Ausführen von komplexen Berechnungen
 - ...

frmKunden

Kunden

txtKndName

Bond

btnOk

```
"Modul" zu Fenster1

' Zugriff auf Oberflächenelement-
' eigenschaften
Let Me.txtKndName.Value = "Bond"
```

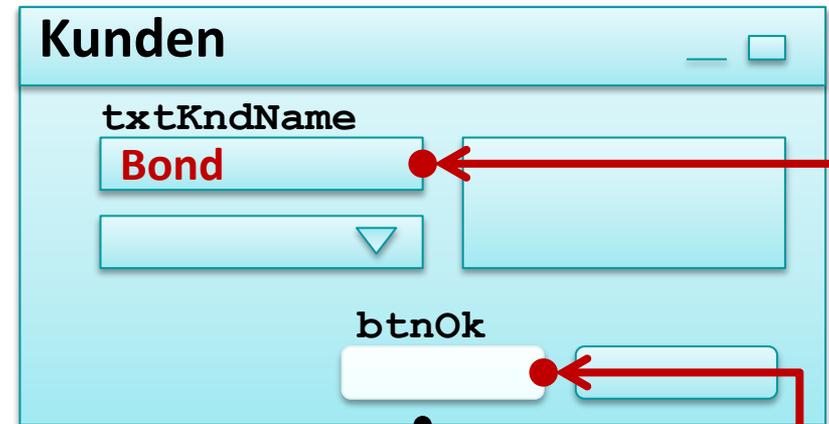
Reaktion auf Benutzeraktionen



Ereignisprozeduren bieten Zugriffsmöglichkeit auf die Oberflächenelemente und dienen zum

- Steuern der Elemente auf der Oberfläche
 - Aktivieren/Dekativieren bzw. Einblenden/Ausblenden von Elementen
 - Navigation zwischen Fenstern
 - ...
- Aufruf der Verarbeitungslogik
 - Übergabe der eingegebenen Daten zur Verarbeitung
 - Ermitteln der anzuzeigenden Daten
 - Ausführen von komplexen Berechnungen
 - ...

frmKunden



```
"Modul" zu Fenster1

' Zugriff auf Oberflächenelement-
' eigenschaften
Let Me.txtKndName.Value = "Bond"

Let Me.btnOk.Enabled = False
```



Reaktion auf Benutzeraktionen

Zugriff über Referenzvariable Me

- Zugriff auf den Wert von Feldern liefert immer String

```
' Syntax (Empfehlung: Hilfsfunktion für Zahlenwerte)
Let <VarString> = Me.<BezeichnerDesFeldes>.Value
Let <VarZahl> = Val(Me.<BezeichnerDesFeldes>.Value)
```

- Genereller Zugriff auf Eigenschaften von Elementen

```
' Generelle Syntax
Let <Var> = Me.<Bez>.<Eigenschaft> ' Lesen
Let Me.<Bez>.<Eigenschaft> = <Var> ' Schreiben/Ändern
```

Beispiele

```
' Wert eines Feldes ermitteln
Let strName = Me.txtName.Value
Let intAlter = Val(Me.txtAlter.Value)
' Sichtbarkeit ändern
Let Me.txtName.Visible = False
```



Reaktion auf Benutzeraktionen

Abhängig von Oberflächenelement, z.B.

- Textfelder (einzeilig/mehrzeilig)
- Aufklapplisten und Auswahllisten
- Radioknöpfe und Checkboxes
- Fenster



Verwendung von Textfeldern



Wert eines Textfeldes

– Ermitteln

```
' Generelle Syntax (ggf. mit Umwandlung in Zahl)
Let <strVar> = Me.<txtBez>.Value ' String lesen
Let <zahlVar> = Val(Me.<txtBez>.Value) ' Zahl lesen
```

– Ändern

```
' Generelle Syntax
Let Me.<txtBez>.Value = <StringWertAusdr>
```

Beispiel

```
' Beispiel
Let Me.txtName.Value = "Müller"
Let Me.txtGebDat.Value = datGebDat
' ...
Let strVorname = Me.txtVorname.Value
Let intAlter = Val(Me.txtAlter.Value)
```



Verwendung von Textfeldern

Sichtbarkeit eines Textfeldes steuern

```
' Generelle Syntax
Let Me.<txtBez>.Visible = True ' Anzeigen
Let Me.<txtBez>.Visible = False ' Verstecken
```

Textfeld aktivieren/deaktivieren

```
' Generelle Syntax
Let Me.<txtBez>.Enabled = True ' Aktivieren
Let Me.<txtBez>.Enabled = False ' Deaktivieren
```

Beispiele

```
' Sichtbarkeit
Let Me.txtName.Visible = True
Let Me.txtVorname.Visible = True

' Aktiviert
Let Me.txtGebDat.Enabled = False
Let Me.txtAlter.Enabled = True
```

Verwendung von Textfeldern (Fortsetzung)

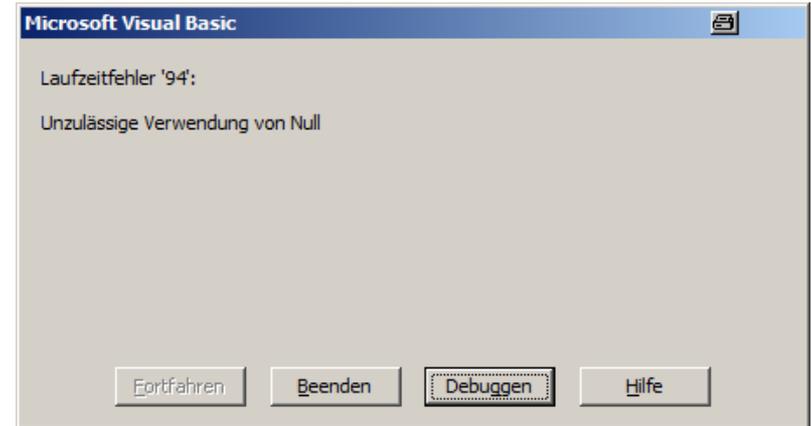


Besonderheit

- Der Wert von Feldern leeren Feldern ist **Null** und kann nicht abgefragt werden

Lösung

- Vor dem Zugriff auf Werte eines Feldes, deren Vorhandensein mit Hilfsfunktion **IsNull ()** prüfen, liefert Boolean



```
' Generelle Syntax  
IsNull (<VariablenWert>)
```

```
' Beispiel  
If IsNull (Me.txtName.Value) Or _  
    IsNull (Me.txtVorname.Value) Then  
    Exit Sub  
End If
```

Verwendung von Textfeldern (Fortsetzung)



Besonderheit

- Leeren von Feldern nicht durch Zuweisung eines leeren String (""), sondern durch Zuweisung von Null
- Nur so erneute Prüfung mit IsNull auf leeres Feld möglich

Beispiel

```
' Generelle Syntax  
Let Me.<Bez>.<Eigenschaft> = Null ' Leeren
```

```
' Beispiel  
Let Me.txtName.Value = Null  
' ...  
If IsNull(Me.txtName.Value) Then  
    Exit Sub  
End If
```

Reaktion auf Benutzeraktionen: Beispiel 08.05



Teil 1: Ziel

- Klick-Ereignis einer Schaltfläche verwenden, um Meldung anzuzeigen

Aufgabenstellung:

- Klickt der Benutzer auf die Schaltfläche "Speichern" soll eine Meldung angezeigt werden, die den Namen und den Vornamen des Kunden anzeigt.



Reaktion auf Benutzeraktionen: Beispiel 08.05



Teil 2: Ziel

- Klick-Ereignis einer Schaltfläche verwenden, um Meldung anzuzeigen

Aufgabenstellung:

- Klickt der Benutzer auf die Schaltfläche "Abbrechen" sollen alle Textfelder der Oberfläche geleert werden



Verwendung von Textfeldern (Fortsetzung)



Relevante Ereignisse eines Textfeldes verarbeiten

- Betreten des Feldes

```
Private Sub txtVorname_Enter()  
    ' ...  
End Sub
```

- Verlassen des Feldes bietet Möglichkeit, das Verlassen abubrechen (d.h. die Eingabemarkierung im Feld zu lassen)

```
' Parameter Cancel wird per Referenz übergeben und  
' ermöglicht "Rückgabe" des Abbruchs  
Private Sub txtVorname_Exit(Cancel As Integer)  
    Call MsgBox("Feld nur verlassen, wenn Wert = 'Max'")."  
    If Me.txtVorname.Value <> "Max" Then  
        ' Abbrechen auf Ja setzen  
        Let Cancel = vbYes  
    End If  
End Sub
```

Verwendung von Textfeldern (Fortsetzung)



Relevante Ereignisse eines Textfeldes verarbeiten

- Drücken einer Taste
 - bekommt als Parameter den Zeichencode der gedrückten Taste
 - verursacht zusätzlich Ereignis "Bei Änderung"

```
Private Sub txtVorname_KeyPress(KeyAscii As Integer)
    ' Umwandeln eingegebener Zeichen in Großbuchstaben
    Dim strZeichen As String
    Dim strZeichenGross As String
    ' Zeichen aus Zeichencode ermitteln mit Hilfsfunktion Chr()
    Let strZeichen = Chr(KeyAscii)
    ' Umwandeln in Großbuchstaben mit Hilfsfunktion UCase()
    Let strZeichenGross = UCase(strZeichen)
    ' Zeichen anzeigen
    Call MsgBox("Taste: " & strZeichen & " umgewandelt in: " & _
                strZeichenGross)
    ' Neuen Zeichencode aus großem Buchstaben ermitteln mit Asc()
    Let KeyAscii = Asc(strZeichenGross)
End Sub
```



Verwendung von Textfeldern (Fortsetzung)

Relevante Ereignisse eines Textfeldes verarbeiten

- Bei Änderung

```
Private Sub txtVorname_Change()  
    ...  
End Sub
```

- Bei Erhalt/Verlust des Eingabefokus

- vergleichbar mit bei Betreten/Verlassen

```
Private Sub txtVorname_GotFocus()  
    ...  
End Sub  
  
Private Sub txtVorname_LostFocus()  
    ...  
End Sub
```



Reaktion auf Benutzeraktionen

Abhängig von Oberflächenelement, z.B.

- Textfelder (einzeilig/mehrzeilig)
- Aufklapplisten und Auswahllisten
- Radioknöpfe und Checkboxes
- Fenster





Verwendung von Aufklapplisten

Wert einer Aufklappliste aus der "gebundenen" Spalte

– Ermitteln

```
' Generelle Syntax (gebundene Spalte enthält eine Zahl)  
Let <intVar> = Me.<cmbBez>.Value
```

– Ändern

```
' Generelle Syntax (gebundene Spalte erhält Zahlenwert)  
Let Me.<cmbBez>.Value = <ZahlWertAusdr>
```

Beispiel

Anrede: Herr
Name: Herr
Straße: Familie
PLZ: Ort

Eigenschaftenblatt	
Auswahltyp: Kombinationsfeld	
cmbAnrede	
Format Daten Ereignis Andere Alle	
Steuerelementinhalt	
Datensatzherkunft	1;"Herr";2;"Frau";3;"Familie";4;"Firma"
Herkunftstyp	Wertliste
Gebundene Spalte	1

```
' Liste mit 1 = "Herr", 2 = "Frau", 3 = "Familie", ...  
Let Me.cmbAnrede.Value = 1 ' Anrede "Herr" auswählen  
' ...  
Let intAnrede = Me.cmbAnrede.Value ' Liefert 1,2 oder 3
```



Verwendung von Aufklapplisten

Löschen und Hinzufügen von Einträgen

– Generelle Syntax

```
' addItem-Funktion erwartet Eintrag als String  
' passend zu Spalten und optionale Indexposition  
Me.<cmbBez>.AddItem(<strEintrag>, <intIndex>)  
Me.<cmbBez>.RemoveItem(<intIndex>)
```

– Beispiel

```
' Aufklappliste hat zwei Spalten (Nr und Text)  
' Eintrag "Sonstiges" wird als erster Eintrag  
' hinzugefügt; Anführungszeichen beachten  
Call Me.cmbAnrede.AddItem("0;" "Sonstige"", 0)  
' Eintrag an Indexpos 3 wird aus Liste gelöscht  
Call Me.cmbAnrede.RemoveItem(3)
```

– Hinweis: Länge des Wertes für Eigenschaft Datensatzherkunft (RowSource) ist auf ca. 32.000 Zeichen begrenzt.



Verwendung von Aufklapplisten

Relevante Ereignisse einer Liste verarbeiten

- Bei Änderung: Reagieren auf Veränderung der Auswahl

```
Private Sub cmbAnrede_Change()  
    ' ...  
End Sub
```

- Bei Klicken: wie "Bei Änderung"

```
Private Sub cmbAnrede_Click()  
    ' ...  
End Sub
```

- Weitere: Feld verlassen bzw. betreten, Erhalt bzw. Verlust der Eingabemarkierung wie bei Textfeldern



Verwendung von Auswahllisten

Werte ermitteln insbesondere bei Mehrfachauswahl

- Schleife über alle Elemente der Liste
- für jedes Element der Liste prüfen, ob es ausgewählt wurde
- wenn ja, dann Wert aus "gebundener" Spalte auslesen

Generelle Syntax

- Anzahl der Elemente ermitteln

```
Me.<lstBez>.ListCount ' Liefert Anzahl Elemente
```

- Prüfen, ob Element an bestimmter Position gewählt wurde

```
' Liefert Wahrheitswert
```

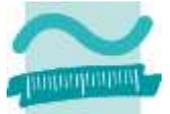
```
Let <bolVar> = Me.<lstBez>.Selected(<intPos>)
```

- Lesen eines Wertes aus einer "gebundenen" Spalte

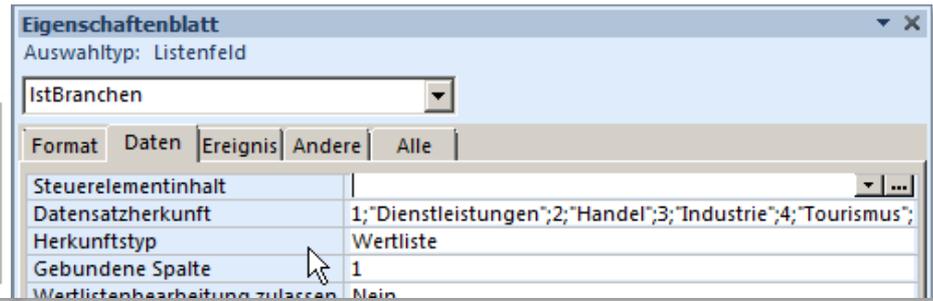
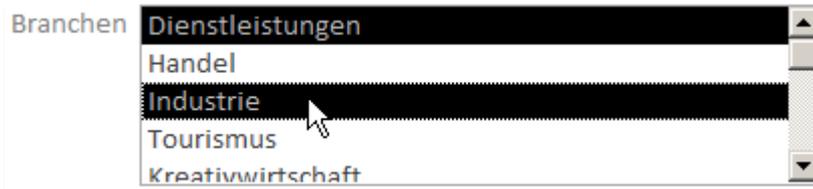
```
' Liefert Wert aus einer bestimmten Spalte der Liste,  
' der sich in einer bestimmten Zeile befindet
```

```
Let <Var> = Me.<lstBez>.Column(<intSpalte>, <intZeile>)
```

Verwendung von Auswahllisten (Fortsetzung)



Beispiel



```
Dim i As Integer ' Schleifenvariable
Dim strAuswahl As String ' String zur Ausgabe in MsgBox
Let strAuswahl = "" ' Initialisierung

' Schleife über alle Elemente der Liste
For i = 0 To Me.lstBranchen.ListCount
    ' Wenn Element an Position i ausgewählt wurde...
    If Me.lstBranchen.Selected(i) Then
        ' ... dann Wert des Elementes aus (gebundener) Spalte merken
        Let strAuswahl = strAuswahl & Me.lstBranchen.Column(0, i) & _
            ":" & Me.lstBranchen.Column(1, i) & vbCrLf
    End If
Next
' Ausgabe aller gewählten Element in MessageBox
Call MsgBox(strAuswahl)
```



Verwendung von Auswahllisten (Fortsetzung)



Relevante Ereignisse einer Liste verarbeiten

- Bei Klick: Auf Änderung der getroffenen Auswahl reagieren (bei Änderung gibt es bei Auswahllisten nicht)

```
Private Sub lstBranchen_Click()  
    ' ...  
End Sub
```

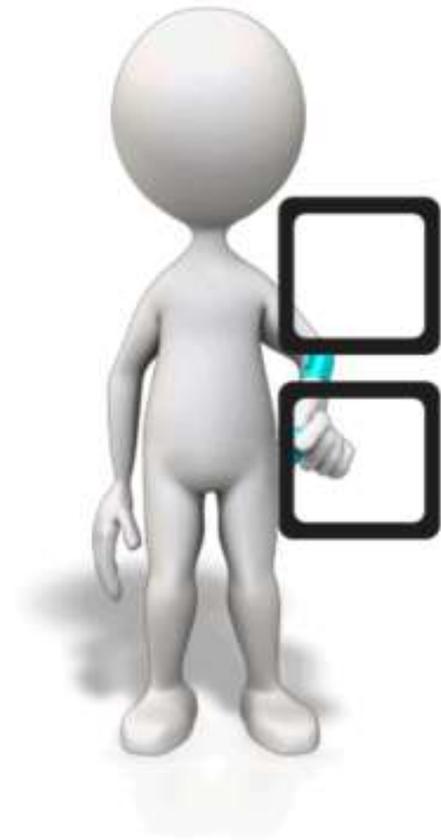
- Weitere: Feld verlassen bzw. betreten, Erhalt bzw. Verlust der Eingabemarkierung



Reaktion auf Benutzeraktionen

Abhängig von Oberflächenelement, z.B.

- Textfelder (einzeilig/mehrzeilig)
- Aufklapplisten und Auswahllisten
- Radioknöpfe und Checkboxes
- Fenster





Verwendung von Radioknöpfen

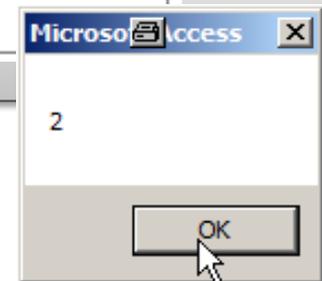
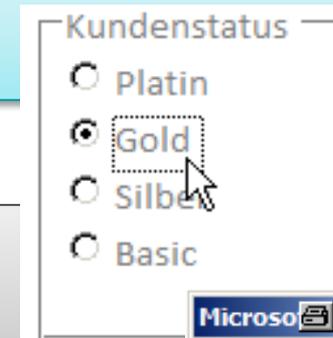
Wert des ausgewählten Radioknopfes durch Abfragen der Gruppe, in der sich der Knopf befindet

– Generelle Syntax

```
' Liefert den Wert des Radioknopfes der in der Gruppe  
' aktuelle ausgewählt ist  
Let <Var> = Me.<grpBezeichner>.Value
```

– Beispiel

```
Private Sub grpKundenstatus_Click()  
  MsgBox (Me.grpKundenstatus.Value)  
End Sub
```



Relevante Ereignisse

- Wie Auswahllisten und Textfelder: Bei Klick, Feld verlassen bzw. betreten, Erhalt bzw. Verlust der Eingabemarkierung



Verwendung von Checkboxes

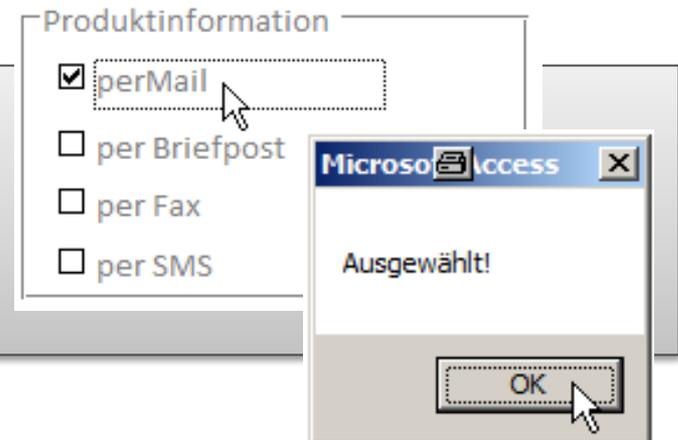
Wert einer Gruppe von Checkboxes durch Abfrage jeder einzelnen Checkbox ermitteln

- Generelle Syntax

```
' Liefert -1 wenn Checkbox ausgewählt ist  
Let <intVar> = Me.<chkBezeichner>.Value
```

- Beispiel

```
Private Sub chkMail_Click()  
  If (Me.chkMail.Value = -1) Then  
    MsgBox ("Ausgewählt!")  
  End If  
End Sub
```



Relevante Ereignisse

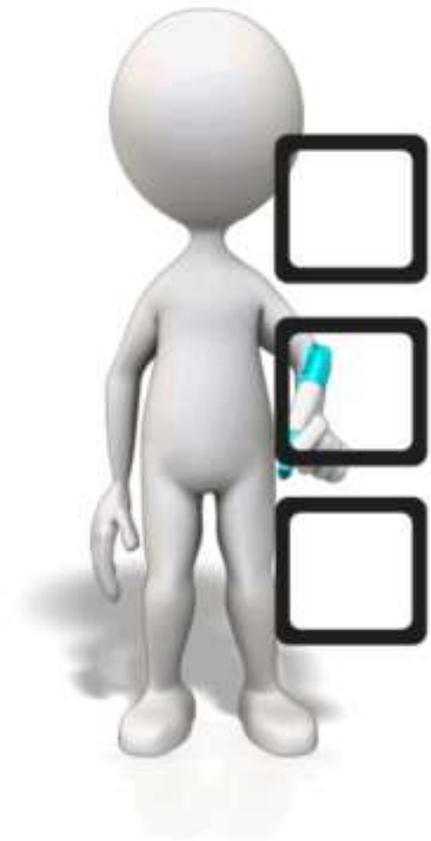
- Wie Auswahllisten und Textfelder: Bei Klick, Feld verlassen bzw. betreten, Erhalt bzw. Verlust der Eingabemarkierung



Reaktion auf Benutzeraktionen

Abhängig von Oberflächenelement, z.B.

- Textfelder (einzeilig/mehrzeilig)
- Aufklapplisten und Auswahllisten
- Radioknöpfe und Checkboxes
- Fenster





Verwendung von Fenstern

Relevante Ereignisse

- Öffnen des Fensters, z.B.
 - Initialisieren von Feldern mit berechneten Werten
 - Einblenden/Ausblenden von Oberflächenelementen
 - mit Möglichkeit zum Abbruch durch Setzen des per Referenz übergebenen Parameters Cancel auf vbYes

```
Private Sub Form_Open(Cancel As Integer)
    ' ...
End Sub
```

- Schließen des Fensters, z.B. mit
 - Prüfung von übergreifenden Plausibilitätsregeln
 - Sicherheitsabfrage, ob wirklich geschlossen werden soll

```
Private Sub Form_Close()
    ' ...
End Sub
```



Verwendung von Fenstern

Relevante Ereignisse

- Aktivieren des Fensters (Fenster wird neu geöffnet oder geöffnetes Fenster in den Vordergrund geholt)

```
Private Sub Form_Activate()  
    ...  
End Sub
```

- Deaktivieren des Fensters (anderes Fenster wird in den Vordergrund geholt, aktuelles Fenster in den Hintergrund oder wird geschlossen)

```
Private Sub Form_Deactivate()  
    ...  
End Sub
```



Verwendung von Fenstern

Relevante Ereignisse

- Laden der Inhalte des Fensters (z.B. aus Datenbank)

```
Private Sub Form_Load()  
    ...  
End Sub
```

- Entladen der Inhalte des Fensters (z.B. in Datenbank speichern und Sperren freigeben)

```
Private Sub Form_Unload()  
    ...  
End Sub
```



Verwendung von Fenstern

Reihenfolge relevanter Ereignisse

- Formular öffnen
 - → Bei Öffnen → Beim Laden → Bei Aktivierung
- Formular schließen
 - → Beim Entladen → Bei Deaktivierung → Beim Schließen
- Formular in den Hintergrund bringen
 - → Bei Deaktivierung
- Formular in den Vordergrund holen
 - → Bei Aktivierung



Reaktion auf Benutzeraktionen

Abhängig von Oberflächenelement, z.B.

- Textfelder (einzeilig/mehrzeilig)
- Aufklapplisten und Auswahllisten
- Radioknöpfe und Checkboxes
- Fenster



Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick





Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

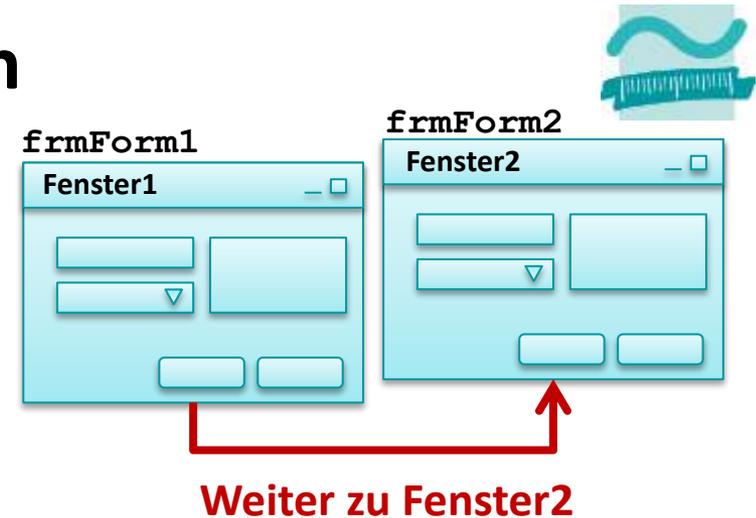
- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick

Reaktion auf Benutzeraktionen

Ereignisprozeduren bieten Zugriffsmöglichkeit auf die Oberflächenelemente und dienen zum

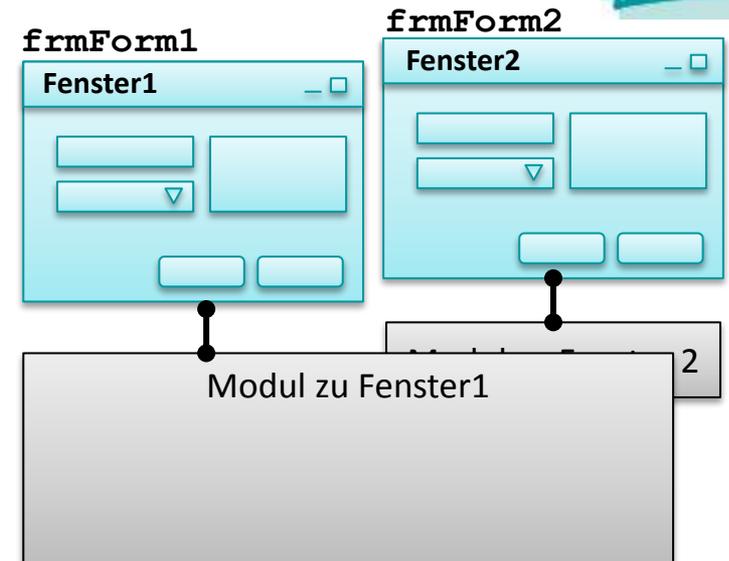
- Steuern der Elemente auf der Oberfläche
 - Aktivieren/Dekativieren bzw. Einblenden/Ausblenden von Elementen
 - Navigation zwischen Fenstern
 - ...
- Aufruf der Verarbeitungslogik
 - Übergabe der eingegebenen Daten zur Verarbeitung
 - Ermitteln der anzuzeigenden Daten
 - Ausführen von komplexen Berechnungen
 - ...



Reaktion auf Benutzeraktionen

Ereignisprozeduren bieten Zugriffsmöglichkeit auf die Oberflächenelemente und dienen zum

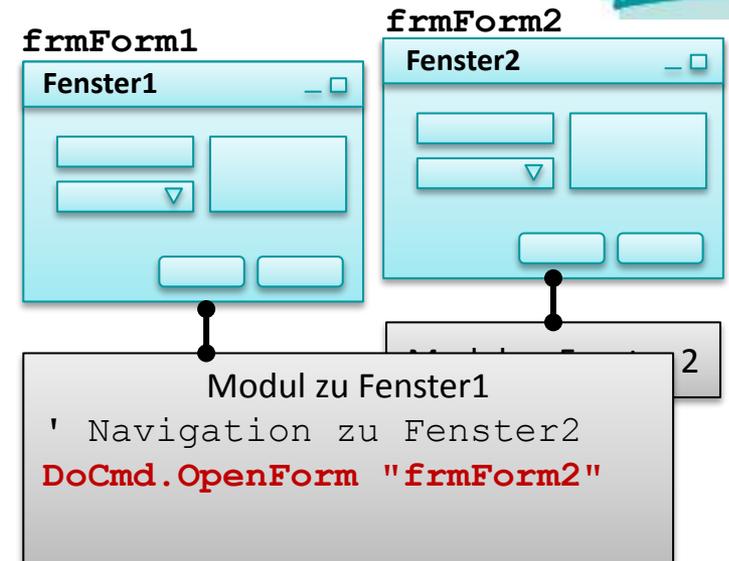
- Steuern der Elemente auf der Oberfläche
 - Aktivieren/Dekativieren bzw. Einblenden/Ausblenden von Elementen
 - Navigation zwischen Fenstern
 - ...
- Aufruf der Verarbeitungslogik
 - Übergabe der eingegebenen Daten zur Verarbeitung
 - Ermitteln der anzuzeigenden Daten
 - Ausführen von komplexen Berechnungen
 - ...



Reaktion auf Benutzeraktionen

Ereignisprozeduren bieten Zugriffsmöglichkeit auf die Oberflächenelemente und dienen zum

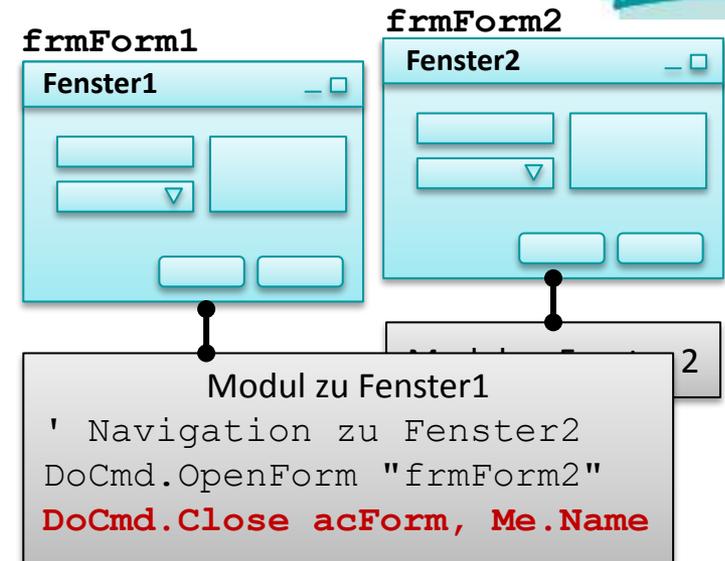
- Steuern der Elemente auf der Oberfläche
 - Aktivieren/Dekativieren bzw. Einblenden/Ausblenden von Elementen
 - Navigation zwischen Fenstern
 - ...
- Aufruf der Verarbeitungslogik
 - Übergabe der eingegebenen Daten zur Verarbeitung
 - Ermitteln der anzuzeigenden Daten
 - Ausführen von komplexen Berechnungen
 - ...



Reaktion auf Benutzeraktionen

Ereignisprozeduren bieten Zugriffsmöglichkeit auf die Oberflächenelemente und dienen zum

- Steuern der Elemente auf der Oberfläche
 - Aktivieren/Dekativieren bzw. Einblenden/Ausblenden von Elementen
 - Navigation zwischen Fenstern
 - ...
- Aufruf der Verarbeitungslogik
 - Übergabe der eingegebenen Daten zur Verarbeitung
 - Ermitteln der anzuzeigenden Daten
 - Ausführen von komplexen Berechnungen
 - ...

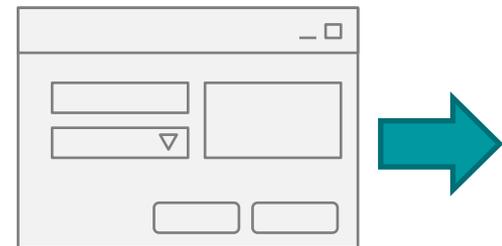
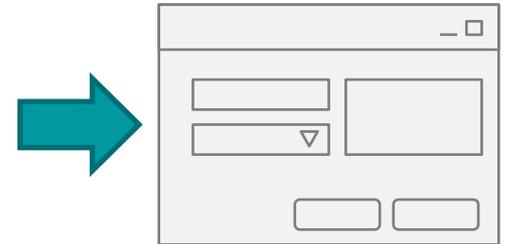


Navigation zwischen Formularen



Möglichkeiten zur Navigation zwischen Formularen

- Fenster öffnen
 - per Name öffnen
 - Parameterübergabe an Fenster
- Fenster schließen
 - aktuelles Fenster schließen
 - ein anderes Fenster schließen
- Von einem Fenster zum nächsten navigieren





Navigation zwischen Fenstern

Generelle Syntax zum Öffnen von Fenstern

- Kommando `DoCmd` stellt Funktionen zur Verfügung

' Generelle Syntax (Auszug)

```
DoCmd.OpenForm <Formularname>
```

```
DoCmd.OpenForm <Formularname>,,,,,<Übergabeparameter>
```

```
DoCmd.OpenForm FormName :=<FrmName>, OpenArgs :=<ÜbParam>
```

Beispiel

- Öffnen eines Formulars anhand des Namens

' Beispiele

```
DoCmd.OpenForm "frmStartseite"
```

```
DoCmd.OpenForm "frmKunden"
```

- Öffnen eines Formulars mit Übergabe von Parametern

' Beispiele

```
DoCmd.OpenForm "frmStart",,,,,,"Meier, Tom"
```

```
DoCmd.OpenForm FormName:="frmKnd", OpenArgs:="Knd4711"
```



Navigation zwischen Fenstern

Generelle Syntax zum Zugriff auf Übergabeparameter

- Referenzvariable **Me** stellt Eigenschaft zur Verfügung
- Nutzung innerhalb von Zuweisung sinnvoll

```
' Generelle Syntax  
Let <Var> = Me.OpenArgs
```

Beispiele

```
' Beispiele  
Let Me.txtGrusszeile.value = Me.OpenArgs  
Let strKundenId = Me.OpenArgs
```

Navigation zwischen Fenstern: Beispiel 08.06



Ziel

- Per Schaltfläche neues Fenster öffnen

Aufgabe

- Formular1
 - mit Textfeld "Name" und
 - Schaltfläche "Weiter"
- Formular2 mit Textfeld "Begrüßung"
- Bei Klick auf Weiter
 - soll Formular2 geöffnet werden
 - im Textfeld soll "Hallo" und der Name angezeigt werden, der in Fenster 1 eingegeben wurde

Formular1

Name Mike

Weiter

Formular2

Begrüßung Hallo Mike!

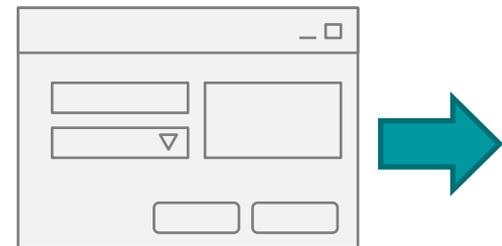
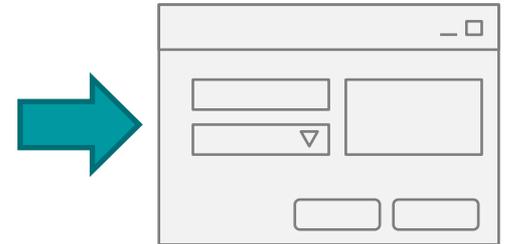


Navigation zwischen Formularen



Möglichkeiten zur Navigation zwischen Formularen

- Fenster öffnen
 - per Name öffnen
 - Parameterübergabe an Fenster
- Fenster schließen
 - aktuelles Fenster schließen
 - ein anderes Fenster schließen
- Von einem Fenster zum nächsten navigieren





Navigation zwischen Fenstern

Generelle Syntax zum Schließen von Fenstern

- Kommando **DoCmd** stellt Funktionen zur Verfügung

```
' Generelle Syntax (Auszug)  
DoCmd.Close <TypZuSchließendesObjekt>, <Name>  
' Syntax zum Schließen von Formularen  
DoCmd.Close acForm, <Formularname>
```

Beispiel

- Schließen eines Formulars anhand des Namens

```
' Beispiele  
DoCmd.Close acForm, "frmStartseite"  
DoCmd.Close acForm, "frmKunden"
```

- Schließen des aktuellen Formulars

```
' Beispiel  
DoCmd.Close acForm, Me.Name
```

Navigation zwischen Fenstern: Beispiel 08.07



Ziel

- Schließen von Fenstern

Aufgabe

- Formular1 unverändert
- Formular2 soll
 - eine Zurück- und Schließen-Schaltfläche bekommen
- unverändert bei Klick auf Weiter
- Beim Klick auf Schließen Formular2 und Formular1 schließen
- Bei Klick auf Zurück zunächst noch nichts

Formular1

Name

Formular2

Begrüßung

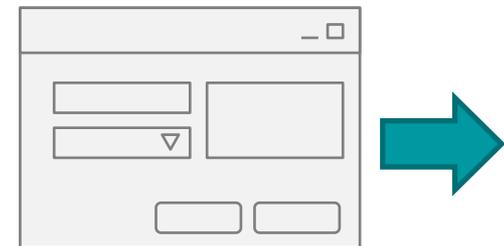
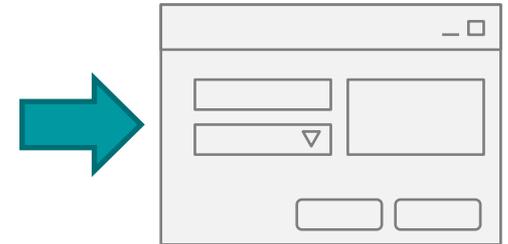


Navigation zwischen Formularen



Möglichkeiten zur Navigation zwischen Formularen

- Fenster öffnen
 - per Name öffnen
 - Parameterübergabe an Fenster
- Fenster schließen
 - aktuelles Fenster schließen
 - ein anderes Fenster schließen
- Von einem Fenster zum nächsten navigieren





Navigation zwischen Fenstern

Generelle Syntax zum Navigieren zwischen Fenstern

- Kommando `DoCmd` stellt Funktionen zur Verfügung

```
' Generelle Syntax (Auszug)
DoCmd.BrowseTo <TypZielObjekt>, <Name>
' Syntax zum Schließen von Formularen
DoCmd.BrowseTo acBrowseToForm, <Formularname>
```

Beispiel

- Navigieren zu einem Formulars anhand des Namens

```
' Beispiele
DoCmd.BrowseTo acBrowseToForm, "frmStartseite"
DoCmd.BrowseTo acBrowseToForm, "frmKunden"
```

- Alternative Schließen des aktuellen und öffnen des neuen F.

```
' Beispiel
DoCmd.Close acForm, Me.Name
DoCmd.OpenForm "frmKunden"
```

Navigation zwischen Fenstern: Beispiel 08.08



Ziel

- Navigation zwischen Fenstern per `BrowseTo` oder durch Schließen und Öffnen

Aufgabe

- Formular1 unverändert
- Formular2 unverändert
- Bei Klick auf Zurück von Formular 2 zu Formular 1 navigieren

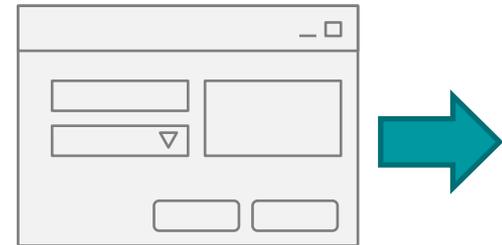
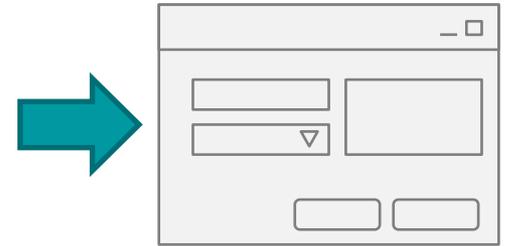


Navigation zwischen Formularen



Möglichkeiten zur Navigation zwischen Formularen

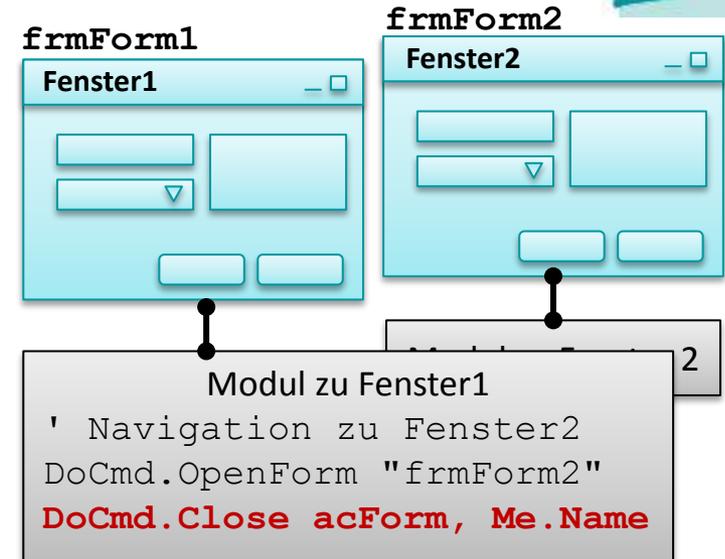
- Fenster öffnen
 - per Name öffnen
 - Parameterübergabe an Fenster
- Fenster schließen
 - aktuelles Fenster schließen
 - ein anderes Fenster schließen
- Von einem Fenster zum nächsten navigieren



Reaktion auf Benutzeraktionen

Ereignisprozeduren bieten Zugriffsmöglichkeit auf die Oberflächenelemente und dienen zum

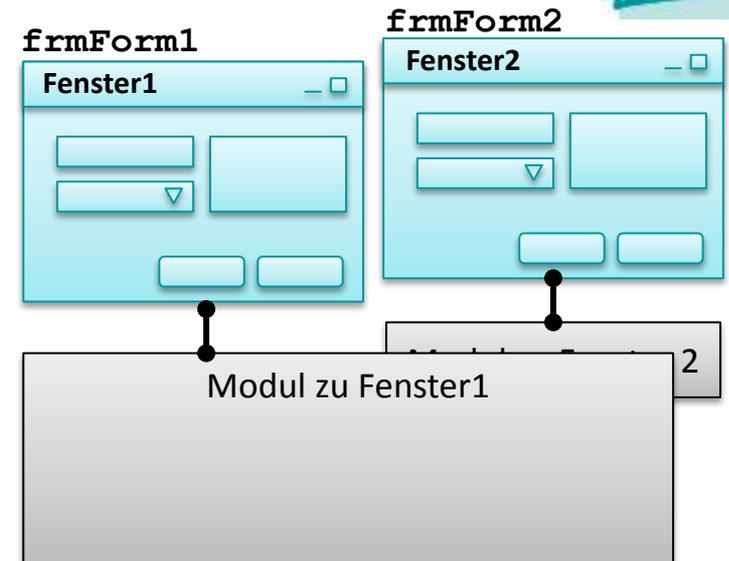
- Steuern der Elemente auf der Oberfläche
 - Aktivieren/Dekativieren bzw. Einblenden/Ausblenden von Elementen
 - Navigation zwischen Fenstern
 - ...
- Aufruf der Verarbeitungslogik
 - Übergabe der eingegebenen Daten zur Verarbeitung
 - Ermitteln der anzuzeigenden Daten
 - Ausführen von komplexen Berechnungen
 - ...



Reaktion auf Benutzeraktionen

Ereignisprozeduren bieten Zugriffsmöglichkeit auf die Oberflächenelemente und dienen zum

- Steuern der Elemente auf der Oberfläche
 - Aktivieren/Dekativieren bzw. Einblenden/Ausblenden von Elementen
 - Navigation zwischen Fenstern
 - ...
- Aufruf der Verarbeitungslogik
 - Übergabe der eingegebenen Daten zur Verarbeitung
 - Ermitteln der anzuzeigenden Daten
 - Ausführen von komplexen Berechnungen
 - ...



Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick





Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

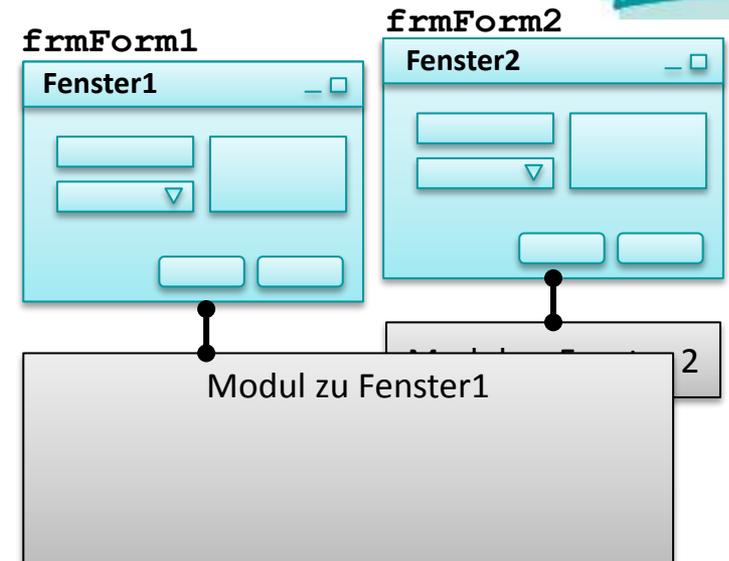
- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick

Aufruf von Verarbeitungslogik

Ereignisprozeduren bieten Zugriffsmöglichkeit auf die Oberflächenelemente und dienen zum

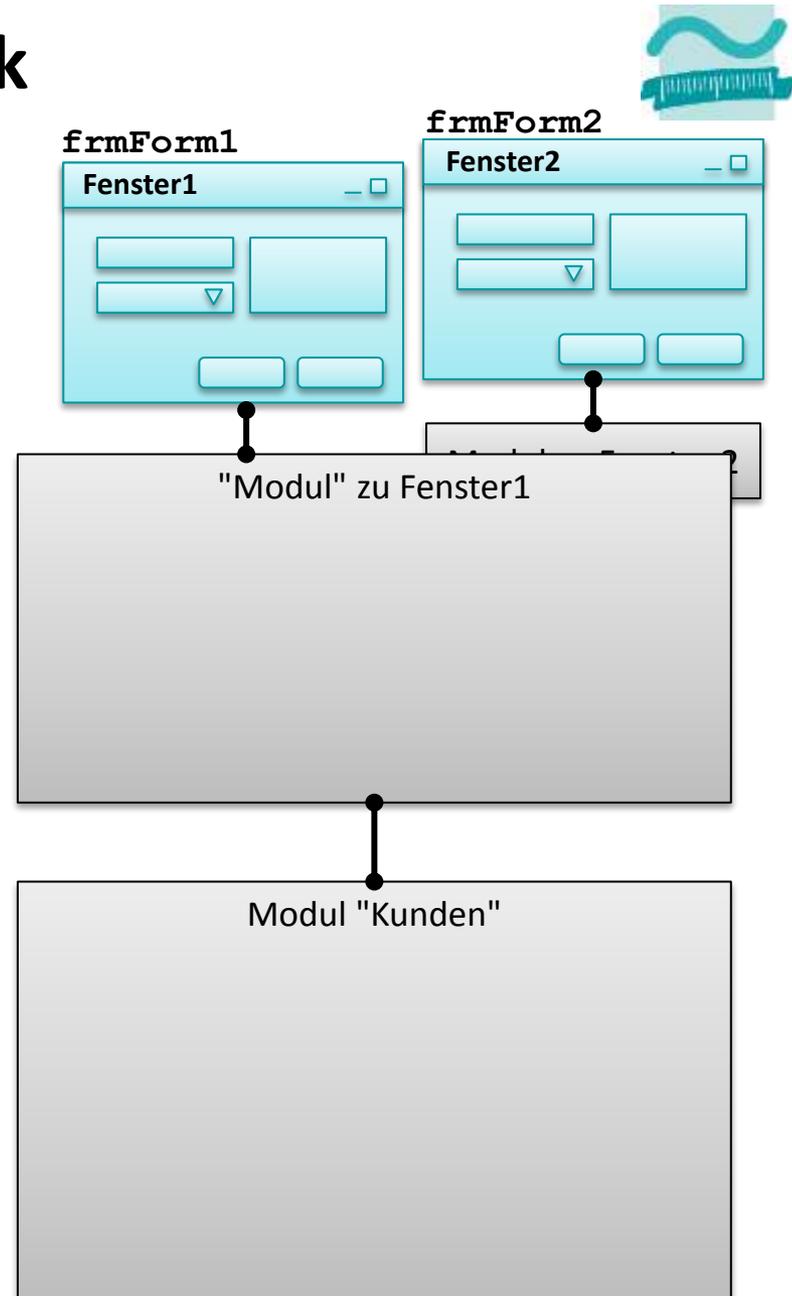
- Steuern der Elemente auf der Oberfläche
 - Aktivieren/Dekativieren bzw. Einblenden/Ausblenden von Elementen
 - Navigation zwischen Fenstern
 - ...
- Aufruf der Verarbeitungslogik
 - Übergabe der eingegebenen Daten zur Verarbeitung
 - Ermitteln der anzuzeigenden Daten
 - Ausführen von komplexen Berechnungen
 - ...



Aufruf von Verarbeitungslogik

Ereignisprozeduren bieten Zugriffsmöglichkeit auf die Oberflächenelemente und dienen zum

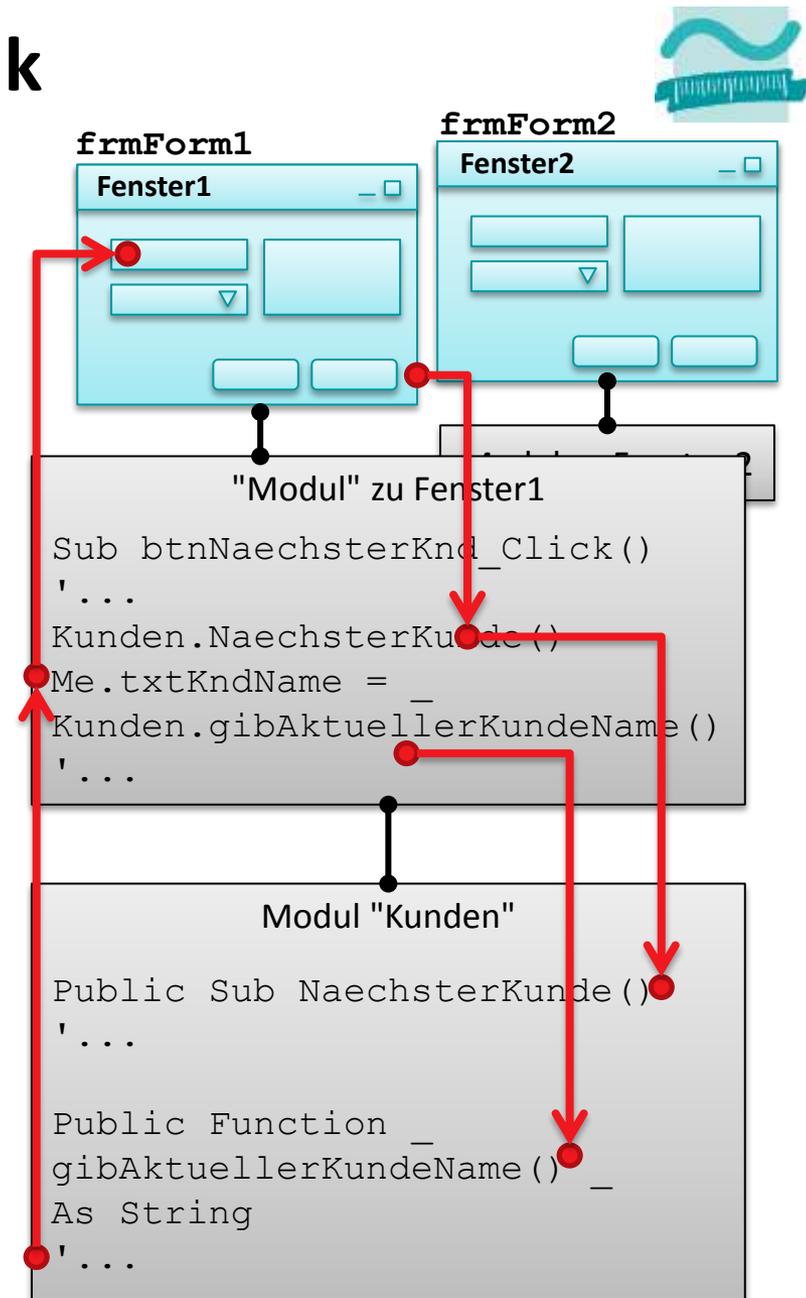
- Steuern der Elemente auf der Oberfläche
 - Aktivieren/Dekativieren bzw. Einblenden/Ausblenden von Elementen
 - Navigation zwischen Fenstern
 - ...
- Aufruf der Verarbeitungslogik
 - Übergabe der eingegebenen Daten zur Verarbeitung
 - Ermitteln der anzuzeigenden Daten
 - Ausführen von komplexen Berechnungen
 - ...



Aufruf von Verarbeitungslogik

Ereignisprozeduren bieten Zugriffsmöglichkeit auf die Oberflächenelemente und dienen zum

- Steuern der Elemente auf der Oberfläche
 - Aktivieren/Dekativieren bzw. Einblenden/Ausblenden von Elementen
 - Navigation zwischen Fenstern
 - ...
- Aufruf der Verarbeitungslogik
 - Übergabe der eingegebenen Daten zur Verarbeitung
 - Ermitteln der anzuzeigenden Daten
 - Ausführen von komplexen Berechnungen
 - ...



Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick





Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick



Inhalt

Einordnung

Rückblick

Ausgangspunkt

Teil 1: Oberflächenelemente und ihre Eigenschaften

- Arten von Elementen und ihre Verwendung
- Eigenschaften von Elementen
- Grundlegende Gestaltung

Teil 2: Oberflächen und Ereignisse programmieren

- Ereignisse als Reaktion auf Benutzeraktionen
- Navigation zwischen Fenstern
- Aufruf der Verarbeitungslogik

Abschluss und Ausblick

Abschluss



Wichtige Oberflächenelemente

- Eingabe und Auswahl
 - Textfelder
 - Aufklappliste/Kombinationsfeld
 - Mehrfachauswahllisten
 - Radioknöpfe (Optionsfeld)
 - Kontrollkästchen (Checkbox)
- Aktionselemente
 - Schaltfläche (einfach)
 - Umschaltfläche (Toggle)
- Container
 - Rahmen/Gruppen
 - Registerkartensatz mit Registerkarten
 - Fenster/Dialoge (in Access als Formulare)

Feld

Mehrzeiliges Feld

Kombinationslistenfeld

- Alternative 1
- Alternative 2
- Alternative 3
- Alternative 4

Radioknöpfe

- Alternative 1
- Alternative 2
- Alternative 3

Möglichkeiten

- Möglichkeit 1
- Möglichkeit 2
- Möglichkeit 3

Mehrfachauswahlliste

Möglichkeit 1	12,50 €
Möglichkeit 2	24,80 €
Möglichkeit 3	37,50 €

Anschrift

Straße Nr.

PLZ Ort

Namen Adressen

Straße Nr.

PLZ Ort

Formular1

Datensatz: 1 von 1 | Kein Filter | Suchen

Abschluss



Referenzvariable Me stellt Funktionen zur Verfügung

- Zugriff auf den Wert von Feldern liefert immer String

' **Generelle Syntax**

```
Let <VarString> = Me.<BezeichnerDesFeldes>.Value
```

```
Let <VarZahl> = Val(Me.<BezeichnerDesFeldes>.Value)
```

- Genereller Zugriff auf Eigenschaften von Elementen

' **Generelle Syntax**

```
Let <Var> = Me.<Bez>.<Eigenschaft> ' Lesen
```

```
Let Me.<Bez>.<Eigenschaft> = <Var> ' Schreiben/Ändern
```

Beispiele

' **Lesen von Werten**

```
Let strName = Me.txtName.Value
```

```
Let intAlter = Val(Me.txtAlter.Value)
```

```
Let bolBrillentraeger = CBool(Me.chkBrille.Value)
```

' **Ändern von Eigenschaften**

```
Let Me.txtName.Visible = False
```

Abschluss



Kommando DoCmd stellt Funktionen zur Verfügung

- Generelle Syntax zum Öffnen von Fenstern

```
' Generelle Syntax (Auszug)  
DoCmd.OpenForm <Formularname>
```

- Generelle Syntax zum Schließen von Fenstern

```
' Generelle Syntax (Auszug)  
DoCmd.Close <TypZuSchließendesObjekt>, <Name>  
' Syntax zum Schließen von Formularen  
DoCmd.Close acForm, <Formularname>
```

- Generelle Syntax zum Navigieren zwischen Fenstern

```
' Generelle Syntax (Auszug)  
DoCmd.BrowseTo <TypZielObjekt>, <Name>  
' Syntax zum Schließen von Formularen  
DoCmd.BrowseTo acBrowseToForm, <Formularname>
```

Ausblick



06 – Zusammenfassung

05 – Debugger und Testen

04 – Fortgeschrittene Konzepte

04.A
Oberflächen (Teil 1)
Elemente und
Eigenschaften

04.B
Oberflächen (Teil 2)
Ereignisverarbeitung

04.C
Zugriff auf
Dateisystem und
Anwendungen

03 – Grundkonzepte

03.A
Wert
Ausdruck
Variable
Konstante
Datentyp

03.B
Bedingte
Ausführung/
Verzwei-
gungen

03.C
Schleifen

03.D
Felder
Mengen

03.E
Prozedur
Funktion
Modul

02 – Grundlagen der Programmierung

01 – Grundbegriffe der Wirtschaftsinformatik



Literatur



[Balzert, 1996] H. Balzert: Lehrbuch der Softwaretechnik: Softwareentwicklung. Spektrum akad. Verlag, Heidelberg u.a. (1996).

Quellen



[1] Windows User Experience Guidelines for Windows XP and Windows 2000.
<http://www.microsoft.com/en-us/download/details.aspx?id=4249>

[2] Design and User Experience Library.
http://www.developer.nokia.com/Resources/Library/Design_and_UX/#!/index.html

[3] Designing for Windows Phone.
http://www.developer.nokia.com/Resources/Library/Design_and_UX/#!/designing-for-windows-phone.html

[4] Nokia Asha Design Guidelines.
http://www.developer.nokia.com/Resources/Library/Asha_UI/#!/index.html

[5] iOS Developer Library, iOS Human Interface Guidelines.
https://developer.apple.com/library/ios/#documentation/userexperience/conceptual/mobilehig/UIElementGuidelines/UIElementGuidelines.html#//apple_ref/doc/uid/TP40006556-CH13-SW1

[6] Android User Interface Guides. <http://developer.android.com/guide/topics/ui/index.html>

[7] Mac Developer Library Developer, OS X Human Interface Guidelines.
http://developer.apple.com/library/mac/#documentation/userexperience/conceptual/applehiguidelines/UEGuidelines/UEGuidelines.html#//apple_ref/doc/uid/TP40002720-TPXREF101

[8] Windows User Experience Interaction Guidelines for Windows® 7 and Windows Vista®.
<http://msdn.microsoft.com/en-us/library/windows/desktop/aa511258.aspx>

[9] iOS 7 UI Transition Guide.
<https://developer.apple.com/library/etc/redirect/WWDR/iOSUITransitionGuide>

[10] iOS Human Interface Guidelines
<https://developer.apple.com/library/etc/redirect/WWDR/iOSHIG>



BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Wirtschaftsinformatik 1

LE 08 – Oberflächen und Ereignisse

Prof. Dr. Thomas Off

<http://www.ThomasOff.de/lehre/beuth/wi1>