Inhaltsverzeichnis

1.	Der schnelle Weg zum ersten Programm	16
	Was bietet Visual Basic 6?	16
	Neue Projekte anlegen	17
	Die Entwicklungsumgebung	18
	Formulare und Steuerelemente einsetzen	19
	Steuerelemente auf einer Form plazieren	19
	Eigenschaften anpassen	21
	Das Steuerelement Befehlsschaltfläche	23
	Formgröße und Formicon verändern	24
	Namensvergabe für Steuerelemente	25
	Projekte speichern	27
	Probelauf starten	27
	Ersten Programmtext erstellen	28
	Ereignisprozeduren anlegen	29
	Eigenschaften zur Laufzeit abfragen und festlegen	30
	Der richtige Umgang mit Variablen	32
	Den richtigen Datentyp auswählen	34
	Typkennzeichen für den schnellen Zugriff	36
	Fehlervermeidung im Vorfeld	37
	Ausführbare Programmdatei erzeugen	39
	Programmtext-Editor-Einstellungen	39
	Anpassung der Entwicklungsumgebung	40
	Check it!	42
2.	Die einzelnen Steuerelemente im Detail	43
	Bestehende Projekte öffnen	43
	Das Steuerelement Rahmen	44
	Schriften und Farben einsetzen	45
	Das Steuerelement Bildlaufleiste	46
	Optionsfelder – Eindeutige Auswahlentscheidungen	47
	Bildfelder – Grafiken anzeigen	47
	Das Steuerelement Listenfeld	48
		.0

Grafische Schaltflächen einrichten Layouts optimieren	49 50
lf-Then-Else – Entscheidungen im Programmablauf	51
Hilfe per MSDN – 1,1 GByte Informationen	53
Listenfeldeinträge hinzufügen	54
Synchronisation von Listenfeldern	55
Listenfeldeinträge löschen	56
Ereignisse im Überblick	57
Das Ereignis Doppelklick	59
If-Then-Elself – Komplexe Entscheidungen	61
Check it!	64

3.	Weitere Grundbegriffe schnell erklärt	66
	Steuerelementfelder sparen Zeit und Arbeit	66
	Tastenerkennung und Eingaben filtern	68
	Parameterlisten von Ereignisprozeduren	69
	Verknüpfungen logischer Ausdrücke	70
	Steuerelemente aktivieren – Den Fokus setzen	71
	Der Select Case-Befehl	72
	Fenstertypen festlegen	75
	Eine Gedenkminute für den Anwender	76
	Feldvariablen verkürzen den Arbeitsaufwand	77
	Mehrdimensionale Felder	79
	Eigene Datentypen erstellen	80
	Moduldateien zur weiteren Verwendung einbinden	82
	Moduldateien hinzufügen	82
	Konstanten – Reduzierte Variablen	85
	Allgemeine Prozeduren und Funktionen	86
	Das Ereignis Programmstart	88
	Der Unterschied zwischen Prozedur und Funktion	88
	Parametrierung von Prozeduren und Funktionen	89
	Schleifen – Die For-Next-Anweisung	91
	Verschachtelung von If-Then-Entscheidungen	93
	Gültigkeitsbereiche von Variablen	97
	Globale Variablen	97
	Das Ereignis Programmende	100
	Check it!	101

4.	Menüs, Grafik und mathematische Operationen	102
	Projekte mit mehreren Formularen erstellen	102
	Formen hinzufügen	103
	Formen entfernen	104
	Vorhandene Formen laden	104
	Das Steuerelement Kontrollfeld	105
	Das Steuerelement Kombinationslistenfeld	105
	Es ist serviert – Menüs erstellen	106
	Menüs anlegen	107
	Menüpunkte erstellen	108
	Die Fensterposition festlegen	111
	Formen laden und anzeigen – Show oder Load?	112
	Modale Anzeige einer Form	114
	Mathematische Operationen und Grafikausgabe	114
	Das Steuerelement Figur	115
	Das Steuerelement Linie	115
	Direkte Grafikanweisungen im Programmtext	116
	Alles Paletti – Das Farbmodell RGB	116
	Reaktion auf Fenstergrößenveränderungen	117
	Formatierung und Kommentierung	119
	Formgrößenveränderung – Das Ereignis Resize	120
	Steuerelemente auf anderen Formen ansprechen	121
	Zahlenformatierung – Die Formatfunktion	121
	Maßsysteme und Maßeinheiten	122
	AutoRedraw & Paint – Fensterneuzeichnung	128
	Die InputBox einsetzen	131
	Eigene Kontextmenüs erzeugen	133
	Menüs dynamisch erweitern	134
	Check it!	137
5.	Debuggen – Aus Fehlern werden Sie lernen	138
	Fehlereliminierung während der Eingabe	138
	Laufzeitfehler abfangen – Variablenwerte erkunden	140
	On Error Goto – Intelligente Fehlerbehandlung	142
	Detaillierte Informationen mit dem Fehler-Objekt	143
	Resume und Resume Next	144

Resume und Resume Next144Die MessageBox – Infos an den Anwender144Rückgabewerte der MessageBox145

Logische Fehler erkennen	
Variablenwerte im Direktfenster begutachten	147
Fallbeispiel zur Fehlersuche	148
Haltestellen – Programmunterbrechung	149
Die Symbolleiste Testen	151
Schrittweiser Programmablauf	152
Überwachungen kreisen den Fehler ein	153
Die Symbolleiste Bearbeiten	156
Lesezeichen setzen	157
Check it!	158

6.	Dateizugriffe und Bildbetrachtung – Diashow	160
	Das Anzeigefeld – Dynamische Grafikgrößen	160
	Stretching – Größenanpassung von Grafiken	161
	Zugriff auf Laufwerke, Ordner und Dateien	161
	Grafiken zur Laufzeit anzeigen	164
	Zeichenkettenfunktionen benutzen	165
	Steuerelemente unsichtbar machen	166
	Standarddialoge einsetzen	168
	Zusatzsteuerelemente einsetzen	168
	Farbdialoge zur Verfügung stellen	169
	Der Datentyp Control	170
	Den Schriftendialog anbieten	172
	Den Datei-Speichern-Dialog benutzen	174
	Den Datei-Öffnen-Dialog anbieten	177
	Dateien löschen	178
	Den Druckerdialog aufrufen	180
	Der Objektkatalog – Infos über alle Objekte	180
	Maximale Fenstergröße für Ihre Formen	182
	Auf die Sekunde genau – Mit Zeitgebern arbeiten	185
	Formulare auf Monitorauflösung anpassen	187
	Steuerelemente zur Laufzeit verschieben	187
	Check it!	189

7.	Drag & Drop – Dateiverwaltung	190
	Dateiverwaltung per Drag & Drop	190
	Einfach und lesbar – Textdateien verwalten	196
	Textdateien öffnen, lesen und schließen	197
	Die maximale Formfläche ausnutzen	199
	Textdateien schreiben	200
	Programmanweisungen zum Drucken	202
	Mehrfachauswahl in Listenfeldern ermöglichen	203
	Datums- und Zeitangaben einsetzen	206
	Datumsdifferenzen ermitteln	207
	Einen Paßwortschutz einrichten	207
	Ohne Form geht's los – Sub Main	208
	Check it!	210

8.	MDI-Formen – Visitenkarteneditor	212
	MDI-Formen erzeugen	212
	Kindformen erzeugen	213
	Steuerelementtypen ermitteln	215
	Kindformen laden und anordnen	217
	Die aktive Kindform ermitteln	219
	Visitenkarten drucken	222
	Check it!	224
	Multimedia-Programmierung mit Visual Basic	225
	Mit API-Funktionen arbeiten	220
	Die Declare-Anweisung	229
	Multimedia mit der API-Funktion MciSendString	230
	Parameterübergabe per Wert oder per Referenz?	231
	Video- und Audioausgabe	234
	MIDI-Dateien abspielen	236
	Wave-Dateien wiedergeben	237

10.	Klassenmodule verwenden – Objekte erstellen	242
	Steuerelemente dynamisch laden	242
	Standardeigenschaften von Steuerelementen	247
	Verwendung von Klassenmodulen	248
	Klassenmodule erzeugen	250
	Datenkapselung	251
	Instanzen einer Klasse erzeugen	252
	Klasseneigenschaften und Klassenmethoden	253
	Private Klasseneigenschaften ansprechen	254
	Klassenmethoden erzeugen	257
	Die Initialisierung einer Klasseninstanz	258
	Objektvariablen einsetzen	260
	Der Objekttyp Object	261
	Kollektionen und dynamische Feldvariablen	263
	Dynamische Feldvariablen sparen Speicher	263
	Kollektionen	264
	Polymorphismus	268
	Freignisse ahwarten	269
	Multitasking – Rechenzeit freigeben	270
	Check it!	271
11.	ActiveX-Steuerelemente erstellen	272
	Eigene Steuerelemente entwerfen	272
	Mehrere Projekte in der Entwicklungsumgebung	276
	OCX-Datei erstellen	278
	Der ActiveX-Schnittstellen-Assistent	280
	Eigenschaftsvariablen und Eigenschaftsprozeduren	283
	Neue Ereignisse verfügbar machen	284
	Initialisierung von Benutzersteuerelementen	285
	Eigenschaftswerte speichern	286
	Eigenschaftswerte einlesen	287
	Benutzersteuerelement-Methoden erzeugen	287
	Benutzerdefinierte Eigenschaftendialoge erzeugen	287
	Anpassung von Eigenschaftenseiten	289
	Einbindung in HTML-Seiten	290
	Check it!	292

12.	Mit der Datenbank zur Terminverwaltung	293
	Zugriff auf Access & Co	293
	Das ADO-Datensteuerelement	294
	Das Datentabellen-Steuerelement	297
	Das Datentabellen-Steuerelement anpassen	298
	SQL-Abfragen	299
	Select – Selektion von Datensätzen	299
	Delete – Löschen von Datensätzen	300
	Update – Aktualisieren von Datensätzen	300
	Insert – Hinzufügen von Datensätzen	301
	Direkter SQL-Zugriff	301
	Das Objekt Recordset – Datensatzvariablen	304
	Das Kalender-Steuerelement einsetzen	308
	Check it!	312
Super	Index	313

8 MDI-Formen – Visitenkarteneditor

Viele bekannte Programme – beispielsweise Word oder Excel – bieten dem Anwender die Möglichkeit, mehrere Fenster für ein und denselben Sachverhalt, aber für verschiedene Daten in einem übergeordneten Fenster anzuzeigen. Der Anwender kann so beispielsweise mehrere Texte gleichzeitig bearbeiten. Die einzelnen Fenster unterscheiden sich nicht in ihrer Funktionalität, sondern in diesem Fall nur durch den jeweiligen Text, also die Daten. Lernen Sie in diesem Kapitel anhand eines Visitenkarten-Editors, wie Sie mehrere gleichartige Formen in einer übergeordneten Form verwalten.

MDI-Formen erzeugen

Bislang erzeugten Sie ausschließlich SDI-Programme (Single Document Interface), bei denen alle enthaltenen Formen gleichberechtigt sind. Eine übergeordnete Form zur Aufnahme mehrerer Kindformen (auch Dokumentenfenster genannt) wird dagegen MDI-Form (Multiple Document Interface) genannt. Vorteile eines solchen Mechanismus sind zum einen die Möglichkeit, beispielsweise alle beteiligten Formen durch Schließen der MDI-Form gemeinsam beenden und zum anderen die Kindformen innerhalb der Hauptform platzgerecht anordnen zu können.

1. Öffnen Sie ein neues Projekt.

Erzeugen Sie nun zunächst die MDI-Form, die alle weiteren Formen aufnehmen soll.

- 2. Wählen Sie den Menübefehl *Projekt/MDI-Form hinzufügen* oder den Befehl *Hinzufügen/MDI-Formular* des Kontextmenüs im Projekt-Explorer. Alternativ wählen Sie den Eintrag *MDI-Formular* aus der Toolbar-Liste.
- 3. Wählen Sie *MDI-Formular* auf der Registerseite *Neu*.

DI-Formular hinzufügen	? ×			
Neu Vorhanden				
5				
MDI-Formular				
	Uğnen			
	Abbrechen			
	Hile			
Dieses Dialogfeld gicht mehr anzeigen				

4 Versuchen Sie anschließend, den gleichen Befehl noch einmal auszuführen.

8

Der Befehl ist nun nicht mehr verfügbar, da ein Programm nur aus einer MDI-Form bestehen kann.

Die MDI-Form erkennen Sie nicht nur am entsprechenden Titelzeilentext, sondern auch an der dunkleren Hintergrundfarbe



gegenüber einer normalen Form. Ihr Projekt besteht nun aus zwei Formen, der ursprünglich vorhandenen Form *Form1* und der neuen Form *MDIForm1*.

- **5.** Da die MDI-Form beim Starten des Programms angezeigt werden soll, müssen Sie diese nun als Startform über den Menübefehl *Projekt/Eigenschaften* deklarieren. Verändern Sie außerdem deren Titelzeilentext zu *Multipler Visitenkarten-Editor*.

MDI-Formen dienen nur zur Aufnahme weiterer Formen. Auf MDI-Formen können keine Steuerelemente mit Ausnahme von Zeitgebern, Standarddialogen, Bildfeldern und Menüs erzeugt

Kindformen erzeugen

werden.

Da gibt es doch noch eine Form! Was machen Sie mit der schon vorhandenen Form *Form1*? Diese erklären Sie einfach zur ersten Kindform der MDI-Form.

- • SUPER STEPS • • • • • •
- **1** Vergeben Sie zunächst für die Form *Form* 1 den Namen *Kindform* und den Titelzeilentext *Visitenkarte*. Setzen Sie dann die Eigenschaft *MDIChild* dieser Form auf *True*, damit ist diese Form als Kindform deklariert.
- 2. Programmieren Sie die Anzeige der Kindform innerhalb der MDI-Form in deren Ereignisprozedur Form_Load.



- 1. Private Sub MDIForm_Load()
- 2. Kindform.Show
- 3. End Sub
- 3. Im Projekt-Explorer erkennen Sie MDI- und Kindformen übrigens leicht an den unterschiedlichen Symbolen.

Speichern Sie die Projektdateien im Ordner *KapitelO8\Leser*. Geben Sie dem Projekt den Namen *MDI*. Starten Sie dann einen Probelauf.

Sie erkennen nun eine Kindform innerhalb der MDI-Form.

4. Verschieben Sie die Kindform innerhalb der MDI-Form.

Der MDI-Form werden automatisch Bildlaufleisten hinzugefügt, wenn das Kindfenster nicht mehr ganz sichtbar ist.

- **5.** Klicken Sie auf die Minimieren-Schaltfläche der Kindform.
- 6. Klicken Sie auf die Maximieren-Schaltfläche der Kindform.



Kindformen sind, wie Sie schon richtig erkannt haben, an die zugehörige MDI-Form gebunden, daher können diese auch nicht außerhalb einer MDI-Form plaziert oder aus dieser heraus bewegt werden.

Die Steuerelemente der Visitenkartenform

Die mit diesem Programm zu erzeugenden Visitenkarten sollen prinzipiell wie oben zu sehen gestaltet werden. Pro Kindform werden also einige Bezeichnungsfelder und ein Bildfeld benötigt. Da Sie die Erzeugung von Steuerelementen schon im Schlaf beherrschen, werden Sie hier damit auch nicht weiter gelangweilt, eine entsprechende Kindform ist schon für Sie vorbereitet.

Die Form *FormKindform.frm* befindet sich im Ordner *Kapitel08\Leser*.



1.

👟 Scheint es durch? 🛛 🗖 🖾

Label2

- **1** Entfernen Sie die Form *Kindform* aus dem Projekt und fügen Sie die vorbereitete Form *FormKindform* hinzu.

2. Ändern Sie den Befehl zur Anzeige dieser neuen Form in der Ereignisprozedur *MDIForm1_Load*:

Wesentliche Eigenschaften dieser Form sind schon voreingestellt. Die Maßeinheit ist beispielsweise auf *Zentimeter* eingestellt, damit anhand der Eigenschaften *ScaleWidth* und *Scale-Height* die Formgröße kontrolliert werden kann, die auf annähernd 8.5 x 5.5 cm eingestellt ist. Als Hintergrundfarbe ist Weiß gewählt, da die Visitenkarten später auch ausgedruckt werden sollen. Der Wert der Eigenschaft *BorderStyle* ist auf *Fest Einfach* und der Wert der Eigenschaft *MinButton* auf *True* eingestellt. Die Formgröße kann demnach nicht verändert, die Form aber weiterhin auf Symbolgröße verkleinert werden.

Transparente Steuerelemente

Die auf der Form befindlichen Bezeichnungsfelder bilden ein Steuerelementfeld, deren Beschriftungstexte sind allesamt gelöscht. Die jeweiligen Eigenschaften *BackStyle* werden zur Laufzeit auf *Transparent* gesetzt. Das bewirkt, daß unter ei-

nem Steuerelement befindliche Steuerelemente komplett angezeigt und nicht verdeckt werden.

Steuerelementtypen ermitteln

Die Eigenschaft *Alignment* aller Bezeichnungsfelder der Form soll zur Laufzeit auf *Zentriert*, deren Eigenschaft *BackStyle* auf *Transparent* sowie deren Eigenschaft *AutoSize* auf *True* eingestellt werden. Ferner sollen diese am linken Rand der Form ausgerichtet und auf maximale Formbreite vergrößert werden. Benutzen Sie einmal mehr eine *For Each*-Schleife für alle Steuerelemente der Form. Sie benötigen nun aber noch Kenntnisse über die Möglichkeit, Steuerelemente anhand ihres Typs zu unterscheiden. Mit der Anweisung

If TypeOf Steuerelement Is Typ



2. FormKindform.Show 3. End Sub

A A

Private Sub MDIForm Load()



👟 Scheint es durch? 🛛 🗖 🖾

Labell

untersuchen Sie das betreffende Steuerelement auf einen speziellen Typ.

- Ergänzen Sie Zeile 4 der Ereignisprozedur Form_Load der Form FormKindform:
 - 1. Private Sub Form_Load()
 - 2. Dim Steuerelement As Control
 - 3. For Each Steuerelement In Me
 - 4. If TypeOf Steuerelement Is Label Then
 - 5. Steuerelement.Alignment = 2
 - 6. Steuerelement.BackStyle = 0
 - 7. Steuerelement.AutoSize = True
 - 8. Steuerelement.Move 0, Steuerelement.Top, ScaleWidth,
 - Steuerelement.Height
 - 9. End If
 - 10. Next
 - 11. Rahmen.Move 0, 0, ScaleWidth, ScaleHeight
 - 12. End Sub

Alle Steuerelemente der Form werden betrachtet und mit Hilfe der Funktion *TypeOf* auf den Steuerelementtyp *Label* hin untersucht. Folgende Typen existieren, hier erhalten Sie zusätzlich die entsprechenden Gegenüberstellungen der deutsch- und englischsprachigen Bezeichnungen für Steuerelemente.

Steuerelementtypen

Steuerelement	Steuerelementtyp
Anzeigefeld	Image
Befehlsschaltfläche	CommandButton
Bezeichnungsfeld	Label
Bildfeld	PictureBox
Dateilistenfeld	FileListBox
Datensteuerelement	Data
Datengebundenes Listenfeld	DBList
Datengebundenes Kombinationslistenfeld	DBCombo
Datengebundenes Gitternetz	DBGrid
Textfeld	TextBox
Figur	Shape
Horizontale Bildlaufleiste	HScrollBar
Kombinationslistenfeld	ComboBox
Kontrollfeld	CheckBox
Laufwerklistenfeld	DriveListBox
Linie	Line
Listenfeld	ListBox
Optionsfeld	OptionButton
Ordnerlistenfeld	DirListBox
Rahmen	Frame
Vertikale Bildlaufleiste	VScrollBar
Zeitgeber	Timer

Der auf der Form befindliche Rahmen wird in Zeile 11 auf die Innenabmessungen der Form eingestellt. Dies ist für den später zu behandelnden Druckvorgang relevant, damit die Visitenkarten mit einer Umrandung versehen werden können.

Kindformen laden und anordnen

Im folgenden geht es darum, dem Anwender die Möglichkeit zu bieten, mehrere Visitenkarten gleichzeitig bearbeiten zu können, was den MDI-Charakter ja ausmacht.

1 Erzeugen Sie aber zunächst folgende Menüs mit den entsprechenden Menüpunkten zur Form *MDIForm* **1**. Achten Sie dabei darauf, daß die Menüpunkte des Menüs *Fenster* ein Steuerelementfeld bilden, veregeben Sie also die entsprechenden Indizes!

Menüname	Menübeschriftung	Übergeordnetes Menü	Index
MenüFenster	&Fenster	-	-
MenüFeld	&Überlappend	MenüFenster	0
MenüFeld	&Nebeneinander	MenüFenster	1
MenüFeld	&Untereinander	MenüFenster	2
MenüFeld	&Symbole anordnen	MenüFenster	3
MenüEinstellungen	&Einstellungen	-	-
MenüDaten	&Daten	MenüEinstellungen	-
MenüVisitenkarten	&Visitenkarten	-	-
MenüNeu	Neue & Visitenkarte	MenüVisitenkarten	-
MenüDrucken	Visitenkarte &drucken	MenüVisitenkarten	-
MenüVorhanden	&Vorhandene Visitenkarten	-	-

Menüs der Form MDIForm1

8

2. Markieren Sie das Kontrollfeld *WindowList* im Menü-Editor beim Menü *MenüVorhanden*.

Beschäftigen Sie sich nun mit der Erzeugung von weiteren Kindformen, da nur eine Kindform innerhalb einer MDI-Form natürlich nicht allzuviel Sinn macht. Wie gehen Sie vor? Fügen Sie dem Projekt zur Entwicklungszeit weitere Formen hinzu und statten Sie diese jeweils mit den benötigten Steuerelementen aus? Erzeugen Sie auf diesem Weg zehn Kindformen, können Sie natürlich später auch nur zehn Kindformen und nicht mehr anzeigen. Vielleicht benötigten Sie irgendwann aber einmal 100 Kindformen. Die benötigten Kindformen unterscheiden sich doch eigentlich nur in ... ja, worin eigentlich? Sie unterscheiden sich natürlich nicht, da in diesem Programm alle Visitenkarten gleich aufgebaut sein sollen. Bedienen Sie sich daher einmal mehr der objektorientierten Programmierung zur Erzeugung von weiteren Kindformen zur Laufzeit. Lediglich zwei Programmzeilen sind dazu notwendig.

- 3. Erzeugen Sie folgende Anweisungen in der Ereignisprozedur MenüNeu_Click:
 - 1. Private Sub MenüNeu_Click()
 - 2. Dim NeueKindform As New FormKindform

- 3. NeueKindform.Caption = InputBox("Bezeichnung", "Visitenkarte")
- 4. NeueKindform.Show
- 5. End Sub

Mit dem Befehl

Dim NeueKindform As New FormKindform

erzeugen Sie eine weitere Instanz – für den Augenblick reicht auch der Begriff "Kopie" aus – der bereits bestehenden Form *FormKindform*. Die neue Kindform erbt damit alle Steuerelemente, Eigenschaften, Methoden, Funktionen und Prozeduren der Form *Form-Kindform*, die diese zur Entwicklungszeit besitzt. Sie besitzt damit automatisch alle Bezeichnungsfelder, das Bildfeld und die Figur auf der neuen Form, ebenso wird der Programmtext der Ereignisprozedur *Form_Load* mit übernommen.



Mit Hilfe der InputBox erzeugt der Anwender einen spezifischen Titelzeilenbeschriftungstext für jede einzelne Kindform, also für jede Visitenkarte.

4 Starten Sie einen Probelauf und wählen Sie den Menübefehl *Vistenkarten/Neue Vistenkarten karte* mehrfach. Wählen Sie anschließend den Menübefehl *Vorhandene Visitenkarten*.

Wo kommen diese Menüeinträge denn auf einmal her? Durch die Festlegung der Eigenschaft *WindowList* des Menüs *MenüVorhanden* werden automatisch alle vorhandenen Kindformen per Titelzeilenbeschriftungstext als Menüpunkte zu diesem Menü hinzugefügt. Durch Auswahl dieser Menüeinträge kann der Anwender somit leicht zwischen den einzelnen Kindformen hin- und herwechseln.

Da Sie mit mehreren Fenstern gleichzeitig arbeiten wollen, ist es sinnvoll, daß Sie diese platzsparend und gleichmäßig anordnen können. Zu diesem Zweck haben Sie schon das Menü *Fenster* erzeugt, das in keiner MDI-Anwendung fehlen sollte. Mit Hilfe der Methode *Arrange*, einer MDI-Form, sind Sie in der Lage, die Kindformen anordnen zu können. Mit dem Befehl

MDIForm.Arrange Wert

Multipler Visitenkarteneditor
 Eenster Einstellungen Visiterkarten
Vorhandene Visitenkarten
Visitenkarte
 Moritz
 Moritz
 Amax

ordnen Sie die Kindformen in verschiedenen Möglichkeiten an.

Möglichkeiten der Methode Arrange einer MDI-Form

Wert	Bedeutung
0	Die Fenster werden kaskadiert, überlappt dargestellt.
1	Die Fenster werden nebeneinander dargestellt.
2	Die Fenster werden untereinander dargestellt.
3	Die Symbole der auf Symbolgröße verkleinerten Formen werden angeordnet.



Die Parameter für die Methode *Arrange* haben nun die gleichen Werte wie die Indizes der Menüpunkte des Menüsteuerelementfelds *MenüFeld* mit den entsprechenden Beschriftungen.

- 5. Das ermöglicht folgenden einfachen Programmbefehl, den Sie nun erzeugen sollen:
 - 1. Private Sub MenüFeld_Click(Index As Integer)
 - 2. MDIForm1.Arrange Index
 - 3. End Sub

8

Der Methode *Arrange* wird der Index des gewählten Menüpunkts übergeben. Dadurch wird die entsprechende Anordnung bewirkt. Wird beispielsweise der Menüpunkt *Untereinander* ausgewählt, hat der Parameter *Index* den Wert 2. Wird der Methode *Arrange* der Parameter 2 übergeben, werden alle Kindformen untereinander angeordnet. Auf diese Art und Weise können Sie sich beispielsweise eine *Select Case*-Anweisung der folgenden Art ersparen:

Select Case Index Case 0 MDIForm1.Arrange 0 Case 1 MDIForm1.Arrange 1 Case 2 MDIForm1.Arrange 3 Case 3 MDIForm1.Arrange 3 End Select

Die aktive Kindform ermitteln

So weit, so gut, langsam sollte der Anwender aber auch in der Lage sein, Visitenkarten zu erstellen und nicht nur mit leeren Kindformen herumspielen.

SUPER STEPS + + + + + + +

1 Fügen Sie dem Projekt die Form *FormDaten.frm* aus dem Ordner *KapitelO8\Leser* hinzu.

Auf dieser Form kann der Anwender die entsprechenden Angaben für die anzuzeigenden Texte treffen sowie deren Positionen verändern. Soll beispielsweise die Firmenbezeichnung nicht angezeigt werden, wird einfach kein Wert im betreffenden Textfeld angegeben. Zusätzlich wird dem Anwender hier die Auswahl einer Grafikdatei ermöglicht. Es ist natürlich ohne

s. Layoutangaben			
Datan		Texteinstellungen	
Neme	e.	Schrittgröße	< >
Stresse	· ·	Position horizontal	< >
Wohnort		Position vertikal	< >
Telefon	· ·	Symbol	
Fex	•		
E-Mail-Adresse	0		
Titel	· ·	Anzeigen	9
Fima	· ·	Position horizontal Position vertikal	< >

weiteres möglich, eine Form im Projekt zu benutzen, die nicht als Kindform fungiert, Projekte mit MDI-Formen können ebenso "normale" Formen beinhalten.

- 2. Erzeugen Sie den notwendigen Befehl zur Anzeige dieser Form:
 - 1. Private Sub MenüDaten_Click()
 - 2. FormDaten.Show
 - 3. End Sub

Der Dateneingabedialog dieser Form soll für die jeweils aktive Kindform gelten, es können ja mehrere Kindformen vorhanden sein. Zur Belegung der Eigenschaften der entsprechenden Steuerelemente ist es dazu notwendig, Kenntnis über die gerade aktive Kindform zu besitzen. Dazu benutzen Sie die Eigenschaft *ActiveForm* der MDI-Form.

MDIForm.ActiveForm

Dem Bildfeld der Form *FormDaten* sowie dem Bildfeld der aktiven Kindform soll nach Klick auf das Bildfeld der Form *FormDaten* mit Hilfe des Standarddialogs die betreffende Grafikdatei zugeordnet werden.

3. Ergänzen Sie dazu den fettgedruckten Programmtext in der Ereignisprozedur *Bildfeld* __*Click* der Form *FormDaten*.

1. Private Sub Bildfeld Click()

- 2. On Error Resume Next
- 3. StandardDialog.ShowOpen
- 4. Bildfeld.Picture = LoadPicture(StandardDialog.Filename)
- 5. MDIForm1.ActiveForm.Bildfeld.Picture = Bildfeld.Picture
- 6. End Sub

Dem auf der aktiven Kindform befindlichen Bildfeld wird das gleiche Bild, das dem Bildfeld auf der Form *FormDaten* zugeordnet ist, zugewiesen. Es wird in dieser Anweisung zunächst die MDI-Form, dann deren aktive Kindform, dann das darauf befindliche Bildfeld und abschließend dessen Eigenschaft *Picture* angesprochen.

MDIForm.ActiveForm.Steuerelement.Eigenschaft

Ebenso wird das Bildfeld auf der aktiven Kindform durch das Kontrollfeld auf der Form *FormDaten* sichtbar oder unsichtbar geschaltet:

- 1. Private Sub CheckSichtbar_Click()
- 2. MDIForm1.ActiveForm.Bildfeld.Visible = Not MDIForm1.ActiveForm.Bildfeld.Visible
- 3. End Sub

Gibt der Anwender Text in ein Textfeld auf der Form *FormDaten* ein, soll dieses zusätzlich im entsprechenden Bezeichnungsfeld auf der aktiven Kindform angezeigt werden. Auf der Form *FormDaten* befinden sich daher acht Textfelder eines Steuerelementfelds, ebenso befinden sich acht Bezeichnungsfelder eines Steuerelementfelds auf der Form *FormKindform*.

- 1. Private Sub Eingabe_Change(Index As Integer)
- 2. MDIForm1.ActiveForm.Anzeige(Index) = Eingabe(Index).Text
- 3. End Sub

Durch Zuhilfenahme des Indexes kann so leicht der eingegebene Text im Textfeld der Form *FormDaten* im entsprechenden Bezeichnungsfeld der aktiven Kindform angezeigt werden. Bedenken Sie, wie viele Programmanweisungen Sie hätten erstellen müssen, wenn für die Bezeichnungs- und Textfelder keine Steuerelementfelder verwendet worden wären.

Beim Wechsel mit dem Cursor in ein Textfeld auf der Form *FormDaten* soll das nebenstehende Optionsfeld aktiviert werden, damit zur Positionierung der Bezeichnungsfelder auf der Kindform eine Unterscheidungsmöglichkeitbesteht.

- 1. Private Sub Eingabe_GotFocus(Index As Integer)
- 2. OptionAuswahl(Index).Value = True
- 3. End Sub

8

Neben der Angabe des betreffenden Textes soll der Anwender die Möglichkeit besitzen, die einzelnen Texte beliebig auf der Form anordnen zu können, um individuelle Visitenkarten zu erstellen. Anhand der Optionsfelder soll angegeben werden, welcher Text – also welches Bezeichnungsfeld – der Kindform in Position oder Schriftgröße verändert werden soll.

- 4. Ergänzen Sie dazu den fettgedruckten Programmtext.
 - 1. Private Sub SchalterGrösser_Click(Index As Integer)
 - 2. Dim Steuerelement As Control
 - 3. Dim Nummer As Integer
 - 4. For Each Steuerelement In Me
 - 5. If TypeOf Steuerelement Is OptionButton Then
 - 6. If Steuerelement.Value = True Then
 - 7. Nummer = Steuerelement.Index
 - 8. End If
 - 9. End If
 - 10. Next
 - 11. Select Case Index
 - 12. Case 1
 - 13. ...
 - 14. End Select
 - 15. End Sub

Mit der Schleife wird zunächst überprüft, welches Optionsfeld ausgewählt ist. Der ermittelte Index dieses Optionsfelds wird in der Variablen *Nummer* gespeichert. Dieser Wert dient im folgenden zur Ansprechung des betreffenden Bezeichnungsfelds auf der aktiven Kindform.

Mit der Befehlsschaltfläche > soll der Anwender die Möglichkeit besitzen, die x- oder y-Koordinaten der Bezeichnungsfelder, des Bildfelds oder die Schriftgröße – je nach Auswahl der betreffenden Schaltfläche, erkennbar am Parameter *Index* – vergrößern zu können.

In der Ereignisprozedur *SchalterKleiner_Click* werden diese Werte entsprechend verringert, der Programmtext unterscheidet sich nur durch die Minuszeichen gegenüber den hier anzutreffenden Pluszeichen für die Wertveränderungen. Sie haben recht, diese Unterscheidung hätte auch in einer Ereignisprozedur stattfinden können.

Die Form *FormDaten* kann vom Anwender nur durch die entsprechende Titelzeilen-Schaltfläche geschlossen werden, da diesem keine spezielle Schaltfläche zu diesem Zweck zur Verfügung steht. Da diese Form für alle Kindformen zuständig ist und in einem gleich folgenden Schritt dort die jeweiligen Daten der aktiven Kindform angezeigt werden sollen, ist es sinvoll, daß die Form permanent angezeigt wird.



- 1. Private Sub Form_Unload(Cancel As Integer)
- 2. Cancel = True
- 3. Me.Hide
- 4. End Sub

Durch Zuweisung des Werts *True* an den Parameter *Cancel* der Ereignisprozedur *Form_Unload* wird die Formentladung unterbunden. Alternativ wird hier die Form durch die Methode *Hide* nur ausgeblendet.

- 6. Zur ordnungsgemäßen Beendigung des Programms benötigen Sie nun aber den Befehl *End* in der Ereignisprozedur *Form_Unload* der MDI-Form, damit auch wirklich alle Kindformen entladen werden.
 - 1. Private Sub Form_Unload(Cancel As Integer)
 - 2. End
 - 3. End Sub

Im Sinne komfortabler Bearbeitungsmöglichkeiten wäre es für den Anwender doch äußerst nützlich, wenn die jeweiligen Daten einer Visitenkarte automatisch bei deren Aktivierung in der Form *FormDaten* angezeigt würden. Dies bewerkstelligen Sie ganz einfach.

- **7**. Erzeugen Sie dazu folgenden Programmtext in der Ereignisprozedur *Form_Activate* der Form *FormKindform*:
 - 1. Private Sub Form_Activate()
 - 2. Dim N As Integer
 - 3. For N = 0 To 7
 - 4. FormDaten.Eingabe(N).Text = Anzeige(N).Caption
 - 5. Next N
 - 6. FormDaten.Bildfeld.Picture = Bildfeld.Picture
 - 7. End Sub

Das Ereignis *Activate* einer Form tritt ein, wenn diese aktiviert wird, also den Fokus erhält. Im Gegensatz dazu tritt das Ereignis *Deactivate* ein, wenn die Form den Fokus verliert. Bei jedem Wechsel zwischen den einzelnen Visitenkarten werden nun die entsprechenden Daten auf der Form *FormDaten* präsentiert.

Visitenkarten drucken

- **1** Zum Ausdrucken einer Visitenkarte benötigen Sie lediglich eine Programmanweisung. Erzeugen Sie diese in der Ereignisprozedur *MenüDrucken_Click*:
 - 1. Private Sub MenüDrucken_Click()
 - 2. ActiveForm.PrintForm
 - 3. End Sub

Die Methode *PrintForm* einer Form druckt die betreffende Formansicht als Grafik aus. Form.PrintForm Sollen durch Grafikoperationen erzeugte Grafiken ebenfalls ausgedruckt werden, ist die Eigenschaft *AutoRedraw* der Form auf *True* zu setzen. Ganz schön simpel, aber auch ganz schön komfortabel, nicht wahr? Testen Sie das Programm nun in aller Ausführlichkeit.

2. Erzeugen Sie 100 Visitenkarten und drucken Sie diese aus! Na ja, gemeint ist eigentlich nur, daß Sie das Programm mal ein bißchen testen sollen!

8

Nicht schlecht, oder? Ärgern Sie sich nicht, daß Sie momentan noch keine Visitenkarten speichern und auch wieder laden können. Das Wissen zu diesem Zweck werden Sie im letzten Kapitel des Buchs erlangen.



MDI-Formen



MDI-Formen dienen zur Aufnahme mehrerer (gleichartiger) Kindformen. MDI-Formen werden auch als Container für die enthaltenen Dokumentenfenster bezeichnet. Kindformen können nicht aus der MDI-Form herausbewegt werden, sie sind an diese gebunden.

Transparente Steuerelemente $\,\, { m O}$

Damit Steuerelemente transparent erscheinen können, müssen Sie den Wert der Eigenschaft *BackStyle* anpassen.

TypeOf-Anweisung

ו

Mit der *TypeOf*-Anweisung ermitteln Sie den Typ eines Steuerelements. Dies ist zum Beispiel bei einer Schleife, die alle Steuerelemente einer Form durchläuft, zur besseren Unterscheidung nützlich.

Fensterliste – WindowList

Markieren Sie die Eigenschaft *WindowsList* bei einem *MDI-Menüpunkt*, werden in diesem Menü zur Laufzeit automatisch alle geladenen Kindformen aufgelistet.

Dynamisch eine neue Form laden

Ο

Mit der Anweisung *Dim NeueForm As New VorhandeneForm* erzeugen Sie eine weitere Instanz (eine Kopie) einer bestehenden Form. Die neue Form erbt alle Steuerelemente, Eigenschaften, Methoden, Funktionen und Prozeduren der vorhandenen Form, die diese zur Entwicklungszeit besitzt.

Kindformen anordnen

Die Methode Arrange ermöglicht vielfältige Anordnungen von Kindformen innerhalb einer MDI-Form.

Aktive Kindform ermitteln



Müssen Sie im Programmtext Bezug auf die jeweils aktive Kindform nehmen, bedienen Sie sich der Anweisung *MDI-Form.ActiveForm*.

Grafischen Inhalt einer Form ausdrucken

Die Methode *PrintForm* druckt die betreffende Formansicht als Grafik aus.