Kapitel 5

Berichte

Der Bericht (*Report*) ist neben dem Formular (*Form*) die wichtigste visuelle Benutzerschnittstelle unter Access. Bei den zur Verfügung stehenden Eigenschaften, Ereignissen und Methoden gibt es viele Gemeinsamkeiten mit dem *Form*-Objekt (siehe Kapitel 4).

Bereits mit Access 2007 wurden die Berichte um eine Reihe neuer Möglichkeiten bei der Interaktion mit dem Anwender (Maus, Tastatur) erweitert. Möglich wurde dies durch zwei neue Ansichten (Berichtsansicht, Layoutansicht), deren Funktionalität in Access 2010 etwas ergänzt wurde.

Außerdem verfügt das *Report*-Objekt (im Gegensatz zum Formular) über ein recht umfangreiches Arsenal an Grafikeigenschaften und -methoden, die dem Programmierer eine Vielzahl von optischen Gestaltungsmöglichkeiten bieten.

5.1 Allgemeines

Bevor wir uns den Eigenschaften, Methoden und Ereignissen des *Report*-Objekts zuwenden, wollen wir uns zunächst allgemeineren Fragen widmen, die mit dem Öffnen eines Berichts zusammenhängen.

5.1.1 Reportansichten

Öffnen Sie einen Bericht durch Doppelklick auf seinen Namen im Navigationsbereich, so wird standardmäßig die *Berichtsansicht* angezeigt (diese Einstellung kann mittels *DefaultView*-Eigenschaft geändert werden).

- In der *Berichtsansicht* können Sie z.B. Filter anwenden, Daten kopieren oder auf Steuerelemente oder Hyperlinks klicken
- In der *Layoutansicht* können Sie das Ergebnis am Format und an weiteren Einstellungen sofort im Kontext mit den Daten betrachten, Sie können z.B. Spalten verschieben, ihre Größe verändern oder neue Felder aus der Feldliste hinzufügen
- Um einen Bericht von Grund auf neu zu erstellen verwenden Sie die *Entwurfsansicht*

- Mit Hilfe der *Seitenansicht* können Sie vor dem Drucken überprüfen, wie das Ergebnis auf dem Papier aussehen wird

5.1.2 Die OpenReport-Methode

Zum Öffnen eines Reports verwendet man die *OpenReport*-Methode des *DoCmd*-Objekts (vergleichbar mit der *OpenForm*-Methode für Formulare).

Die Syntax zeigt, dass außer dem Berichtsnamen alle weiteren Argumente optional sind:

```
DoCmd.OpenReport(ReportName, [View], [FilterName], [WhereCondition], [WindowMode], [OpenArgs])
```

- *ReportName*
  Der Name des Berichts.
- *View*
  Eine *AcView*-Konstante, die die Ansicht bestimmt, in welcher der Report geöffnet wird .
- *FilterName*
  Namen einer gültigen Abfrage in der aktuellen Datenbank.
- *WhereCondition*
  Eine gültige SQL-WHERE-Bedingung (ohne WHERE).
- *WindowMode*
  Eine *AcWindowMode*-Konstante, welche den Modus angibt, in welcher das Formular geöffnet wird (*acWindowNormal, acHidden, acIcon, acDialog*).
- *OpenArgs*
  Bestimmt die *OpenArgs*-Eigenschaft, mit welcher Parameter an den Report übergeben werden können.

Die *AcView*-Konstanten:

Konstante	Wert	Erklärung
acViewNormal	0	Druckansicht (Standard)
acViewDesign	1	Entwurfsansicht
acViewPreview	2	Seitenansicht (Berichtsvorschau)
acViewPivotTable	3	PivotTable-Ansicht
acViewPivotChart	4	PivotChart-Ansicht
acViewReport	5	Berichtsansicht
acViewLayout	6	Layoutansicht

BEISPIEL: Der Bericht *Mitarbeiter* wird geöffnet, alle Mitarbeiter mit dem Namen "Müller" werden angezeigt.

```
DoCmd.OpenReport "Mitarbeiter", , , "Nachname = 'Müller'"
```

5.1.3 Parameterübergabe

Das *OpenArgs*-Argument der *OpenReport*-Methode kann für die Übergabe zusätzlicher Informationen an den Bericht genutzt werden.

BEISPIEL: Zusätzlich zur WHERE-Bedingung wird ein Datum, welches in einem Textfeld steht, per *OpenArgs* an einen Bericht übergeben.

```
Dim whereCond As String
whereCond = "Anzahl > 100 AND Auslaufartikel = True"                    ' WHERE-Bedingung
definieren
DoCmd.OpenReport "Bericht1", acViewPreview, , whereCond, , Text1.Value  ' OpenArgs = Text1.Value
```

Anzeige des Datums im Bericht:

```
Private Sub Report_Open(Cancel As Integer)
   Bezeichnungsfeld0.Caption = "Geräteliste vom " & Me.OpenArgs
   Me.FilterOn = True
End Sub
```

HINWEIS: Eine komplette Anwendung, die die verschiedenen Ansichten eines Reports demonstriert, finden Sie im Praxisbeispiel "Aufruf eines Berichts mit Datenfilter" (Seite 295).

5.2 Wichtige Berichtseigenschaften

Bei der folgenden knappen Zusammenstellung der Eigenschaften, Ereignisse und Methoden werden nur diejenigen näher erläutert, die neu gegenüber dem *Form*-Objekt (siehe Kapitel 4) sind bzw. die sich in ihrer Funktion wesentlich unterscheiden.

5.2.1 Formateigenschaften

Diese Eigenschaften bestimmen das "Outfit" eines Berichts und entsprechen größtenteils denen des *Form*-Objekts.

Eigenschaft	Deutscher Bezeichner	Standard	Erläuterung
AllowLayoutView	Layoutansicht zulassen	True	erlaubt/verbietet Öffnen in Layoutansicht
AllowReportView	Berichtsansicht zulassen	True	erlaubt/verbietet Öffnen in Berichtsansicht
Caption	Beschriftung	Bericht	Text in Titelleiste
DefaultView	Standardansicht	0	0=Seitenansicht, 1=Berichtsansicht
PageHeader, PageFooter	Seitenkopf, Seitenfuß	0	Der Seitenkopf/-fuß wird auf allen Seiten eines Berichts gedruckt (1=außer Berichtskopf, 2=außer Berichtsfuß, außer Berichtskopf/-fuß).

Eigenschaft	Deutscher Bezeichner	Standard	Erläuterung
GrpKeepTogether	Gruppe zusammenhalten	1	Gruppen werden pro Spalte zusammengehalten (0=pro Seite)
GridX, GridY	Raster X, Raster Y	10	Horizontale bzw. vertikale Einheiten des Gitternetzes (Entwurfsansicht)
Height	Höhe		Gesamthöhe aller Bereiche
LayoutForPrint	Drucklayout	*False*	Verwenden von Druckerschriftarten (*True*) oder Bildschirmschriftarten (*False*)
Width	Breite		Breite des Detailbereichs
Picture	Bild	(keines)	Pfad/Dateinamen der anzuzeigenden Bitmap
PictureAlignment	Bildausrichtung		Ort, wo die Hintergrundbitmap erscheinen soll (siehe *Form*-Objekt)
PictureSizeMode	Bildgrößenmodus		Darstellung des Hintergrundbildes (siehe *Form*-Objekt)
PictureTiling	Bild nebeneinander		Bilder nebeneinander anordnen
PictureType	Bildtyp	0	Bild wird als eingebettetes (0) oder verknüpftes (1) Objekt gespeichert
PicturePages	Bildseiten	0	Seite(n), auf welcher ein Bild angezeigt werden soll (alle=0, erste=1, 2=keine)

5.2.2 Dateneigenschaften

Dateneigenschaften bilden eine Untermenge der entsprechenden Eigenschaften des *Form*-Objekts.

Eigenschaft	Deutscher Bezeichner	Standard	Erläuterung
RecordLocks	Datensätze sperren	0	siehe *Form*-Objekt
RecordSource	Datenherkunft		siehe *Form*-Objekt
Filter	Filter		siehe *Form*-Objekt
FilterOn	Filter aktiv	*False*	siehe *Form*-Objekt
FilterOnLoad	Beim Laden filtern	*False*	siehe *Form*-Objekt
OrderBy	Sortiert nach		siehe *Form*-Objekt
OrderByOn	Sortierung aktiv	*True*	siehe *Form*-Objekt
OrderOnLoad	Beim Laden sortieren	*True*	siehe *Form*-Objekt

5.2.3 Grafikeigenschaften

Eigenschaften (Properties) dieser Kategorie stehen im Zusammenhang mit der Ausführung von Grafikmethoden. Für das den Grafikoperationen zugrunde liegende Koordinatensystem sind folgende Eigenschaften maßgebend:

Eigenschaft	Standard-wert	Erläuterung
ScaleMode	1	Maßeinheit für Koordinatenangaben (Twips)
ScaleWidth, ScaleHeight		Anzahl der Einheiten für horizontale oder vertikale Innenab-messungen der Seite
ScaleLeft, ScaleTop	0, 0	Koordinaten für linke obere Ecke einer Seite
CurrentX, CurrentY	0, 0	Festlegung der aktuellen Koordinaten für Grafikmethoden

ScaleMode

Durch Ändern der *ScaleMode*-Eigenschaft können Sie sich für eine andere Maßeinheit als die standardmäßig angebotenen *Twips* entscheiden:

Wert	Bezeichnung	Erläuterung
0	User	Nutzerdefiniert, siehe *ScaleWidth*, *ScaleHeight*
1	Twips	Standardeinstellung: 1.440 Twips = 1 Inch (Zoll)
2	Point	1 Point = 0,353 mm; 72 Point = 1 Inch
3	Pixel	Entspricht Bildschirmauflösung
4	Character	1 Zeichenhöhe = 1/6 Inch; 1 Zeichenbreite = 1/12 Inch
5	Inch	1 Inch (Zoll) = 2,54 cm
6	Millimeter	1 mm = 56,7 Twips
7	Zentimeter	1 cm = 567 Twips

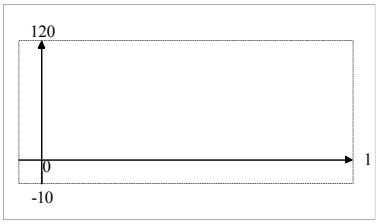
ScaleLeft und ScaleTop, ScaleWidth und ScaleHeight

Mit Hilfe dieser Eigenschaften lässt sich auf der Berichts-Oberfläche ein vollständiges Koordinatensystem mit positiven und negativen Koordinaten einrichten. Standardmäßig verläuft die y-Koordinate von oben nach unten. Durch geeignete Skalierung können Sie aber auch die gewohnte, von unten nach oben verlaufende y-Achse einstellen.

BEISPIEL: Die Anweisungen:

```
ScaleWidth = 140
ScaleHeight = 130
ScaleLeft = - 10
ScaleTop = - 120
```

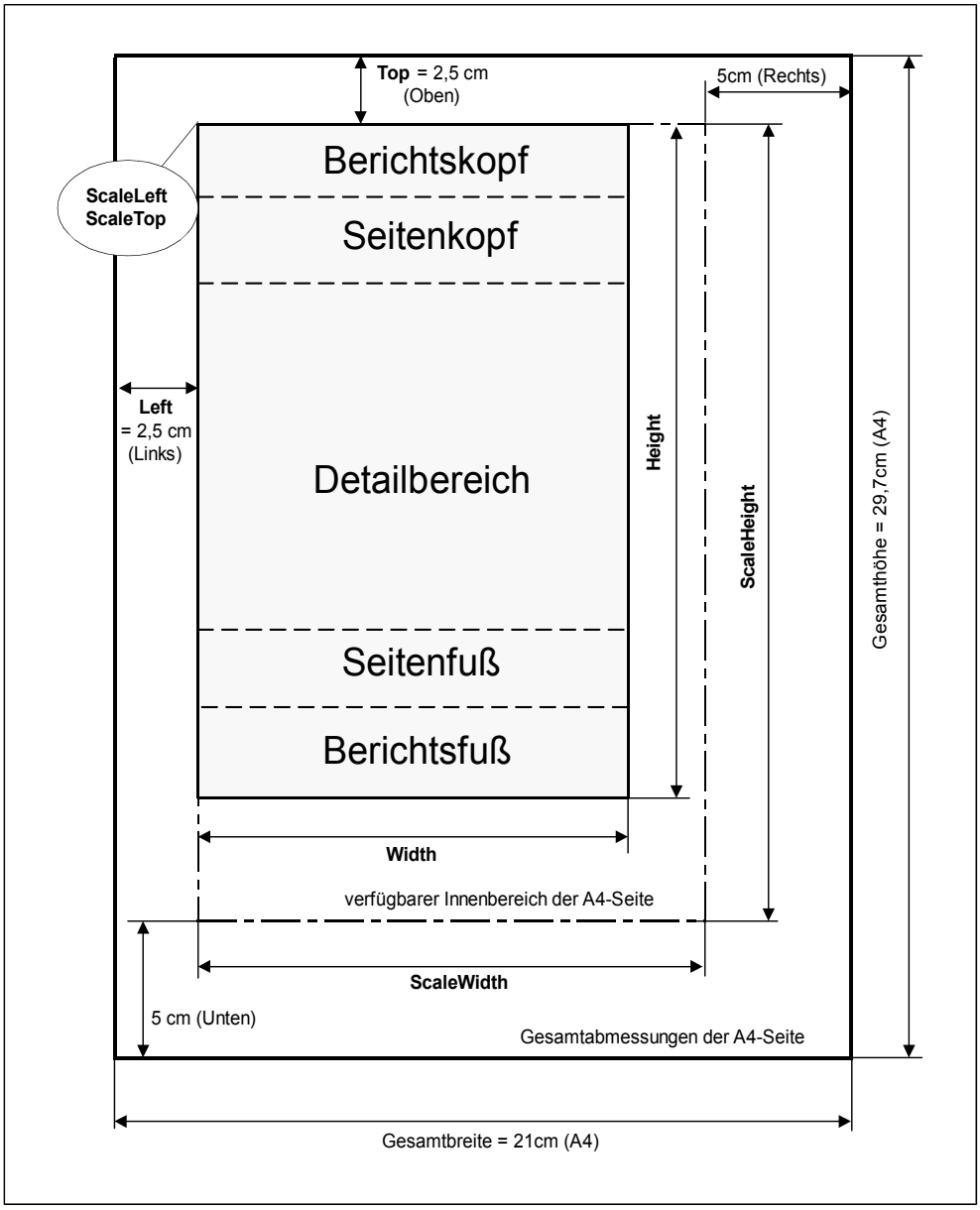
definieren das in der folgenden Abbildung gezeigte Koordinatensystem.



Left, Top, Width und Height

Die folgende Abbildung zeigt die Verhältnisse für das *Page*-Ereignis.

Hier entsprechen *Left* und *Top* dem linken und dem oberen Randabstand, wie er im Dialogfeld zum Menübefehl *Seite einrichten/Seitenlayout/Seite einrichten* festgelegt ist. *Width* und *Height* beziehen sich auf die Gesamtabmessungen des vom Berichts- und Seitenkopf, Detailbereich und Seiten- und Berichtsfuß definierten Gebietes, wie es beim Entwurf aufgezoomt werden kann.



HINWEIS: Verwechseln Sie *ScaleLeft, ScaleTop, ScaleHeight und ScaleWidth* nicht mit den Eigenschaften *Left, Top, Height* und *Width*!

Die *Scale*-Eigenschaften beziehen sich ausschließlich auf das innere Koordinatensystem des druckbaren Innenbereichs der Berichtsseite und auf die mit *ScaleMode* festgelegte Maßeinheit. Hingegen ist die Bedeutung von *Left*, *Top*, *Width* und *Height* davon abhängig, ob sie im *Page*-Ereignis des *Report*-Objekts oder im *Print*- bzw. *Format*-Ereignis eines seiner Bereiche (Detailbereich, Seitenkopf/-fuß, Berichtskopf/-fuß) stehen.

HINWEIS: *Left*, *Top*, *Width* und *Height* werden immer in *Twips* interpretiert, sind also unabhängig von der *ScaleMode*-Einstellung!

BEISPIEL: Der folgende Event-Handler liefert die kommentierten Ergebnisse, wenn die Seitenränder des Berichts durchgängig auf 2,5 cm eingestellt wurden (A4-Format) und Kopf-, Fuß- und Detailbereich die beim Entwurf festgelegten Abmessungen haben.

```
Private Sub Report_Page()
 ScaleMode = 7          ' Zentimeter
 Print Left / 567       ' linker Rand = 2,5 cm
 Print Top / 567        ' oberer Rand = 2,5 cm
 Print Height / 567     ' Gesamthöhe Kopf-/Fuß-/Detailbereich = 2 cm
 Print Width / 567      ' Breite Detailbereich = 10 cm
 Print ScaleLeft        ' X-Verschiebung des Koordinatensystems = 0
 Print ScaleTop         ' Y-    "         "       "             = 0
 Print ScaleHeight      ' Höhe des druckbaren Bereichs = 24,7 cm
 Print ScaleWidth       ' Breite des druckbaren Bereichs = 16 cm
End Sub
```

Sie können die Korrektheit überprüfen, indem Sie zum Beispiel zu *ScaleWidth* die linke und die rechte Randbreite addieren. Als Ergebnis erhalten Sie im vorliegenden Fall 21 cm (DIN A4-Breite).

HINWEIS: Wenn Sie nicht innerhalb des *Report_Page*-, sondern innerhalb des *Format*- bzw. *Print*-Events zeichnen wollen, bezieht sich *Height* immer nur auf die Höhe des entsprechenden Bereichs (Detailbereich, Berichtskopf etc.), und der Koordinatenursprung (0,0) liegt in der linken oberen Ecke des Bereichs.

Bemerkungen

- Statt der Eigenschaften *ScaleHeight*, *ScaleWidth*, *ScaleLeft* und *ScaleTop* könnten Sie auch die *Scale*-Methode verwenden. Allerdings eignet sich diese nur zum Zuweisen, nicht aber zum Lesen der Abmessungen eines Koordinatensystems.
- Wenn Sie für *ScaleMode* einen Wert größer als 0 einstellen, werden die Eigenschaften *ScaleHeight* und *ScaleWidth* in die neue Maßeinheit geändert. Für *ScaleLeft* und *ScaleTop* wird automatisch der Wert 0 eingestellt. Darüber hinaus ändern sich auch die Einstellungen der Eigenschaften *CurrentX* und *CurrentY*.

CurrentX und CurrentY

Mit diesen Eigenschaften stellen Sie die aktuelle Druckposition für Grafikmethoden ein, wie es das folgende Beispiel zeigt.

BEISPIEL: Etwa in die Mitte der untersten Zeile eines A4-Blattes wird der Text "Hallo" ausgedruckt.

```
Private Sub Report_Page()
 Me.ScaleMode = 7            ' Zentimeter
 CurrentX = 8
 CurrentY = 24
 Print "Hallo"
End Sub
```

HINWEIS: Beachten Sie, dass sich die aktuelle Druckposition (*CurrentX/CurrentY*) nach Abschluss einer Grafikoperation ebenfalls verschoben hat.

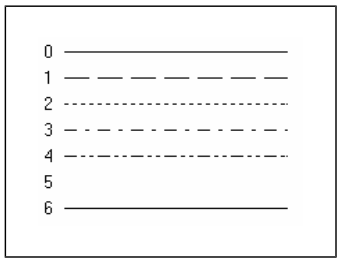
5.2.4 Linien- und Stifteigenschaften

Damit legen Sie die Art der Linien fest, mit denen die Grafikmethoden *Line*, *Circle* oder *PSet* zeichnen.

Eigenschaft	Bezeichnung	Standard	Erläuterung
DrawWidth	Linienbreite	1	Stärke der gezogenen Linien
DrawStyle	Linientyp	0	Nur für *DrawWidth* = 1
DrawMode	Zeichnungsmodus	13	Art der Verknüpfung mit dem Hintergrund (*CopyPen)*

DrawWidth und DrawStyle

DrawWidth gibt die Breite einer Linie in Pixel an. Die Voreinstellung ist 1, was einer Breite von 1 Pixel entspricht. Ist der Wert größer als 1, erzeugen die Einstellungen 1 bis 4 für *DrawStyle* eine durchgezogene Linie. Wenn Sie für die *DrawWidth* den Wert 1 wählen, können Sie durch Zuweisen von *DrawStyle* die in der folgenden Übersicht enthaltenen Linienarten für die Methoden *Line* und *Circle* erhalten:



DrawMode

Mit dieser Eigenschaft (1...16) können Sie festlegen, ob und wie sich die für Grafikmethoden verwendete Farbe mit der Hintergrundfarbe des Berichts zur resultierenden Linienfarbe kombiniert. Standardmäßig ist der Wert 13 eingestellt (*CopyPen*), d.h., die Stiftfarbe entspricht der *ForeColor*-Property. Eine Beschreibung der übrigen Einstellungen finden Sie im Übersichtsteil dieses Kapitels (Seite 294).

5.2.5 Schrifteigenschaften

Diese Eigenschaften beziehen sich auf die *Print*-Methode.

Eigenschaft	Bezeichnung	Standardwert	Erläuterung
FontBold	Fettschrift	False	Nicht fett
FontItalic	Kursivschrift	False	Nicht kursiv
FontName	Schriftart		Abhängig von Systemschriftarten
FontSize	Schriftgröße	8	1 Punkt = 0,0353cm (1/72 Zoll)
FontStrikeThru	Durchgestrichen	False	Nicht durchgestrichen
FontUnderline	Unterstrichen	False	Nicht unterstrichen

5.2.6 Farb- und Mustereigenschaften

Für die Ausgabe von Farben und Mustern sind folgende Eigenschaften von Bedeutung:

Eigenschaft	Bezeichnung	Standard	Bemerkung
ForeColor	Textfarbe	Schwarz	Auch Linien- bzw. Rahmenfarbe für Rechtecke und Kreise
FillColor	Füllfarbe	Schwarz	Füllfarbe für Rechtecke und Kreise
FillStyle	Füllmuster	Transparent	Füllmuster für Rechtecke und Kreise
BackColor	Hintergrundfarbe	Weiß	&HFFFFFF

ForeColor und BackColor

Die Eigenschaft *ForeColor* gibt die Farbe an, welche die Methoden *Print*, *Line* und *Circle* für die Ausgabe verwenden. Wenn Sie einen Farbwert in einer Variablen speichern wollen, brauchen Sie den Datentyp *Long*.

BEISPIEL: Die folgende Sequenz druckt "Hallo" in roter Schrift:

```
Dim rot As Long
rot = vbRed
Me.ForeColor = rot
Me.Print "Hallo"
```

Die Eigenschaft *BackColor* können Sie auf bestimmte Bereiche des Berichts anwenden.

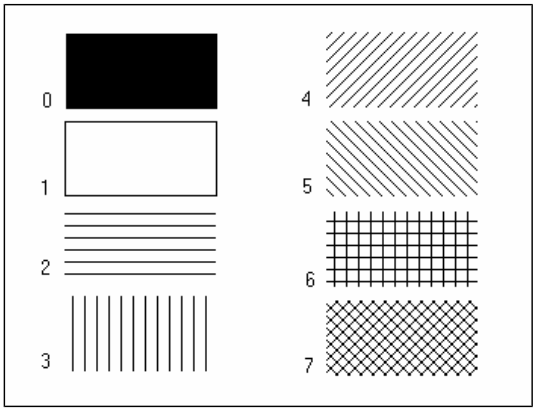
BEISPIEL: Die folgende Sequenz färbt den Detailbereich gelb.

```
Private Sub Report_Open(Cancel As Integer)
 Me.Section(0).BackColor = QBColor(14)
End Sub
```

FillColor und FillStyle

Die *FillColor*-Eigenschaft wird nur berücksichtigt, wenn die *FillStyle*-Property nicht transparent ist (also ungleich der Voreinstellung 1!). Die Farbe des Füllmusters hängt von *FillColor* ab.

Die möglichen Werte für *FillStyle* entnehmen Sie bitte der Abbildung.



QBColor- und RGB-Funktion

Insgesamt 16 Grundfarben lassen sich mit der *QBColor*-Funktion erzeugen.

Code	Farbe	Code	Farbe	Code	Farbe	Code	Farbe
0	Schwarz	4	Rot	8	Dunkelgrau	12	Hellrot
1	Blau	5	Magenta	9	Hellblau	13	Hellmagenta
2	Grün	6	Ocker	10	Hellgrün	14	Hellgelb
3	Zyan	7	Hellgrau	11	Hellzyan	15	Weiß

BEISPIEL: Die Hintergrundfarbe eines Bildfeldes erhält einen zufälligen Wert:

```
Bild1.BackColor = QBColor(Rnd * 15)
```

Wesentlich feinere Farbabstufungen lassen sich mit der *RGB*-Funktion erreichen, mit welcher Sie aus den drei Grundfarben Rot, Grün und Blau eine beliebige Farbe mischen können.

```
farbe = RGB (rot, grün, gelb)
```

Die Argumente *rot, grün, blau* können Integer-Werte zwischen 0 und 255 annehmen.

Die folgende Tabelle enthält einige Standardfarben mit den zugehörigen Rot-, Grün- und Blau-Anteilen:

Farbe	Rot	Grün	Blau
Schwarz	0	0	0
Blau	0	0	255
Grün	0	255	0
Zyan	0	255	255
Rot	255	0	0
Magenta	255	0	255
Gelb	55	255	0
Weiß	255	255	255

BEISPIEL: Verwendung von *RGB*:

```
rot = RGB(255,0,0)
schwarz = RGB(0,0,0)
```

Farbkonstanten

Für eine Vielzahl von Farben können auch die in VBA integrierten Farbkonstanten wie *vbRed, vbWhite* etc. Verwendung finden.

BEISPIEL: Eine rote Stiftfarbe wird eingestellt:

```
Me.ForeColor = vbRed
```

HINWEIS: Eine Zusammenstellung wichtiger Farbkonstanten finden Sie im Übersichtsteil des Kapitels (Seite 294).

5.2.7 Sonstige Eigenschaften

Im folgenden Sammelsurium treffen Sie hauptsächlich solche Eigenschaften an, die es in äquivalenter Bedeutung bereits beim Formular gibt (siehe Kapitel 4).

Eigenschaft	Bezeichnung	Erläuterung
DateGrouping	Datumsgruppierung	Datumsangaben (0 = amerikanisch, 1 = deutsch)
FastLaserPrinting	Schneller Laserdruck	Siehe *Form*-Objekt
HelpFile	Hilfedatei	Siehe *Form*-Objekt
HelpContextID	Hilfekontext	Siehe *Form*-Objekt
Section(n)	Bereich	Siehe Erläuterung unten
Tag	Marke	Siehe *Form*-Objekt

Section

In Analogie zum *Form*-Objekt werden die einzelnen Bereiche eines Berichts durch die *Section*-Eigenschaft (schreibgeschützt) des *Report*-Objekts dargestellt.

Section kann auch als Objekt (quasi als Control) betrachtet werden. Dies ist kein Widerspruch, denn Eigenschaften können bekanntlich wiederum Objekte sein.

Der Index der *Section*-Eigenschaft ist ein *Integer*-Wert, der einen bestimmten Bereich bezeichnet:

Index	Konstante	Beschreibung
0	*acDetail*	Detailbereich
1	*acHeader*	Berichtskopfbereich
2	*acFooter*	Berichtsfußbereich
3	*acPageHeader*	Seitenkopfbereich
4	*acPageFooter*	Seitenfußbereich
5	*acGroupLevel1Header*	Gruppenebene 1 Kopfbereich
6	*acGroupLevel1Footer*	Gruppenebene 1 Fußbereich
7	*acGroupLevel2Header*	Gruppenebene 2 Kopfbereich
8	*acGroupLevel2Footer*	Gruppenebene 2 Fußbereich

Section hat wiederum zahlreiche Eigenschaften (*BackColor*, *AlternateBackColor*, *Visible*, *DisplayWhen* etc.), die man sich am besten anhand von Beispielen verdeutlicht:

BEISPIEL: Der Fußbereich eines Berichts (oder Formulars) wird ausgeblendet:

```
Me.Section(acFooter).Visible = False
```

BEISPIEL: Die Anweisung:

```
Me.Section(acDetail).AlternateBackColor = vbYellow
```

stellt alternierende Zeilenfarbe für den Detailbereich des Berichts (oder Formulars) ein.

HINWEIS: Auch für Steuerelemente können Sie die *Section*-Eigenschaft verwenden, um zu ermitteln, in welchem Bereich eines Formulars oder Berichts sich das Steuerelement befindet.

BEISPIEL: Die Variable *i* wird auf den Wert 2 gesetzt, falls sich das Textfeld *Text30* im Berichtsfußbereich (oder Formularfußbereich) befindet.

```
Dim i As Integer
i = Me!Text30.Section
```

5.3 Berichtsereignisse

Obwohl das eigentliche Medium eines Berichts das Papier und nicht der Bildschirm ist, gibt es bezüglich der Anzahl der Ereignisse (Events) kaum noch Unterschiede zwischen einem Report und einem Formular[1].

5.3.1 Allgemeine Ereignisse

Die folgende Tabelle zeigt wichtige *Report*-Ereignisse:

Ereignisroutine (Event-Handler)	Ereigniseigenschaft (deutsch)
***Open**(Cancel As Integer)*	Beim Öffnen
***Close**()*	Beim Schließen
***Activate**()*	Bei Aktivierung
***Deactivate**()*	Bei Deaktivierung
***NoData**(Cancel As Integer)*	Bei Ohne Daten
***Page**()*	Bei Seite
***Error**(DataErr As Integer, Response As Integer)*	Bei Fehler
ApplyFilter	Bei angewendetem Filter
Filter	Bei Filter
Load	Bei Laden
Unload	Bei Entladen
GotFocus	Bei Fokuserhalt
LostFocus	Bei Fokusverlust
Resize	Bei Größenänderung
Timer	Bei Zeitgeber

Detailbereich, Kopf- und Fußzeile für die Seite bzw. für einen Bericht haben andere Ereignisse:

Ereignisroutine (Event-Handler)	Ereigniseigenschaft (deutsch)
***Retreat**()*	Bei Rücknahme (nur Detailbereich und Berichtskopf/-fuß)
***Print**(Cancel As Integer, PrintCount As Integer)*	Beim Drucken
***Format**(Cancel As Integer, FormatCount As _ Integer)*	Beim Formatieren
Paint	Beim Anzeigen

[1] Das betrifft vor allem die Preview- und die Layout-Ansicht des Reports.

Open, Close, Activate, Deactivate, Error

Diese Ereignisse sind äquivalent zu den gleichnamigen Formular-Ereignissen. Dies betrifft auch die Reihenfolge, in der sie auftreten (siehe Kapitel 4).

Page, NoData, Print, Format

Das *Page*-Ereignis wird ausgelöst, nachdem Access die Seite eines Berichts zum Drucken formatiert hat, jedoch bevor die Seite gedruckt wird. Dieses Ereignis wird verwendet, um einen Rahmen für die Seite zu erstellen oder um der Seite andere grafische Elemente hinzuzufügen.

Das *NoData*-Ereignis wird vor dem erstmaligen Auftreten des *Page*-Events ausgelöst. Es tritt ein, nachdem Microsoft Access einen Bericht zum Drucken formatiert hat, der keine Daten enthält (leere Datensatzgruppe), aber bevor der Bericht gedruckt wird. Man verwendet dieses Ereignis um zu verhindern, dass ein leerer Bericht gedruckt wird.

Das *Print*-Ereignis tritt auf, wenn Daten in einem Berichtsbereich zum Drucken formatiert werden, jedoch bevor der Bereich gedruckt wird. Bei *Detailbereichen* wird *Print* für jeden Datensatz im Bereich ausgelöst, unmittelbar bevor die Daten gedruckt werden. In Gruppenköpfen/-füßen tritt *Print* bei jeder neuen Gruppe auf. Hier könnten Sie z.B. Seitensummen berechnen, die in die Kopf- oder Fußzeile gedruckt werden.

Das *Format*-Ereignis wird ausgelöst, wenn Access die Daten ermittelt, die in einen Berichtsbereich gehören, jedoch bevor der Bereich für die Vorschau oder für das Drucken formatiert wird. Das *Format*-Ereignis verwenden Sie auch bei Änderungen, die das Seitenlayout betreffen, wie z.B. das Ein-/Ausblenden von Steuerelementen oder wenn Sie auf Daten aus Bereichen zugreifen müssen, die nicht gedruckt werden (z.B. wenn Sie eine fortlaufende Summe berechnen, aber nur einzelne Seiten des Berichts drucken wollen).

BEISPIEL: Das aktuelle Datum wird fett formatiert an den durch *CurrentX* und *CurrentY* festgelegten Koordinaten ausgegeben:

```
Sub ReportHeader0_Print (Cancel As Integer, PrintCount As Integer)
 Me.FontBold = True
 Me.Print Date()
End Sub
```

BEISPIEL: In der Ereignisprozedur *Detailbereich_Print* wird die Farbe des Rechteck-Hintergrundes nach jeder ausgedruckten Zeile geändert:

```
Dim frb As Boolean

Private Sub Detailbereich_Print(Cancel As Integer, PrintCount As Integer)
    If frb Then
        Rechteck.BackColor = vbYellow
    Else
        Rechteck.BackColor = vbWhite
    End If
    frb = Not frb
End Sub
```

5.3.2 Tastatur- und Mausereignisse

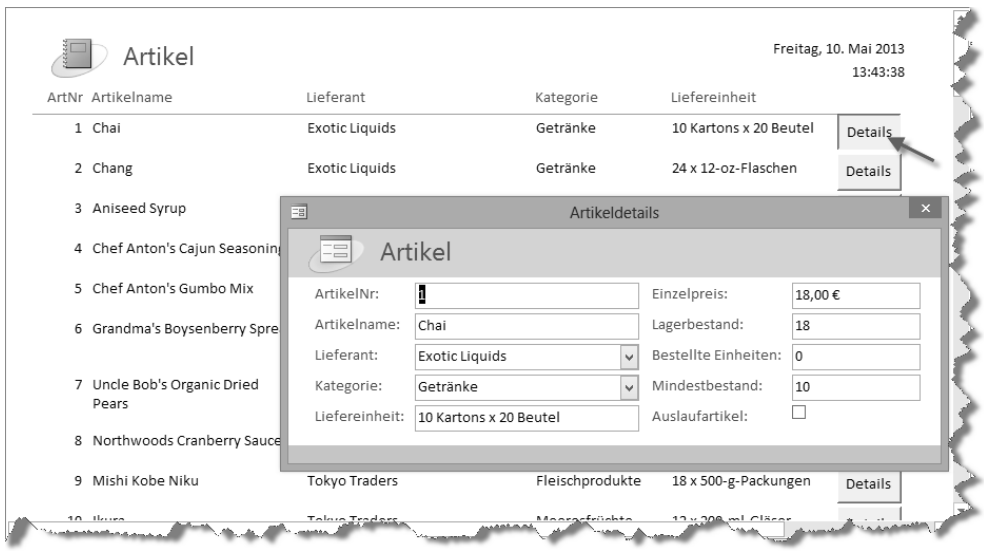
Die meisten der im Folgenden aufgelisteten Ereignisse des *Report*-Objekts treffen nur für die Berichtsansicht (manchmal auch Layoutansicht) zu.

Ereignisroutine (Event-Handler)	Ereigniseigenschaft (deutsch)
KeyDown*(KeyCode As Integer, Shift As Integer)*	Bei Taste Ab
KeyPress*(KeyAscii As Integer)*	Bei Taste
KeyUp*(KeyCode As Integer, Shift As Integer)*	Bei Taste Auf
MouseDown	Bei Maustaste Ab
Click	Beim Klicken
DblClick	Beim Doppelklicken
MouseMove	Bei Mausbewegung
MouseUp	Bei Maustaste Auf
MouseWheel	Bei Mausrad

BEISPIEL: Um das *Click*-Ereignis zu demonstrieren, lassen wir uns mit Assistentenhilfe einen Bericht für die *Artikel*-Tabelle der *Nordwind*-Datenbank generieren und fügen in den Detailbereich ganz rechts eine Befehlsschaltfläche *Befehl1* ein (damit diese Schaltfläche nicht mit ausgedruckt wird, setzen wir deren *Anzeigen*-Eigenschaft auf *Nur am Bildschirm*). Der *Click*-Eventhandler unterscheidet sich nicht von dem eines Formulars:

```
Private Sub Befehl1_Click()
    DoCmd.OpenForm "Artikeldetails", acNormal, , "Artikelnr=" & [ArtikelNr], , acDialog
End Sub
```

Nach Öffnen des Berichts genügt ein Klick auf eine bestimmte Schaltfläche, um das Dialogfeld *Artikeldetails* zu öffnen.



5.4 Berichtsmethoden

Ein wesentlicher Unterschied zwischen Report und Formular ist, dass man auf die Oberfläche eines Reports auch zeichnen kann. Der überwiegende Teil der Methoden eines Reports sind deshalb Grafikmethoden.

5.4.1 Grafikmethoden (Übersicht)

Die im Folgenden erörterten Grafikmethoden werden grundsätzlich nur im *Page*-Ereignis des Reports bzw. im *Print*- bzw. *Format*-Ereignis eines seiner Bereiche ausgeführt.

Methode	Erklärung
Print *Ausgabeliste*	Druckt Text
Line *[[Step](x1, y1)] – [Step](x2, y2)[, [Farbe][, B[F]]]*	Zeichnet Linie oder Rechteck
PSet *[Step](x, y)[, Farbe]*	Setzt einen Punkt
Circle *[Step](x, y), Radius[, [Farbe][, [Startwert] [, [Endwert][, Seitenverhältnis]]]]*	Zeichnet Kreis oder Kreissegment
TextHeight*(Zeichenfolge)/TextWidth(Zeichenfolge)*	Gibt Höhe/Breite einer Textzeichenfolge zurück
Scale *[(x1, y1) – (x2, y2)]*	Definiert Koordinatensystem

5.4.2 Scale

Mit dieser Methode können Sie die umständliche Definition eines Koordinatensystems mit den Eigenschaften *ScaleLeft*, *ScaleTop*, *ScaleWidth* und *ScaleHeight* vermeiden.

Syntax:

```
Scale [(x1, y1) - (x2, y2)]
```

Argumente im Einzelnen:

x1, y1 linke obere Ecke des Koordinatensystems

x2, y2 rechte untere Ecke des Koordinatensystems

BEISPIEL: Die Anweisung:

```
Scale (-10, 120)-(130, -10)
```

führt zum gleichen Koordinatensystem:

```
ScaleWidth = 140
ScaleHeight = 130
ScaleLeft = - 10
ScaleTop = 120
```

HINWEIS: Die *Scale*-Methode ohne Argumente setzt das Koordinatensystem auf die Maßeinheit *Twips* (Standard)!

5.4.3 Line

Diese Methode zieht eine gerade Linie.

Syntax:

```
Line [[Step1](x1, y1)] - [Step2](x2, y2)[, [Farbe][, B[F]]]
```

Die Argumente in der umfangreichen Parameterliste haben folgende Bedeutung:

Step1 Der Parameter gibt an, dass die Koordinaten des Anfangspunkts relativ zur aktuellen Grafikposition liegen, die durch *CurrentX* und *CurrentY* festgelegt ist.

x1, y1 Dies sind *Single*-Werte, welche die Koordinaten des Anfangspunktes der Linie oder der linken oberen Ecke des Rechtecks bezeichnen. Die Eigenschaften *ScaleMode*, *ScaleLeft*, *ScaleTop*, *ScaleHeight* und *ScaleWidth* legen die verwendete Maßeinheit fest. Werden *x1*,*y1* nicht angegeben, so beginnt die Linie an der von *CurrentX CurrentY* definierten Position.

Step2 Gibt an, dass die Endpunktkoordinaten relativ zu *x1*,*y1* liegen.

x2, y2 Dies sind *Single*-Werte für die Endpunktkoordinaten der zu zeichnenden Linie.

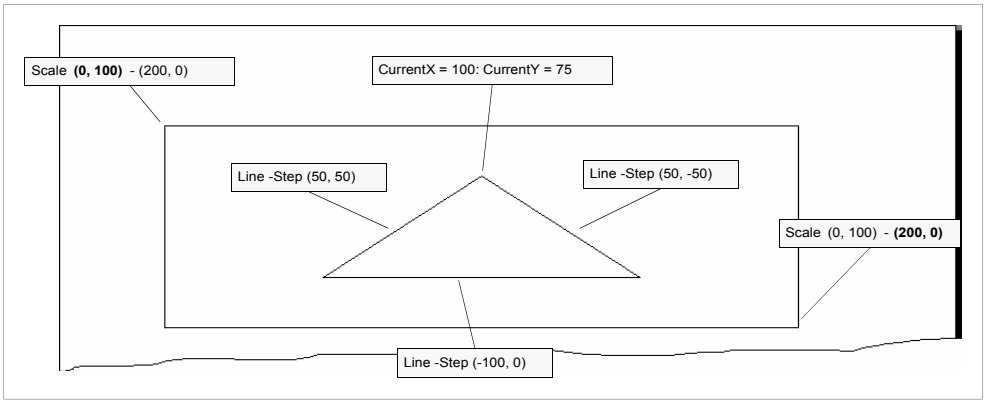
farbe Legt die Farbe für die zu zeichnende Linie fest (*Long*-Wert). Wenn dieses Argument fehlt, wird die *ForeColor*-Eigenschaft verwendet. Sie können für dieses Argument auch direkt die Funktionen *RGB* oder *QBColor* einsetzen.

B Bewirkt, dass ein Rechteck gezeichnet wird. Die angegebenen Koordinaten bestimmen dabei die linke obere und die rechte untere Ecke des Rechtecks.

F Kann nur zusammen mit der Option *B* verwendet werden und gibt an, dass das Rechteck mit der Zeichenfarbe ausgefüllt werden soll. Bei B ohne F wird das Rechteck mit der Farbe und dem Muster ausgefüllt, die durch die Eigenschaften *FillColor* und *FillStyle* festgelegt sind.

BEISPIEL: Der folgende Code im *Print*-Ereignis des Seitenkopfes erzeugt dort ein neues Koordinatensystem (200 breit, 100 hoch, positiv verlaufende x- und y-Achse). Der Seitenkopf wird umrahmt und ein Dreieck in die Mitte gezeichnet.

```
Private Sub Seitenkopf_Print(Cancel As Integer, PrintCount As Integer)
  Scale (0, 100)-(200, 0)              ' neues Koordinatensystem
  Line (0, 100)-(200, 0), , B          ' umrahmendes Rechteck
  CurrentX = 100: CurrentY = 75        ' obere Dreieckspitze
  Line -Step(50, -50)                  ' rechte Dreieckkante
  Line -Step(-100, 0)                  ' untere Dreickkante
  Line -Step(50, 50)                   ' linke Dreieckkante
End Sub
```



BEISPIEL: Die Anweisung:

```
Line (0, 100)-(200, 0), QBColor(12), BF
```

würde ein rot ausgefülltes umrahmendes Rechteck erzeugen.

5.4.4 PSet

Damit setzen Sie einen einzelnen Punkt.

Syntax:

```
PSet [Step](x, y)[, Farbe]
```

Die Argumentliste entspricht im Prinzip der (verkürzten) *Line*-Methode.

BEISPIEL: Umständlich, aber lehrreich: Längs der horizontalen Achse eines Berichts wird mit *PSet* eine rote Linie gezeichnet.

```
Dim mitte As Single, i As Integer
Me.ScaleMode = 3               ' Maßeinheit: Pixel
mitte = Me.ScaleHeight / 2       ' Mittelachse
For = 1 To Me.ScaleWidth
  Me.PSet(i, mitte), QBColor(12)
Next i
```

Bemerkungen

- Die Größe eines Punktes hängt von der Eigenschaft *DrawWidth* ab. Die Art, wie er gezeichnet wird, ist abhängig von den Eigenschaften *DrawMode* und *DrawStyle*.
- Für das Löschen eines Pixels verwenden Sie die Farbe Weiß (&HFFFFFF).

5.4.5 Circle

Damit zeichnen Sie einen Kreis, eine Ellipse oder einen Bogen auf einem *Report*-Objekt.

Syntax:

```
Circle [Step](x, y), Radius[, [Farbe][, [Startwert][, [Endwert][, Seitenverhältnis]]]]
```

Die Bedeutung der Argumente im Einzelnen:

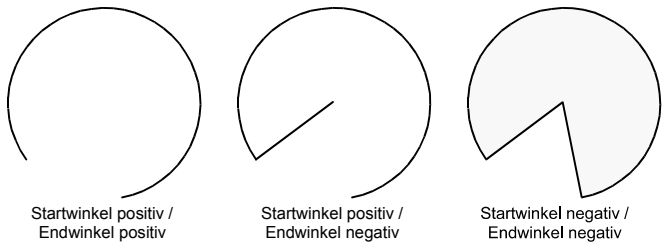
Step	Gibt an, dass der Mittelpunkt relativ zu den aktuellen Koordinaten liegt
x, y	Definiert die Koordinaten des Mittelpunktes (*Single*-Werte)
radius	Radius (*Single*-Wert)
farbe	Parameter, der die Farbe der Kreislinie angibt (*Long*-Wert). Fehlt dieses Argument, so wird *ForeColor* genommen.
startwert, *endwert*	Für einen Kreis- oder Ellipsenbogen geben Start- und Endwert die Anfangs- und die Endposition des Bogens an (in Radiant). Standardmäßiger Startwert ist 0 Radiant, standardmäßiger Endwert ist 2 Pi Radiant (Vollkreis).
seitenverhältnis	Das Seitenverhältnis der Ellipse, Standardwert ist 1.0 (Kreis).

Bemerkungen

- Um den Start- bzw. Endwert in Grad angeben zu können, muss jeweils mit *Pi/180* multipliziert werden.

```
Const Pi = 3.1416
Dim rad As Single, grad As Single
...
rad = grad * Pi / 180
```

- Die Winkelangaben zählen immer entgegen der Uhrzeigerrichtung.



- Sind Start- bzw. Endwert negativ, so wird der jeweilige Radius mit eingezeichnet (Tortenstück).
- Nach Beendigung der *Circle*-Methode sind *CurrentX* und *CurrentY* auf die Mittelpunktskoordinaten eingestellt.
- Linienart und -breite, Füllfarbe und Füllmuster sind abhängig von den Eigenschaften *DrawMode*, *DrawStyle*, *DrawWidth*, *FillColor* und *FillStyle*.

BEISPIEL: Ein Tortendiagramm soll genau in der Mitte einer Berichtsseite erzeugt werden.

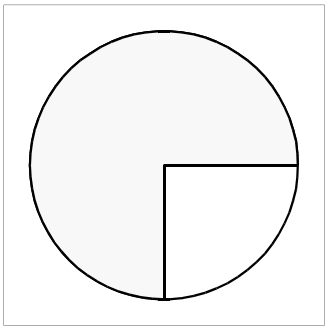
Im ersten Schritt zeichnen wir einen (leeren) Vollkreis:

```
Const Pi = 3.14159
Dim mitteX As Single, mitteY As Single, radius As Single
mitteX = Me.ScaleWidth / 2
mitteY = Me.ScaleHeight / 2
radius = Me.ScaleHeight / 4
Me.Circle (mitteX, mitteY), radius
```

Im nächsten Schritt wird die gelb gefüllte "Dreivierteltorte" erzeugt. Dazu müssen Start- und Endwinkel negative Vorzeichen haben.

```
Dim start As Single, ende As Single
start = -0.00000001
ende = -3 * Pi / 2                    ' 270 Grad
Me.FillColor = QBColor(14)            ' gelb
Me.FillStyle = 0                      ' solide Füllung
Me.Circle (mitteX, mitteY), radius, , start, ende
```

Wenn wir diesen Code in das *Page*-Event des Reports einbauen, erscheint das Kreisdiagramm sehr groß (Durchmesser entspricht der Hälfte der Seitenhöhe, ohne Ränder). Wir können den gleichen Code aber auch in das *Print*-Event des Detailbereichs einsetzen. Dann beziehen sich die *Scale*-Eigenschaften auf die von uns eingestellte Höhe und Breite des Detailbereichs, das Kreisdiagramm wird erheblich kleiner ausfallen.



5.4.6 Print

Auch die Textausgabe zählt unter Windows mit zu den Grafikmethoden! Die *Print*-Methode gibt eine Zeichenkette (String) oder auch nur den Wert von Variablen bzw. Funktionen auf einem *Report*-Objekt (oder im *Direkt*-Fenster) aus. Dabei werden die aktuelle Farbe (*ForeColor*) und Schriftart (*Font*-Eigenschaften) verwendet.

Syntax:

```
Print {Spc(n) | Tab(n)} ausdruck zeichPos
```

Die Argumente:

Spc(n)	Wird optional verwendet, um n Leerzeichen auszugeben.
Tab(n)	Wird optional verwendet, um die Druckposition zur n-ten Spaltennummer zu bewegen. Tab ohne Argumente verschiebt die Druckposition zum Anfang des nächsten Ausgabebereichs.
ausdruck	Wird optional ausgedruckt (numerischer Ausdruck oder String)
zeichPos	Legt die Druckposition für das nächste Zeichen fest. Mehrere Ausdrücke können mit einem Semikolon bzw. einem Leerzeichen getrennt werden.

BEISPIEL: Die folgenden zwei Programmzeilen liefern ähnliche (aber nicht identische) Ergebnisse.

```
Print "Hallo"; Tab(50); "Hallo"

Print "Hallo"; Space(50); "Hallo"
```

5.4.7 TextWidth und TextHeight

Beide Methoden geben die Breite bzw. Höhe einer Textzeichenfolge zurück.

Syntax:

```
TextHeight(Zeichenfolge)
```

bzw.

```
TextWidth(Zeichenfolge)
```

Mit *TextWidth/TextHeight* werden die horizontale bzw. vertikale Ausdehnung einer Textzeichenfolge in der mit *Scale* festgelegten Maßeinteilung ermittelt, wenn der Bericht formatiert und gedruckt wird. Das Ergebnis ist abhängig von den Schriftarteigenschaften (*FontName*, *FontSize* etc.). Sind Wagenrücklaufzeichen enthalten, so ermittelt *TextWidth* die Breite der längsten Zeile (vom Zeilenanfang bis zum Wagenrücklaufzeichen). *TextHeight* gibt in diesem Fall die kumulierte Höhe der Zeilen zurück, inklusive des normalen Abstandes über und unter jeder Zeile.

Sie können die von den Methoden *TextWidth* und *TextHeight* zurückgegebenen Werte verwenden, um den erforderlichen Platz für mehrere Textzeilen innerhalb eines Berichts zu berechnen.

BEISPIEL: Der Event-Handler druckt die Überschrift eines Berichtes 20 Pixel unterhalb des oberen Randes zentriert aus:

```
Private Sub Report_Page()
  Dim schrift As String
  schrift = "Produktbericht"
  Me.ScaleMode = 3              ' Pixel
```