

Einführung in die prozedurale Programmierung mit Delphi

Lektion 3
März 2010

„Theoretische Grundlagen“

1. Vorschläge zur Wahl der Variablennamen
2. Datentypumwandlungen (Auswahl)
3. Delphi-Objekte (Auswahl)
4. Mathematische Operatoren und Funktionen (Auswahl)
5. Grundstrukturen (Sequenz, Auswahl, Zyklus)
6. Datentypen

1. Vorschläge zur Wahl der Variablennamen

- Variablennamen sollen möglichst selbsterklärend sein
- Namen von Objekten sollen am Anfang des Namens einen entsprechenden Kennbuchstaben tragen, z.B. ein Edit-Objekt-Name beginnt immer mit e, ein Button-Objekt-Name immer mit b usw.

<i>Objekt</i>	<i>Bezeichner-Beispiel</i>
Programmname	meinname
Projektname	pmeinname
Hauptformular	fmeinname
Unterformulare	fnoten, fname, fxyz,
Units	unoten, uname, ...
Buttons	bname, bschule, ...
Editorfelder	ename, enote, ezahl, ...
Label	lname, lxyz, ...
ListBox	lnoten, lbwerte, ...
Tabellen	tnoten, twerte

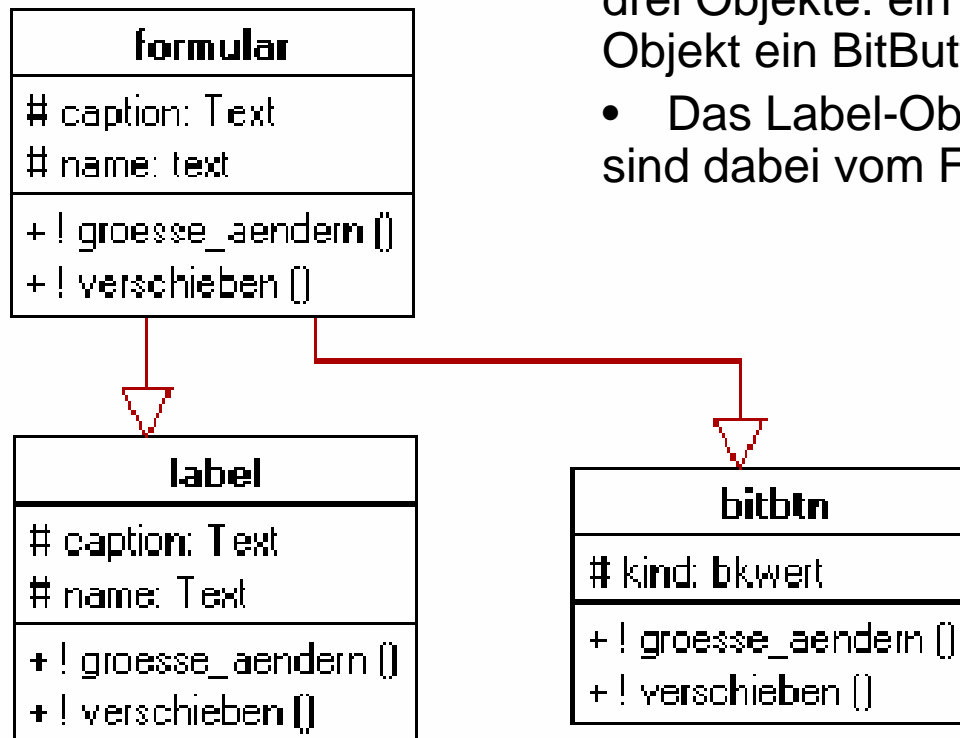
2. Datentypumwandlungen (Auswahl)

- Bei der Übernahme von Daten aus der Texteigenschaft z.B. eines Edit-Objekts müssen diese bei Bedarf in eine Zahl umgewandelt werden, damit mit diesen Daten gerechnet werden kann
- Das Gleiche gilt bei der Ausgabe von Daten: Berechnete Werte müssen vor der Übertragung an die Texteigenschaft eines Edit-Objekts wieder in einen Text umgewandelt werden
- Das alles erfolgt mittels Umwandlungsfunktionen (siehe Tabelle)

Function	Umwandlung	Umwandlungsbeispiel
IntToStr	GanzZahl in Zeichenkette	<code>eausgabe.text:=IntToStr(zahl);</code>
StrToInt	Zeichenkette in GanzZahl	<code>zahl:=StrToInt(eausgabe.text);</code>
FloatToStr	KommaZahl in Zeichenkette	<code>ezinsen.text:=FloatToStr(zinsen);</code>
StrToFloat	Zeichenkette in KommaZahl	<code>zinsen:=StrToFloat(ezinsen.text);</code>
FloatToStrF	KommaZahl in formatierte Zeichenkette	<code>ezinsen.text:=FloatToStrF(zinsen,ffnumber,8,2)</code>

3. Delphi-Objekte (Auswahl) (I)

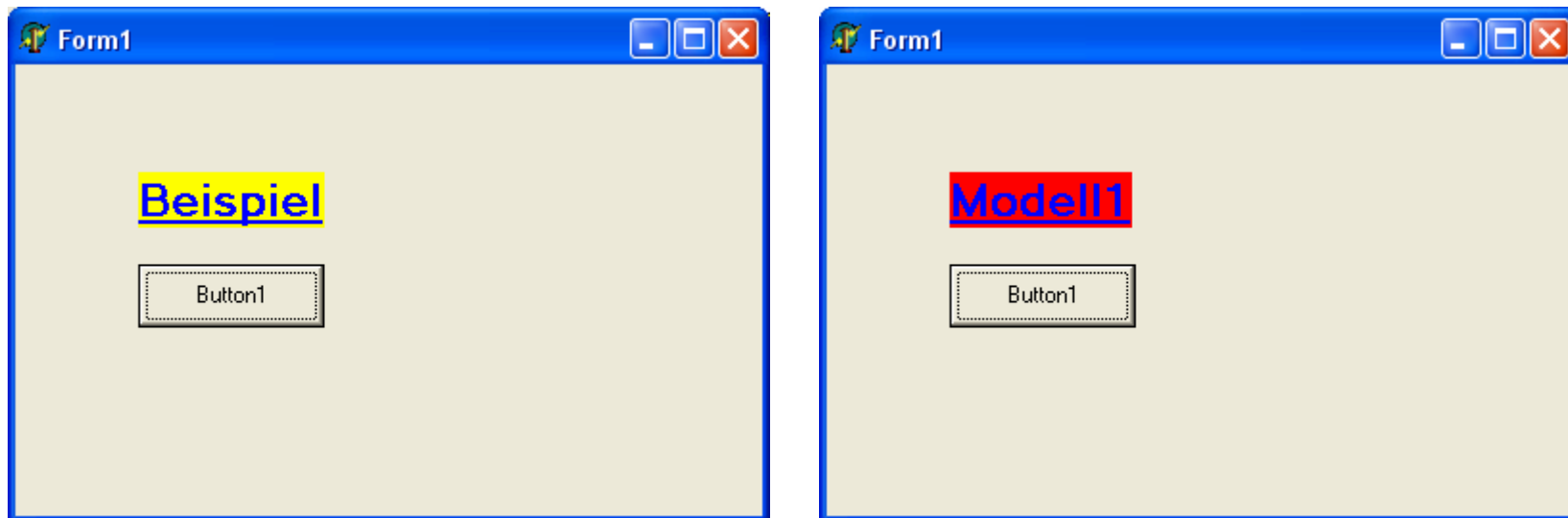
- Das Programm „Hello World“ enthält genau drei Objekte: ein Formularobjekt, ein Label-Objekt ein BitButton-Objekt
- Das Label-Objekt und das BitButton-Objekt sind dabei vom Formularobjekt abgeleitet.



3. Delphi-Objekte (Auswahl) (II)

Mittels Objektinspektor können (fast) alle Eigenschaften eines Objekts direkt definiert werden. Diese Einstellungen sind allerdings statisch. Sollen die Eigenschaften zur Laufzeit des Programms geändert werden, können diese mittels Punktnotation im Quelltext dynamisch verändert werden.

3. Delphi-Objekte (Auswahl) (III)



Beim Klick auf den Button werden die Eigenschaften des Labels dynamisch verändert. Dafür werden folgende Punktnotationen der Objekteigenschaften benötigt:

- `lbeispiel.Caption:='Modell1';`
- `lbeispiel.Color:=clred;`

4. Mathematische Operatoren und Funktionen (Auswahl) (I)

Operator	Operation	Beispiel	Operandentyp	Ergebnistyp
+	Addition	x:=a + b	Integer	Integer
			Real	Real
-	Subtraktion	x:=a - b	Integer	Integer
			Real	Real
*	Multiplikation	x:=a * b	Integer	Integer
			Real	Real
/	reelle Division	x:=a / b	Integer	Real
			Real	Real
DIV	Integer-Division	x:=a div b	Integer	Integer
MOD	Divisionsrest	x:=a mod b Bsp.: x:=6 mod 4	Integer	Integer x = 2

4. Mathematische Operatoren und Funktionen (Auswahl) (II)

Operator	Operation	Beispiel	Operandentyp	Ergebnistyp
SQRT	Wurzelziehen	x:=SQRT(a)	Integer	Real
			Real	Real
SQR	Quadrieren	x:=SQR(a)	Integer	Integer
			Real	Real
SIN	Sinusfunktion	x:=SIN(a) a im Bogenmaß	Integer	Real
			Real	Real
COS	Kosinusfunktion	x:=COS(a) a im Bogenmaß	Integer	Real
			Real	Real
TAN	Tangensfunktion	x:=TAN(a) a im Bogenmaß	Integer	Real
			Real	Real
ARCTAN	Arkustangensfunktion	x:=ARCTAN(a)	Integer	Real
			Real	Real
EXP	Eulersche Funktion	x:=EXP(a)	Integer	Real
			Real	Real
LN	Natürlicher Log.	x:=LN(a)	Integer	Real
			Real	Real

4. Mathematische Operatoren und Funktionen (Auswahl) (III)

Operator	Operation	Beispiel	Operandentyp	Ergebnistyp
ABS	Betragsfunktion	x:=ABS(a)	Integer Real	Real Real
PI	ZahlPi	x:=PI	Real-Konstante	3,14159...
FRAC	Nachkomma- stellen	x:=FRAC(a) Bsp.:a:=12.345	Real	Real x = 345
INT	rundet ab	x:=INT(a) Bsp.:a:=12.98	Real	Real x = 12
TRUNC	rundet ab	x:=TRUNC(a)	Integer/Real	Integer
ROUND	rundet mathemat.	x:=ROUND(a)	Integer/Real	Integer

5. Grundstrukturen (Sequenz)

Dreiecksfläche

Grundstruktur Sequenz, Anweisungen werden in gegebener Reihenfolge abgearbeitet

E: Seite a

E: Seite b

E: Seite c

$s \leftarrow (a+b+c)/2$

$f \leftarrow \text{Wurzel}(s(s-a)(s-b)(s-c))$

A: f

Übersetzung in PASCAL (Delphi)

//Eingabe

`a:=STRTOFLOAT(eseitea.Text);`

`b:=STRTOFLOAT(eseiteb.Text);`

`c:=STRTOFLOAT(eseitec.Text);`

//Verarbeitung – Flächenberechnung mit Heron

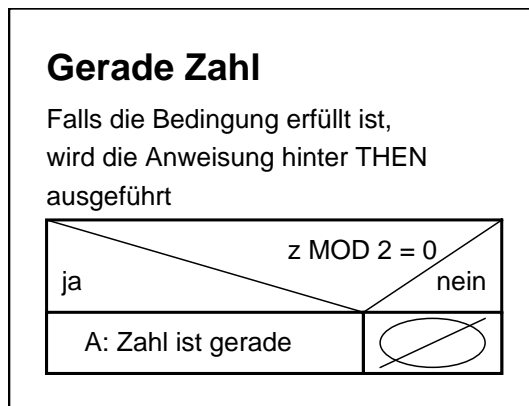
`s:=(a+b+c)/2;`

`f:=SQRT(s*(s-a)*(s-b)*(s-c));`

//Ausgabe

`eflaeche.Text:=FLOATTOSTRF(f,ffnumber,8,2);`

5. Grundstrukturen (Auswahl - einseitig)



Übersetzung in PASCAL (Delphi)

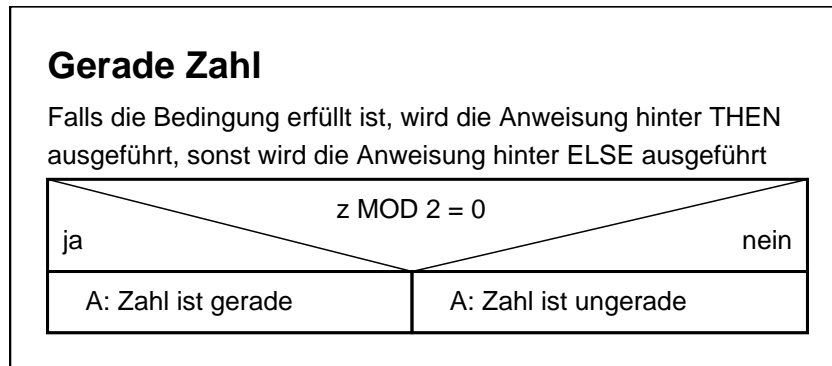
//Eingabe

z:=STRTOINT(ezahl.Text);

//Verarbeitung und Ausgabe

IF z MOD 2 = 0 THEN eausgabe.Text:='Zahl ist gerade'

5. Grundstrukturen (Auswahl - zweiseitig)



Übersetzung in PASCAL (Delphi)

```
//Eingabe  
z:=STRTOINT(ezahl.Text);  
//Verarbeitung und Ausgabe  
IF z MOD 2 = 0 THEN eausgabe.Text:='Zahl ist gerade'  
    ELSE eausgabe.Text:='Zahl ist ungerade'
```

5. Grundstrukturen (Auswahl - mehrseitig)

Übersetzung in PASCAL (Delphi)

```
//Eingabe
z:=STRTOINT(ezahl.Text);
//Verarbeitung
FOR n:=1 TO 10 DO
  IF z MOD n = 0 THEN
    CASE n OF
      1 : label5.caption:='teilbar durch 1';
      2 : label5.caption:=label5.caption+' und durch 2';
      3 : label5.caption:=label5.caption+' und durch 3';
      4 : label5.caption:=label5.caption+' und durch 4';
      5 : label5.caption:=label5.caption+' und durch 5';
      6 : label5.caption:=label5.caption+' und durch 6';
      7 : label5.caption:=label5.caption+' und durch 7';
      8 : label5.caption:=label5.caption+' und durch 8';
      9 : label5.caption:=label5.caption+' und durch 9';
      10 : label5.caption:=label5.caption+' und durch 10';
    END;
```

Teilbarkeit

Falls $n=1$ ist, dann gebe aus: Teilbar durch 1
 Falls $n=2$ ist, dann gebe aus: (...) und durch 2
 usw.

(...) enthält Vorgängertext, falls bereits andere Teilbarkeiten erkannt wurden

Hinweis: Aus Platzgründen werden hier nur 4 Fälle angezeigt, im Beispiel nebenan treten 10 Fälle auf.

1	2	3	4	n	sonst
A: teilbar durch 1	A: (...) teilbar durch 2	A: (...) teilbar durch 3	A: (...) teilbar durch 4		

5. Grundstrukturen (Zyklus - Zählschleife)

Übersetzung in PASCAL (Delphi)

Kontoentwicklung

Wiederhole für den Zähler i von 1 bis zum Ende der Laufzeit t die Anweisungen zwischen BEGIN und END (Verbundanweisung)

Hinweis: Die Zählschleife wird genau t mal abgearbeitet

Wiederhole für i von 1 bis t

$z \leftarrow k * zs / 100;$

$k \leftarrow k + z;$

A: Jahresnummer, Kapital

//Eingaben

$k := \text{STRTOFLOAT}(\text{ekapital.text});$

$zs := \text{STRTOFLOAT}(\text{ezinssatz.text});$

$t := \text{STRTOINT}(\text{elaufzeit.text});$

//Verarbeiten und Ausgeben

FOR $i := 1$ TO t DO

BEGIN

$z := k * zs / 100;$

$k := k + z;$

$\text{listbox1.Items.Add}(\text{INTTOSTR}(i) + '. ' + \text{FLOATTOSTR}(k));$

END;

5. Grundstrukturen (Zyklus - anfangsgeprüft)

Zahlenfolge

Solange die Bedingung $i \leq 20$ erfüllt ist, werden die Anweisungen im Verbund ausgeführt

Solange $i \leq 20$ tue

$a \leftarrow 1 + 1/i$

A: Nr., Zahlenfolgenglied

$i \leftarrow i + 1$

Übersetzung in PASCAL (Delphi)

$i := 1;$

WHILE $i \leq 20$ **DO**

BEGIN

$a := 1 + 1/i;$

`listbox1.Items.Add(INTTOSTR(i) + '.' + FLOATTOSTR(a));`

$i := i + 1;$

END;

5. Grundstrukturen (Zyklus - endgeprüft)

Zahlenfolge

Wiederhole die Anweisungen im Verbund bis die Bedingung $i > 20$ erfüllt ist.

$a \leftarrow 1 + 1/i$

A: Nr., Zahlenfolgenglied

$i \leftarrow i + 1$

Wiederhole bis $i > 20$

Übersetzung in PASCAL (Delphi)

```
i:=1;
```

```
REPEAT
```

```
  a:=1+1/i;
```

```
  listBox1.Items.Add(INTTOSTR(i)+'.'+FLOATTOSTR(a));
```

```
  i:=i+1;
```

```
UNTIL i>20;
```

6. Datentypen (I)

einfache Datentypen		Zeichenketten- datentypen	strukturierte Datentypen	Zeiger- typen
<i>ordinale</i>	<i>reelle</i>			
INTEGER	REAL	STRING	ARRAY	POINTER
SHORTINT	REAL48	STRING[n]	RECORD	^ZeigerTyp
SMALLINT	SINGLE			
LONGINT	DOUBLE	SHORTSTRING	SET OF	
INT64	EXTENDED	ANSISTRING		
BYTE	COMP	WIDESTRING	DATEI	
WORD	CURRENCY		OBJECT	
LONGWORD		CHAR	FILE	
BOOLEAN			FILE OF Type	
		ANSICHAR	TEXT	
CHAR		WIDECHAR		
TeilbereichsTyp				
AufzählungsTyp				
w eitere w ichtige Hinw eise siehe DELPHI-Hilf e				

6. Datentypen (II)

Wertebereiche - Auswahl			
Typ	Bereich	Speicherbelastung	Genauigkeit
INTEGER	-2147483648..2147483647	4 Byte	mit Vorzeichen
SHORTINT	-128...127	1 Byte	mit Vorzeichen
SMALLINT	-32768...32767	2 Byte	mit Vorzeichen
LONGINT	-2147483648...2147483647	4 Byte	mit Vorzeichen
INT64	$-2^{63}..2^{63} - 1$	8 Byte	mit Vorzeichen
BYTE	0...255	1 Byte	ohne Vorzeichen
WORD	0...65535	2 Byte	ohne Vorzeichen
LONGWORD	0..4294967295	4 Byte	ohne Vorzeichen
REAL	$-1,7 \cdot 10^{308} \dots +1,7 \cdot 10^{308}$; kleinster Wert: $5,0 \cdot 10^{-324}$	8 Byte	15-16 Stellen
REAL48	$-1,7 \cdot 10^{38} \dots +1,7 \cdot 10^{38}$; kleinster Wert: $2,9 \cdot 10^{-39}$	6 Byte	11-12 Stellen
SINGLE*	$-3,4 \cdot 10^{38} \dots +3,4 \cdot 10^{38}$; kleinster Wert: $1,5 \cdot 10^{-45}$	4 Byte	7-8 Stellen
DOUBLE*	$-1,7 \cdot 10^{308} \dots +1,7 \cdot 10^{308}$; kleinster Wert: $5,0 \cdot 10^{-324}$	8 Byte	15-16 Stellen
EXTENDED*	$-1,1 \cdot 10^{4932} \dots +1,1 \cdot 10^{4932}$; kleinster Wert: $3,4 \cdot 10^{-4932}$	10 Byte	19-20 Stellen
BOOLEAN	TRUE, FALSE	1Byte	
CHAR	ASCII-Zeichen	1Byte	