

Algorithmus

Ein **Algorithmus** (auch Lösungsverfahren) ist eine formale Handlungsvorschrift zur Lösung eines Problems oder einer bestimmten Art von Problemen in endlich vielen Schritten. Algorithmen sind eines der zentralen Themen der Informatik und Mathematik (wikipedia.org)

Beispiel: Verschlüsseln eines Textes mit Grafik

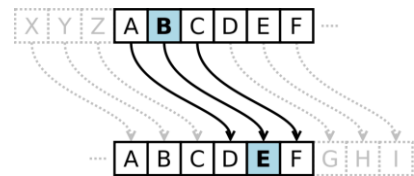


Entschlüssele diesen Text!



Beispiel: Verschlüsseln eines Textes mit Caesar

Zum Zwecke der Verschlüsselung wird dabei jeder Buchstabe des lateinischen Standardalphabets um eine bestimmte Anzahl von Positionen zyklisch verschoben (rotiert).



s	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
k	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Stelle	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

s = Klartext k = Schlüssel (k=3)

HIXRPRO = ???

Beispiel: GTIN/EAN Code (Strichcode)



Die **Prüfziffer** stellt sicher, dass bei der Eingabe keine Fehler gemacht wurden, sie wird nach einem festen mathematischen System berechnet: Die (von links beginnend) ersten 12 Ziffern werden abwechselnd mit 1 bzw. 3 multipliziert und dann addiert. Die Ziffer auf der Einerstelle des Ergebnisses wird von 10 subtrahiert und als Prüfziffer an der letzten Stelle des Codesymbols eingebaut.

EAN	5	2	0	4	7	6	0	0	0	0	9	1	?
Faktor	1	3	1	3	1	3	1	3	1	3	1	3	Pz

EAN	7	3	1	4	8	1	0	2	2	0	3	9	?
Faktor	1	3	1	3	1	3	1	3	1	3	1	3	Pz

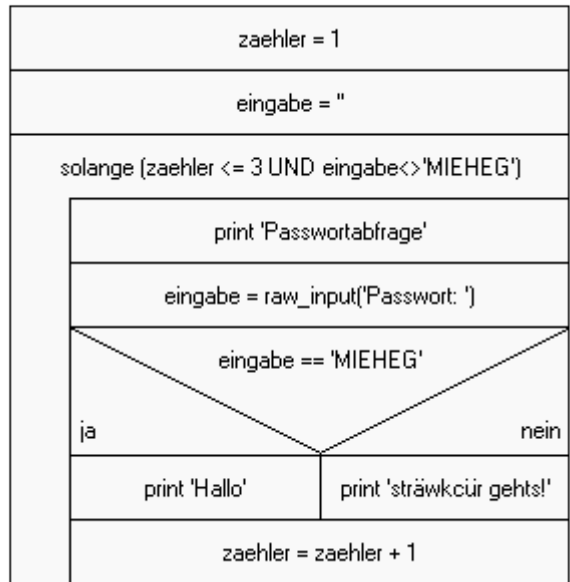
Beispiel: Computerprogramm

Quelltext in DELPHI

```

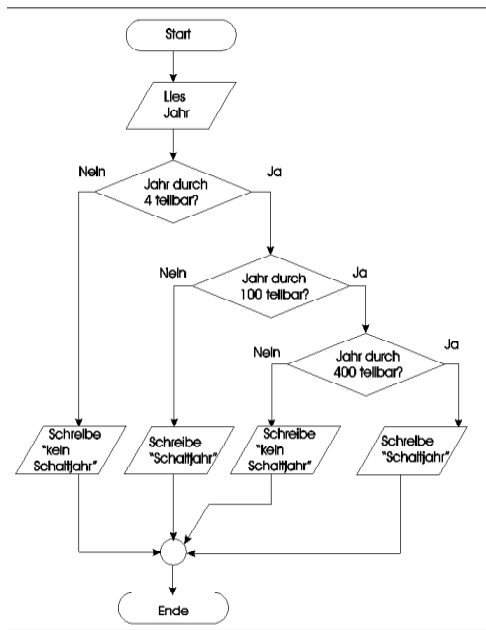
Procedure quadrGl(a,b,c:real;var Anz:integer;var
x1,x2:real);
  var d,q:real;
begin
  d:=b*b-4*a*c;
  if d < 0 then Anz:=0 else
  if d=0 then Begin
    Anz:=1;
    x1:=-b/2/a;
  End else Begin
    Anz:=2;
    q:=sqrt(d);
    x1:=(-b+q)/2/a;
    x2:=(-b-q)/2/a;
  End;
end;
    
```

Was kann dieser Algorithmus?

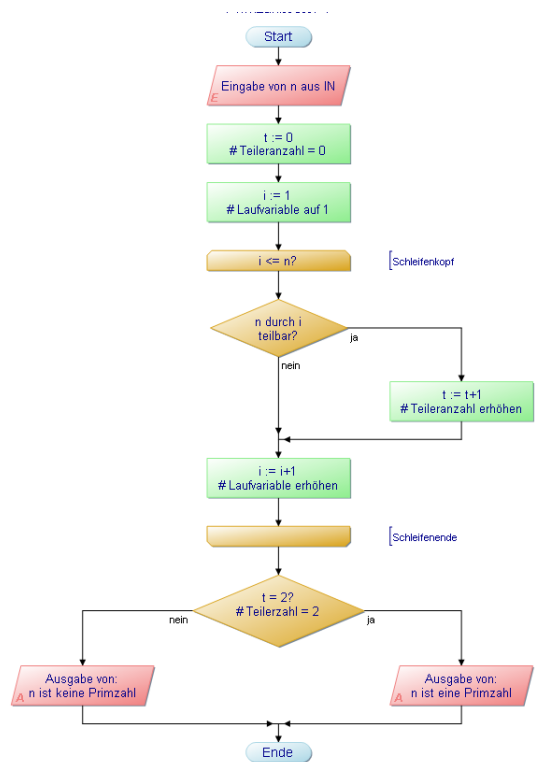


Was kann dieser Algorithmus?

Programmablaufpläne



Was kann dieser Algorithmus?



Was kann dieser Algorithmus?

Programmiersprachen- Welche gibt es?

-
-

-
-

1. Grundlagen der Objektorientierung mit DELPHI

1.1. Die Programmiersprache DELPHI

Delphi ist eine Version der weit verbreiteten Programmiersprache PASCAL. Sie enthält Konzepte, die auch in anderen Programmiersprachen vorkommen. Mit ihr können relativ leicht Programme geschrieben werden, die genauso bedient werden wie das Anwenderprogramm WORD. Delphi beinhaltet das Konzept der Objekt Orientierung.

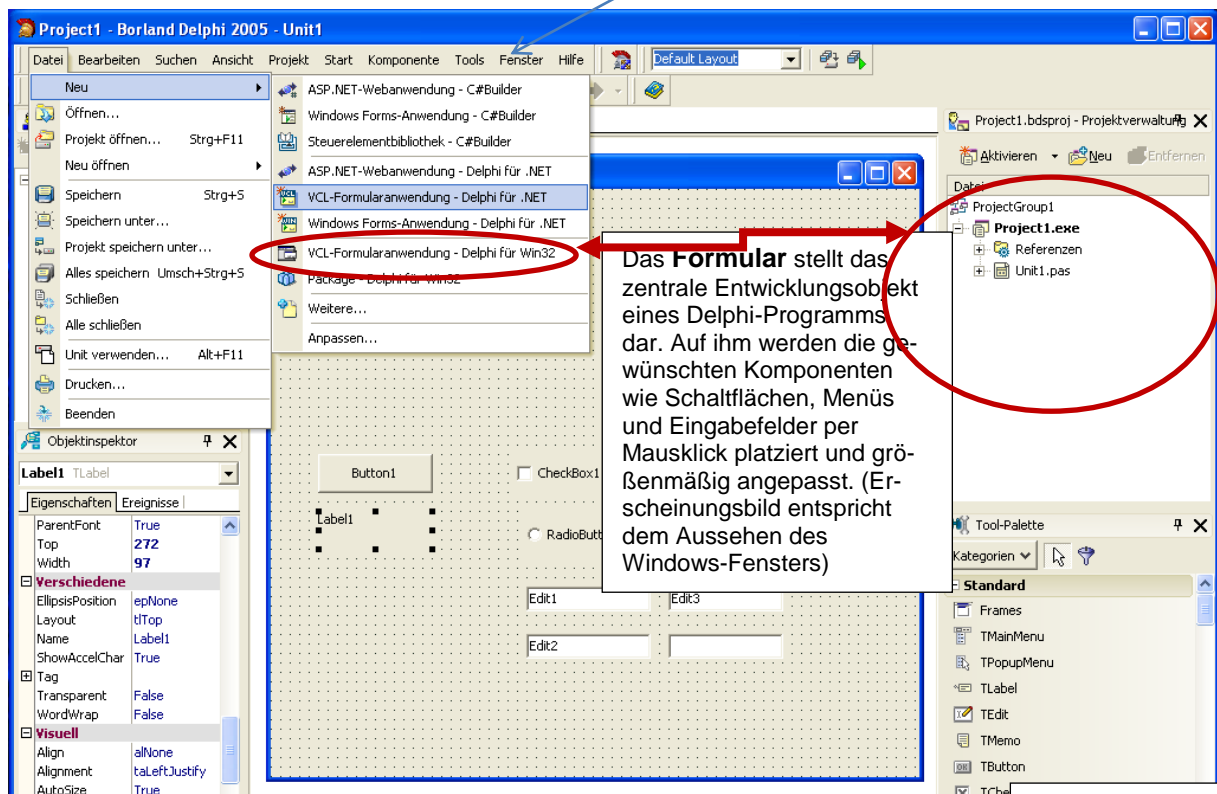
Hinter den visuellen Komponenten verbergen sich nicht nur grafische Darstellungen. Vielmehr stellt jede Komponente dem Programm eine oder mehrere Funktionen zur Verfügung.

Die Erarbeitung eines Programms erfolgt entsprechend der Methode der schrittweisen Verfeinerung (algorithmische Seite). Gleichzeitig gilt es, die Benutzeroberfläche und die Eigenschaften der Objekte festzulegen (visuelle Seite)

1.2. Die Entwicklungsumgebung

Diese Arbeitsfläche wird als Integrierte Entwicklungsumgebung, kurz IDE (engl. Integrated Development Enviroment) bezeichnet. Alle Fenster der IDE können frei und einzeln auf dem Bildschirm positioniert oder geschlossen werden. Durch das Schließen des Hauptfensters wird Delphi beendet.

Menüführung



Das **Formular** stellt das zentrale Entwicklungsobjekt eines Delphi-Programms dar. Auf ihm werden die gewünschten Komponenten wie Schaltflächen, Menüs und Eingabefelder per Mausclick platziert und größtmäßig angepasst. (Erscheinungsbild entspricht dem Aussehen des Windows-Fensters)

Umschalten zwischen Formulardesign und Quellcode

Die **Tool-Palette** ist in verschiedene Register (Standard, Zusätzlich usw.) unterteilt, und diese erlauben die Auswahl der benötigten Windows-Komponenten.

Mit Hilfe des **Objektinspektors** werden Darstellungsweise und Verhalten der Objekte (Komponenten) in einem Formular sowie das Aussehen des Formulars selbst festgelegt. Das Erscheinungsbild wird über die Seite "Eigenschaften", das Verhalten beim Eintritt eines Ereignisses über die Seite "Ereignisse" eingestellt.

```

1: 1  Einfügen  Geändert  Code / Design
-----
IMIS - Dozent  Delphi2005  Klasse 9  Project1 - Borland De...

32  end;
33
34  var
35  Form1: TForm1;
36  gp,rab,rp, ep:real;
37
38  implementation
39
40  {$R *.dfm}
41
42  procedure TForm1.Rabattberechnung(Sender: TObject);
43  begin
44  gp:=StrToFloat(edit1.Text)*StrToFloat(edit2.Text);
45  edit3.Text:=FloatToStrf(gp,ffixed,10,2);
46  if StrToFloat(edit1.Text)<100 then rab:=0;
47  if (StrToFloat(edit1.Text)>=100) and (StrToFloat(ed
48  if (StrToFloat(edit1.Text)>=200) and (StrToFloat(ed
49  if (StrToFloat(edit1.Text)>=500) then rab:=10;
50  edit4.Text:= FloatToStrf(rab,ffixed,10,0);
51  edit5.Text:=FloatToStrf(rab*gp/100,ffixed,10,2);
52  ep:=gp-StrToFloat(edit5.Text);
53  edit6.Text:= FloatToStrf(ep,ffcurrency,10,2);

```

Der **Quelltexteditor** wird zum Schreiben des Programmcodes in der Programmiersprache genutzt. Dem Programmierer obliegt daher die Einarbeitung der Algorithmen zur Ereignisbehandlung.

1.3. Projektverwaltung

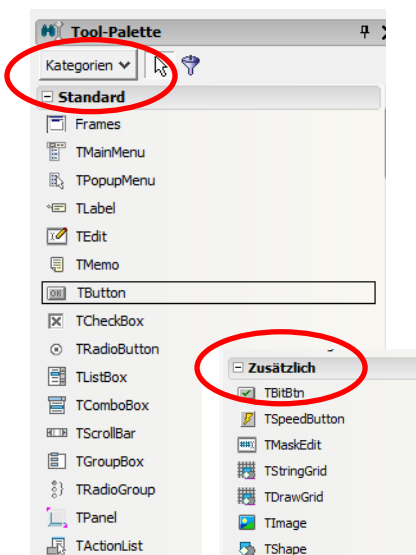
Ein Delphi-Projekt ist eine Sammlung aller Dateien, die zusammen eine Delphi-Anwendung auf dem Entwicklungssystem ergeben. Einige dieser Dateien werden im Entwurfsmodus erzeugt, andere werden bei der Compilierung des Projekt-Quelltextes angelegt.

Jedes Projekt sollte unbedingt in einem separaten Verzeichnis gespeichert werden.

1.4. Ausführen und Testen von Programmen

1. Datei → Neu → VCL Formularanwendung für WIN32
2. Layout erstellen (Objekte anordnen)
3. **Projekt speichern** unter `H:\.....\DELPHI\Übungs Nr\Name.pas`
`H:\.....\DELPHI\Übungs Nr\Name.bdsproj`
4. Testen (F9)
5. Zuordnung von Ereignissen zu vorhandenen Objekten (Ereignisname darf keine Umlaute und Leerzeichen enthalten, darf nicht den Namen von Objekten oder Variablen haben)
6. Alles wiederum **Speichern**
7. Testen
8. Drucken (bitte im Code //Name angeben!!!!)

1.5. Tool-Palette (Auswahl)



TLabel (Text, den der Benutzer weder markieren noch bearbeiten kann, z.B. Titel oder Beschriftungen)

TEdit (Ein Eingabebereich, in dem der Benutzer einzeiligen Text eingeben und bearbeiten kann.)

TButton (Eine Schaltfläche, mit der der Benutzer eine Aktion auslösen kann.)

TCheckBox (Eine Option, die der Benutzer zwischen Ja/Nein oder Wahr/Falsch umschalten kann.)

TRadioButton (Eine Option, die der Benutzer zwischen Ja/Nein oder Wahr/Falsch umschalten kann.)

TImage (Eine Komponente zur Anzeige von Bitmaps, Symbolen oder Metadateien.)

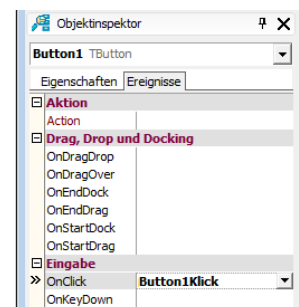
Nach Auswahl der Komponente legt man die Lage im Formular fest- danach fixiert man im Objektivector den Namen und die äußeren Merkmale!

1.6. Reaktion auf Ereignisse

Die Programmierung mit Delphi wird auch als ereignisgesteuerte Programmierung bezeichnet. Ein Ereignis ist eine Aktion, die den Programmablauf beeinflusst. Die Auswahl des Ereignisses erfolgt im Objektivector. Die Notation der jeweiligen Anweisungen erfolgt im Quelltexteditor in einer procedure.

Beispiele:

- Ausführung bei Programmstart (OnActivate)
- Klicken auf eine Schaltfläche (OnClick)
- Doppelklicken (OnDblClick)
- Drücken einer Taste beliebigen Taste (OnKeyPress) oder speziell [RETURN] oder [ESC]
- Bewegen der Maus (OnMouseDown, OnMouseUp, OnMouseMove)



Übung 1

Zur Erfassung der grundlegenden Zusammenhänge gestalten wir auf dem leeren Formular unterschiedliche Komponenten:

- 1.) Benennen Sie das Formular über die Eigenschaft **Caption**: Komponententest
- 2.) Gestalten Sie auf dem Formular nacheinander folgende Komponenten:

- Label
- Edit (mit und ohne Vorgabetext)
- Button
- CheckBox
- RadioButton
- Image

- 3.) Speichern Sie das Projekt unter
H:\.....\DELPHI\Übung1\Unit1.pas
H:\.....\DELPHI\Übung1\Project1.bdsproj
- 4.) Starten Sie das Programm mit F9 oder über das Menü Start!
- 5.) Analysieren Sie die entstandene UNIT , vor allem im Bereich TYPE!
- 6.) Wenn Sie nun in Ihrem angelegten Verzeichnis nachsehen, haben sich folgende Dateien (selbst) erzeugt:



Name	Änderungsdatum	Typ	Grö
._history	14.02.2014 10:48	Dateiordner	
Project1.bdsproj	14.02.2014 10:49	Borland Developer...	
Project1.bdsproj.local	14.02.2014 10:49	LOCAL-Datei	
Project1.cfg	14.02.2014 10:49	CFG-Datei	
Project1.dpr	14.02.2014 10:49	Delphi Project File	
Project1.res	14.02.2014 10:48	RES-Datei	
Unit1.dfm	14.02.2014 10:48	Delphi Form	
Unit1.pas	14.02.2014 10:48	Delphi Source File	

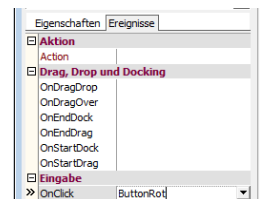
Übung 2

Teil 1: Ereignisse – Beim Klick auf die entsprechenden Buttons sollen sich die Farben des Formularhintergrundes ändern

Teil 2: Ereignisse – Bei Bewegung der Maus über dem Button soll sich die Farbe ändern



1. Komponenten erzeugen (siehe oben)
2. Speichern Sie das Projekt unter
H:\.....\DELPHI\Übung2\Unit1.pas
H:\.....\DELPHI\Übung2\Project1.bdsproj
3. Testen mit F9
4. Eigenschaften festlegen, z.B. rote Schrift
5. Festlegung der Ereignismethoden, dabei auf treffende Ereignisnamen achten!!!!
6. Zuordnung des Quelltextes zum Ereignis:



```

26 procedure TForm1.ButtonRot(Sender: TObject);
27 begin
28   form1.color:=clred;
29 end;
30

```

7. Testen mit F9 und evtl. erneut speichern

2. Datentypen

Numerische Datentypen (Auswahl)

Bezeichnung	Bereich	Genauigkeit	Speicherplatzbedarf
Integer	-2147483648 ... 2147483647		32 Bit mit Vorzeichen
Shortint	-128 ... 127		8 Bit mit Vorzeichen
Longint	-2147483648 ... 2147483647		32 Bit mit Vorzeichen
Real	$5.0 \cdot 10^{-324} \dots 1.7 \cdot 10^{308}$	15-16	8 Byte

Alphanumerische Datentypen

Bezeichnung	Bereich	Genauigkeit	Speicherplatzbedarf
Char	Genau 1 Zeichen		8 Bit = 1 Byte
String	„unbegrenzte“ Länge		4 Byte – 2 GByte
String[10]	bei Compiteroption {\$H+}		10 Zeichen

Operatoren und Funktionen

Analog anderer Programmiersprachen kommen in DELPHI Vergleichsoperatoren, Berechnungsoperatoren und logische Operatoren in folgender Reihenfolge zur Anwendung: (Die Pfeilrichtung gibt „abnehmend“ an!)

1. () Klammern
 2. not, + - (Vorzeichen)
 3. *, /, div, mod, and
 4. +, -, or, xor
 5. <, <=, =, <>, >=, >
- ↓
-

Ausgewählte Standardfunktionen: sqr(), sqrt()

3. Typumwandelnde Funktionen:

StrToInt(s)	konvertiert einen String S, der einen Integer Wert repräsentiert (in dezimaler oder hexadezimaler Form), in eine Zahl	Integer
StrToFloat(s)	der String S wird in einen Gleitkommawert konvertiert	Extended
IntToStr()	konvertiert einen Integer-Wert in einen String	String
FloatToStr(x)	konvertiert eine Gleitkommazahl x in den entsprechenden String-Wert	String
FloatToStrF(x)	konvertiert eine Gleitkommazahl in einen String-Wert mit dem angegebenen Format, der Genauigkeit und den Stellen	String

Übung 3 – Mathematische Berechnungen

Edit Objekte sind zur Ein- und Ausgabe von Daten geeignet. Sie enthalten aber nur Text. Um Berechnungen durchführen zu können, sind Konvertierungen zwischen den Datentypen für Text und denen für Zahlen notwendig (siehe Lehrbrief)

Grundrechenarten
Geben Sie bitte 2 Ganze Zahlen ein!

x y

Ich bin ein EDIT-Feld

Bei mir kannst du Daten im **Textformat** eingeben und ausgeben

Wenn du Zahlen eingibst, werden diese als Text erfasst und bei der Ausgabe musst du Zahlen wieder in Text **„umwandeln“**

1. Gestalten Sie das Formular zunächst entsprechend der Vorlage! (edit Objekte durchnummerieren – edit1, edit2, edit3, edit4)
2. Speichern Sie das Projekt unter **H:\.....\DELPHI\Übung3\Unit1.pas** und **H:\.....\DELPHI\Übung3\Project1.bdsproj**
3. Testen mit F9
4. Festlegung der Ereignismethoden für die Buttons , dabei auf treffende- Ereignisnamen achten!!!!

```

47 procedure TForm1.summe(Sender: TObject);
48 begin
49     sum:=strToInt(edit1.text)+strToInt(edit2.Text) ;
50     edit3.Text:=IntToStr(sum);
51 end;
52
53 procedure TForm1.Different(Sender: TObject);
54 begin
55     diff:=strToInt(edit1.text)-strToInt(edit2.Text) ;
56     edit4.Text:=IntToStr(diff);
57 end;
58

```

ACHTUNG: Die Konvertierung in ein numerisches Format funktioniert nur, wenn die benutzten Edit-Objekte (Eingabefelder) zur Laufzeit des Ereignisses **nicht leer** sind!

Grundrechenarten
Geben Sie bitte 2 Ganze Zahlen ein!

x y

- Ergänzen Sie das Formular um die neben stehenden Objekte
- Löschen realisiert man durch die Zuweisung eines Leerzeichens in ein Textfeld, z.B. edit1.text:=' ';
- Beenden des Programms realisiert man mit form1.close;
- Drucken Sie den Quelltext aus

Übung 4 – Mathematische Berechnungen

CD/ DVD Kauf

Anzahl der CD's: Edit1

Einzelpreis: Edit2

Gesamtpreis: Edit3

19% Mwst: Edit4

Endpreis: Edit5

1. Gestalten Sie das Formular zunächst entsprechend der Vorlage! (edit Objekte durchnummerieren!)
2. Speichern Sie das Projekt unter **H:\.....\DELPHI\Übung4\Unit1.pas** und **H:\.....\DELPHI\Übung4\Project1.bdsproj**
3. Testen mit F9
4. Festlegung der Ereignismethode für den Button „Berechne“
5. Variablen, z.B. gp, mwst, ep sind entsprechend des Datentyps zu deklarieren:
var Form1: TForm1;
gp,mwst,endpreis:real;

```

16 |
17 | procedure TForm1.berechne(Sender: TObject);
18 | begin
19 |   gp:=strtfloat(edit1.text)*strtfloat(edit2.text);
20 |   mwst:=0.19*gp;
21 |   endpreis:=gp+mwst;
22 |   edit3.Text:=floattostrF(gp,ffCurrency,10,2);
23 |   edit4.Text:=floattostrF(mwst,ffCurrency,10,2);
24 |   edit5.Text:=floattostrF(endpreis,ffCurrency,10,2);
25 |

```

Zur Darstellung eines Wertes im Euro- Format nutzt man die Formatierung die Funktion **FloatToStrF(...)**, z.B. edit4.Text:=floattostrF(ra15,ffCurrency,10,2);

CD/ DVD Kauf

Anzahl der CD's:

Einzelpreis:

Gesamtpreis:

19% Mwst:

Endpreis:

Rabatt 10% Edit6

Rabatt 15% Edit7

Rabatt 20%

Edit8

Berechnen Sie zusätzlich über „OnClick-Buttons“ die angegebenen Rabattbeträge!

Speichern
Notieren Sie Ihren Namen in das Programm
Drucken Sie den Quelltext aus

Function **FloatToStrF**(Wert; Format;Vorkommastellen; Nachkommastellen);

ffGeneral, ffExponent, ffFixed, ffNumber, ffCurrency)

Umwandeln
in kürzeste
Form

Exponential-
format

Fest-
komma-
format

Zahlenformat
mit Tausender-
trennzeichen

Währungsformat
entsprechend der
Voreinstellung

Beispiele:

FloatToStrF(wert, ffFixed,8,1) → 123,7 oder 0,5

FloatToStrF(wert, ffNumber,12,2) → 1.385.445,13 oder 5.323,03

FloatToStrF(wert, ffCurrency,8,2) → 243,55 €

Übung 5

Erarbeiten Sie selbständig (nach bekannter Schrittfolge) ein Programm zur Erstellung einfacher Rechenoperationen entsprechend dem nachstehenden Muster!
Nach Eingabe der drei Zahlen, sollen auf Klick deren Mittelwert mit zwei Nachkommastellen und deren Produkt berechnet und ausgegeben werden.

Berechnungen


Schriftart: Arial
Schriftgröße: 20
rot, kursiv, unterstrichen

Zahl 1:

Zahl 2:

Zahl 3:

zahl.jpg



Mittelwert

37,67

Produkt

22800

Löschen

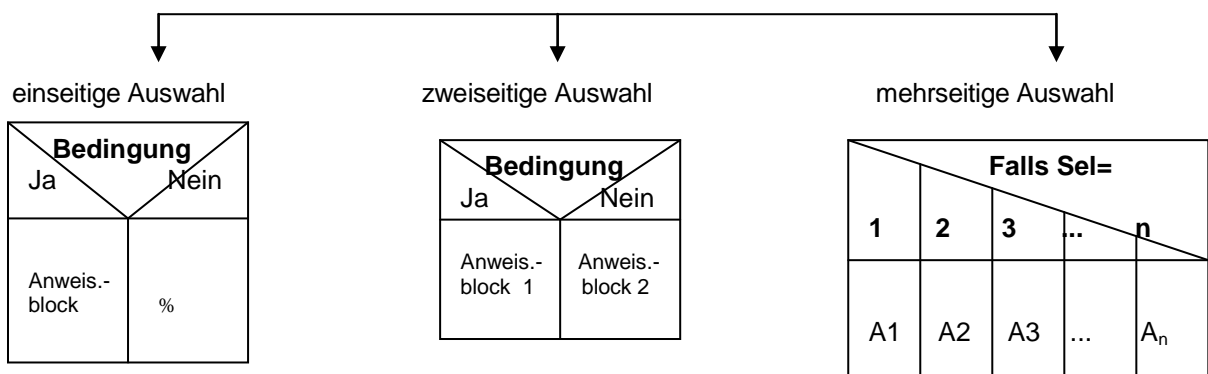
Schriftart: Arial
Schriftgröße: 20
rot, kursiv, unterstrichen
Ausgabe mit 2 Nachkommastellen

Übung 6 – Formatierungen

Ändern Sie in der Übung 3 verschiedene Eingabe- und Ausgabeformate!

1. Die Eingabe soll auch für Dezimalzahlen möglich sein
2. Die Ausgabe des Produktes soll mit 2 Dezimalstellen genau berechnet werden.
3. Die Ausgabe des Quotienten soll mit 5 Dezimalstellen genau ausgegeben werden.

4. Auswahlstrukturen (Verzweigungen)



Einseitige Auswahl: IF Bedingung wahr THEN Aktion(en);

- IF x=0 THEN Edit1.Text:='Division durch Null ist nicht möglich';
 - IF y>50 THEN x:=a+b;
 - IF anzahl>100 THEN begin
 - edit1.Text:='Sie erhalten 5% Rabatt.';
 - p:=p*0.95; {p*0.95 =p -p*0,05}
 - Edit2.Text:=floatToStrF(p,ffcurrency,5,2);
- end;

Zweiseitige Auswahl (Alternativen): IF Bedingung wahr THEN Aktion(en) ELSE Aktion(en);

IF anzahl<10 THEN gesamt:=preis*anz ELSE gesamt:=preis*anz*0.98;

```
IF ... THEN begin
    ...
    end
ELSE begin
    ...
    end;
```

Bei der Verwendung logischer Operatoren müssen die Bedingungen in Klammern stehen !

IF (a<0) **OR** (a>100) THEN...

IF (a>10) **AND** (a<=100) THEN ...

Vor ELSE darf kein Semikolon stehen !

Übung 7 – Verzweigungen (Alternativen)

Gewichtsbestimmung

Gewichtsbestimmung

Bitte geben Sie die Körpergröße in cm ein:

Normalgewicht: Idealgewicht:

Geben Sie Ihr tatsächliches Gewicht ein:

Erarbeiten Sie ein Programm, das nach Eingabe der Körpergröße in cm das Normalgewicht und das Idealgewicht in kg ausgibt.

Nach Eingabe des tatsächlichen Gewichtes soll eine Wertung ausgegeben werden:

Wenn das tatsächliche Gewicht < Idealgewicht dann *Du bist viel zu dünn!*

Wenn das tatsächliche Gewicht zwischen den Normal und Idealgewicht liegt dann *Du bist.....!*

Wenn das tatsächliche Gewicht > Normal dann *Du bist*

Formeln: Normalgewicht:=
Idealgewicht:=

Übung 8 – Verzweigungen (Alternativen)

Rabattberechnung

Erarbeiten Sie ein Programm, das nach Eingabe der Kaufstückzahl den Rabatt in % und den Endbetrag ausgibt. Der Rabatt staffelt sich nach folgendem Muster:

- Kaufstückzahl < 100 → kein Rabatt
- Kaufstückzahl >=100 und <=250 → 5% Rabatt
- Kaufstückzahl >250 und <500 → 10% Rabatt
- Kaufstückzahl >=500 und <1200 → 15% Rabatt
- Kaufstückzahl >=1200 → 20% Rabatt

Rabattberechnung

Kaufstückzahl:

Einzelpreis in €:

Gesamtpreis in €:

- Rabatt in €:

Endbetrag in €:

Rabatt in %:

Rabattbetrag in €:

Übung 9 – Verzweigungen (Alternativen)

Zensur

Erarbeiten Sie ein Programm, das nach Eingabe der erreichten Punktzahl die Note einer Klassenarbeit anzeigt.

Es gelten folgende Werte:
bis 92% - Note 1, bis 81% - Note 2, bis 67% - Note 3, bis 50% - Note 4, bis 30% - Note 5

Gesamtpunktzahl der Arbeit:

Erreichte Punktzahl: