

Informatik OTG	Rekursion	
	Berechnung der Fakultät	Delphi Quelltext

Verfahren Rekursion

ist die Abbruchbedingung erreicht,
dann Abbruchanweisung(en)
sonst Anweisung(en) mit Aufruf(en) des Unterprogramms "Rekursion"

Ende des Verfahrens Rekursion

Die Variable(n), die in der Abbruchbedingung abgeprüft werden, müssen sich ändern, damit die Bedingung erfüllt werden kann.

Beispiel - **Berechnung der Fakultät** einer ganzen, positiven Zahl:

$n! = 1*2*3*4*5* \dots (n-1)*n$	$n! = 1$, falls $n = 0$ oder 1 $n! = n * (n-1)!$, falls $n > 1$
---------------------------------	------------------------------------------------------------------------

Also:

$1! = 1 = 1$
 $2! = 1*2 = 2$
 $3! = 1*2*3 = 6$
 $4! = 1*2*3*4 = 24$
 $5! = 1*2*3*4*5 = 120$
 $n! = 1*2*3*4*5* \dots *n = \dots$

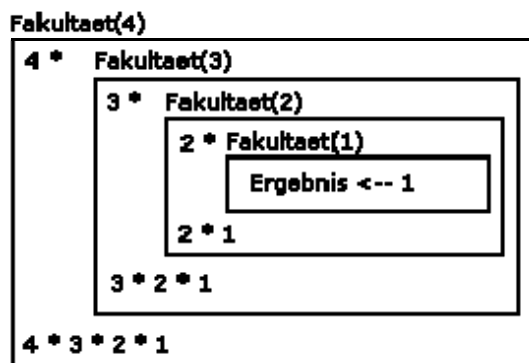
Aufbau eines entsprechenden Unterprogramms:

Funktion Fakultaet (Eingabeparameter n - positive, ganze Zahl) liefert eine positive, ganze Zahl

Ist $n \leq 1$
dann ist das Ergebnis des Unterprogramms 1
sonst ist das Ergebnis des Unterprogramms $n * \text{Fakultaet}(n-1)$

Ende des Unterprogramms Fakultaet

Ruft man nun zum Beispiel Fakultaet(4) auf, läuft Folgendes ab:



An diesem Beispiel erkennt man einen Nachteil der Rekursion: von jedem Unterprogramm, das einen erneuten Rekursionsaufruf startet, werden die laufenden Variablenbelegungen und die Einsprungstelle nach Beendigung des Aufrufs aufgehoben. Dieser Speicherbereich (stack = Stapel) kann bei großer Rekursionstiefe schnell anwachsen und den verfügbaren Speicher ausfüllen, was einen Programmabbruch zur Folge hat.

Informatik OTG	Rekursion	
	Berechnung der Fakultät	Delphi Quelltext

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Spin;
```

```
type
```

```
TForm1 = class(TForm)
```

```
Button1: TButton;
```

```
Label1: TLabel;
```

```
Button2: TButton;
```

```
Label2: TLabel;
```

```
Eingabe: TSpinEdit;
```

```
Ergebnis: TSpinEdit;
```

```
procedure Button2Click(Sender: TObject);
```

```
procedure Button1Click(Sender: TObject);
```

```
private
```

```
{ Private-Deklarationen }
```

```
public
```

```
{ Public-Deklarationen }
```

```
end;
```

```
var
```

```
Form1: TForm1;
```

```
implementation
```

```
{ $R *.dfm }
```

```
function fakult(n: integer): integer;
```

```
begin
```

```
if n=0 then fakult := 1
```

```
else fakult := fakult(n-1)*n;
```

```
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
```

```
begin
```

```
close;
```

```
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
```

```
Ergebnis.Value := fakult(Eingabe.Value);
```

```
end;
```

```
end.
```

// SpinButton ist unter Beispiele zu finden

