

Unterprogramme: Funktionen

Projekt Dreiecksberechnungen

Verwendung von Funktionen

Rückgabe genau eines Ergebniswertes

vgl. $\sin(x)$, $\cos(x)$, $\tan(x)$, $\text{random}(x)$, $\text{round}(x)$, etc.

Deklaration der Funktion in der Typdefinition der Schnittstelle

```
type
  TForm1 = class(TForm)
  private
    function Pythagoras(Kathete1, Kathete2 : Real) : Real;
  public
  end;
```

- Liste der **Übergabeparameter** und deren **Datentyp**
- **Datentyp** des (eindeutigen) Rückgabeparameters

Unterprogramme: Funktionen

Projekt Dreiecksberechnungen

Definition der Funktion im Implementierungsteil der Unit:

➤ Wiederholung der **Funktions-Deklaration** (Kopfzeile).

```
function TForm1.Pythagoras(Kathete1,Kathete2: Real): Real;  
var a,b,c,d: Integer;  
begin  
  Result := Sqrt(Kathete1*Kathete1 + Kathete2*Kathete2);  
end;
```

- Deklaration **lokaler Variablen** (falls vorhanden)
- **Anweisungsteil** (begin end)
- Vordefinierte **Ergebnis-Variable** namens **Result** nimmt **Funktionswert** auf (Kompatibel zum **Ergebnis-Typ**).

Unterprogramme: Funktionen

Projekt Dreiecksberechnungen

Weitere Beispiele:

```
function TForm1.Kreisflaeche(Radius: Real): Real;  
begin  
    Result := Radius*Radius*Pi;  
end;
```

```
A := Kreisflaeche(r);
```

```
function TForm1.Maximum(Zahl1,Zahl2: Real): Real;  
begin  
    if Zahl1 > Zahl2  
    then Result := Zahl1  
    else Result := Zahl2;  
end;
```

```
Top := Maximum(a,b);
```

```
if Kreisflaeche(r) > Maximum(a,b) then
```

```
function TForm1.Vergleich(Eingabe,Ergebnis:Integer): String;  
begin  
    if Eingabe := Ergebnis  
    then Result := ,Richtig\  
    else Result := ,Falsch\  
end;
```

```
Edit.Text := Vergleich(a,b);
```

Unterprogramme: Funktionen

Projekt Dreiecksberechnungen

Aufgabenstellung:

Schreiben Sie ein Programm, welches Flächeninhalt und Umfang eines beliebigen Dreiecks mit den Seitenlängen a , b , c berechnet.

The screenshot shows a window titled "Dreiecksberechnungen" with a standard Windows title bar. The window content is as follows:

Eingabe:		
Seite a	<input type="text" value="8"/>	cm
Seite b	<input type="text" value="6"/>	cm
Seite c	<input type="text" value="8"/>	cm

Ausgabe:		
Fläche	<input type="text" value="22,25"/>	cm ²
Umfang	<input type="text" value="22,00"/>	cm

At the bottom right of the window is a button labeled "Berechne".

Unterprogramme: Funktionen

Projekt Dreiecksberechnungen

```
procedure TFoDreieck.BuBerechneClick(Sender: TObject);
```

```
var a,b,c: Real;
```

```
begin
```

```
  a := StrToFloat(EdSeiteA.Text);
```

```
  b := StrToFloat(EdSeiteB.Text);
```

```
  c := StrToFloat(EdSeiteC.Text);
```

```
  if DrProbe(a,b,c) = true
```

```
  then
```

```
  begin
```

```
    EdUmfang.Text := FloatToStrF( Umfang (a,b,c),          ffFixed,10,2);
```

```
    EdFlaeche.Text := FloatToStrF( Flaeche(a,b,c),          ffFixed,10,2);
```

```
    EdKRadius.Text := FloatToStrF( KRadius(Flaeche(a,b,c)), ffFixed,10,2);
```

```
    EdKUmfang.Text := FloatToStrF( KUmfang(Flaeche(a,b,c)), ffFixed,10,2);
```

```
  end
```

```
  else showMessage('Dreieck nicht konstruierbar');
```

```
end;
```

```
private
```

```
  { Private declarations }
```

```
  function Umfang (a,b,c: Real):Real;
```

```
  function Flaeche(a,b,c: Real):Real;
```

```
  function DrProbe(a,b,c: Real):Boolean;
```

```
  function KRadius(f: Real):Real;
```

```
  function KUmfang(f: Real):Real;
```

```
public
```

```
  { Public declarations }
```

```
end;
```

Unterprogramme: Funktionen

Projekt Dreiecksberechnungen

```
private
  function Umfang(a,b,c: Real):Real;
  function Flaechе(a,b,c: Real):Real;
  function DrProbe(a,b,c: Real):Boolean;
  function KRadius(f: Real):Real;
  function KUmfang(f: Real):Real;
public
```

```
function TFoDreieck.DrProbe(a,b,c: Real):Boolean;
begin
  if (a+b > c)
    and (b+c > a)
    and (c+a > b)
  then Result := True
  else Result := False;
end;
```

```
function TFoDreieck.Umfang(a,b,c: Real):Real;
begin
  Result := a + b + c;
end;
```

```
function TFoDreieck.Flaechе(a,b,c: Real):Real;
var s: real;
begin
  s := Umfang(a,b,c)/2;
  Result := Sqrt(s*(s-a)*(s-b)*(s-c));
end;
```