

"Minirechner": Pascal-Code

Wenn der Anwender auf einen Button klickt, soll ein bestimmtes Ereignis ausgelöst werden – zB. die Addition der eingegebenen Zahlen und Ausgabe des Ergebnis.

Dazu muss der Programmierer eine Ereignisbehandlungsroutine bzw. Prozedur in der Programmiersprache "Pascal" schreiben.

Der grundsätzliche Aufbau einer solchen Prozedur wird am Beispiel der Addition erläutert:

```

procedure TForm1.addClick(Sender: TObject);
var a:double; b:double; s:double;
begin
  a:=StrToFloat(zahl1.text);
  b:=StrToFloat(zahl2.text);
  s:=a+b;
  ergebnis.caption:='Das Ergebnis der Addition: ';
  erg_zahl.text:=FloatToStr(s);
end;

```

procedure TForm1.addClick(Sender: TObject);	Der Name der Prozedur inkl. der nötigen Parameter wird Dir nach einem Doppelklick auf den jeweiligen Button im Formular von der Lazarus-IDE im Quelltexteditor bereits vorgegeben.
var a:double; b:double; s:double;	Mit dem Schlüsselwort var erfolgt die Deklaration der Variablen. Wir benötigen hier zwei Variablen (a,b), denen wir die Eingabewerte des Anwenders zuweisen wollen. Außerdem brauchen wir noch die Variable (s) für das Ergebnis. All diesen Variablen müssen wir einen Datentyp zuweisen, wie z.B.: double Dezimalzahl (Fließkommazahl, Float) integer ganze Zahl string Zeichenkette (Text)
begin	Schlüsselwort für den Anfang unserer Prozedur
a:=StrToFloat(zahl1.text); b:=StrToFloat(zahl2.text);	Nun müssen wir die Eingabewerte des Anwenders gewissermaßen an unsere Variablen binden, um dann mit ihnen rechnen zu können. Da Eingabewerte in einem Edit-Feld immer als Text (String) interpretiert werden, müssen diese vorher mit der Funktion StrToFloat (String to Float) in eine Dezimalzahl umgewandelt werden. := ist der sog. Zuweisungsoperator (sprich: "ergibt sich aus...")
s:=a+b;	Nun folgt die eigentliche Berechnung: "s ergibt sich aus a + b"
ergebnis.caption:='Das Ergebnis der Addition: ';	In unserem (bisher leeren) Ergebnis-Label wollen wir dem Anwender noch eine Meldung ausgeben. Nicht unbedingt nötig - aber hübsch.
erg_zahl.text:=FloatToStr(s);	Um den Ergebniswert im Edit-Feld ausgeben zu können, muss dieser wieder in einen Text (String) umgewandelt werden.
end;	Schlüsselwort für das Ende unserer Prozedur

Weitere Methoden:

<code>zahl1.clear;</code>	Löscht den Inhalt des angegebenen Edit-Feldes.
<code>ergebnis.caption:='';</code>	Löscht den Inhalt des angegebenen Labels indem ihm eine leere Zeichenkette zugewiesen wird.
<code>close;</code>	Schließt die gesamte Anwendung.

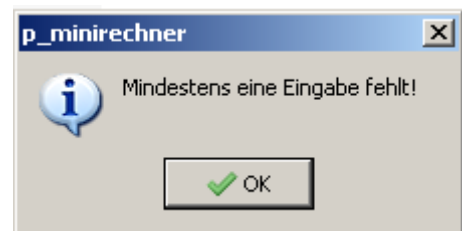
Unsere Anwendung wird "DAU"-sicher...

Auch bei dem dümmsten anzunehmenden User muss unser Programm zuverlässig funktionieren! D.h. auch bei fehlenden oder fehlerhaften Nutzereingaben darf unser Programm nicht abstürzen, sondern muss in diesem Fall eine kontrollierte Fehlermeldung ausgeben.

Fehlende Eingabewerte:

Wenn der User vergisst, seine Zahlen einzugeben, soll er darauf hingewiesen werden. Dafür bietet sich eine Selektion an:

```
procedure TForm1.AddClick(Sender: TObject);
var a:double; b:double; s:double;
begin
  if (zahl1.text='') or (zahl2.text='') then
    ShowMessage('Mindestens eine Eingabe fehlt!')
  else
    begin
      a:=StrToFloat(zahl1.text);
      b:=StrToFloat(zahl2.text);
      s:=a+b;
      ergebnis.caption:='Das Ergebnis der Addition: ';
      erg_zahl.text:=FloatToStr(s);
    end;
end;
```

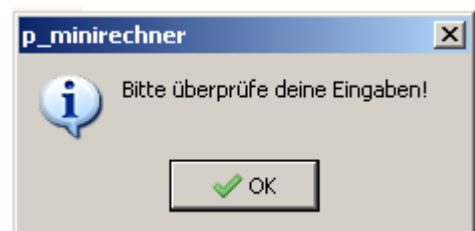


Die Selektion könnte man so übersetzen: "Wenn im Edit-Feld **zahl1** oder im Edit-Feld **zahl2** eine leere Zeichenkette steht, dann zeige die Mitteilung 'Mindestens eine Eingabe...', andernfalls führe alle Anweisungen normal aus".

Fehlerhafte Eingabewerte:

Wenn der User statt Zahlen irgendwelche Buchstaben oder Zeichen eingibt, soll er darauf hingewiesen werden.

```
procedure TForm1.AddClick(Sender: TObject);
var a:double; b:double; s:double;
begin
  try
    if (zahl1.text='') or (zahl2.text='') then
      ShowMessage('Mindestens eine Eingabe fehlt!')
    else
      begin
        a:=StrToFloat(zahl1.text);
        b:=StrToFloat(zahl2.text);
        s:=a+b;
        ergebnis.caption:='Das Ergebnis der Addition: ';
        erg_zahl.text:=FloatToStr(s);
      end;
  except
    ShowMessage('Bitte überprüfe deine Eingaben!');
  end;
end;
```



Zum Verständnis:

Alle Anweisungen, die zwischen **try** und **except** stehen, werden vorher intern quasi "probehaltig" ausgeführt. Sollte dabei ein Fehler auftreten - z.B. durch eine fehlerhafte Eingabe - wird die Nachricht im **except**-Zweig ausgegeben.