

# Taschenrechner mit Fehlerrechnung v1.0

Jan Krieger, *Heidelberg*

jan@jkrieger.de --- <http://www.jkrieger.de/>

5. März 2003

## Zusammenfassung

Mit diesem Programm kann man mathematische Formeln evaluieren. Es bietet dabei zusätzlich zu den Standard-Optionen die Möglichkeit mit fehlerbehafteten Werten zu arbeiten. Außerdem können während der Laufzeit eigene Konstanten/Variablen definiert und verwendet werden.

## 1 Bedienung

Das Programm besitzt eine schlichte Oberfläche: ein „DOS-Fenster“ mit einem Eingabeprompt. Man kann direkt Ausdrücke eingeben. Der Wert des jeweils letzten Ausdruckes in einer Zeile wird als Ergebnis ausgegeben. Ein Beispiel:

```
> x=5; x^2+1
= 26
```

Falls, wie im obigen Beispiel mehrere Ausdrücke hintereinander eingegeben werden sollen, kann man sie durch ; trennen. Ein abschließendes ; ist möglich, aber nicht erforderlich, das es automatisch hinzugefügt wird.

Um eine kurze Beschreibung der Befehle zu erhalten kann man `help` eintippen. Der Befehl `end` beendet das Programm.

## 2 Programmstruktur, Lizenz und Bemerkungen

Das Programm ist in *Delphi 5* geschrieben und verwendet die Bibliothek (unit) *JKErrorCalculation* zur Fehlerrechnung und die Komponente *TjkMathParser* als Formelinterpreter. Beide stehen auf meiner Homepage als **FREEWARE** zur Verfügung. Der Quellcode dieses Interpreters liegt als Beispiel bei.

Die units dürfen für alle Anwendungen verwendet werden. Auch einer Weiterentwicklung steht nichts im Wege, solange der ursprüngliche Autor genannt wird (bei Programmen z.B. im Info-Dialog). Bearbeitungen (und Ableitungen) dieser Unit müssen allerdings bei Veröffentlichung wieder **FREEWARE** sein. Eine kostenpflichtige Weitergabe (Verkauf) dieser unit durch Dritte ist nicht gestattet (dies schießt die Verwendung in kommerzieller Software ausdrücklich nicht aus).

Intern arbeitet das Programm mit dem Datentyp `extended` von Delphi 5. Es handelt sich um einen Gleitkommatyp mit 19-20 signifikanten Stellen. Sein Wertebereich umspannt  $3.4 \cdot 10^{-4932} \dots 1.1 \cdot 10^{-4932}$

Der Interpreter basiert auf einem Programmbeispiel aus dem folgenden Buch:

*Bjarne Stroustrup*, „**Die C++-Programmiersprache**“; Addison-Weseley, 2000, S. 116ff

Wer Fehler im Programm findet möge mir doch ein e-mail an [jan@jkrieger.de](mailto:jan@jkrieger.de) schicken.

## 3 Die Syntax

Die folgenden Abschnitte beschreiben die Syntax, die von diesem programm (und der Komponente *TjkMathParser* unterstützt wird.

### 3.1 Struktur

Eine gültige Eingabe besteht aus einer Aneinanderreihung von mehreren Ausdrücken. Sie werden durch das Zeichen ; getrennt. Ein Ausdruck kann eine Formel, oder eine Zuweisung an eine Konstante (Variable) sein. Groß-/Kleinschreibung ist nicht relevant.

Formeln werden wie in der Mathematik üblich eingegeben. Es sind diverse Operatoren (+, -, \*, /, !, %, ^), Klammern ( , ), Absolutbetrag-Striche (|·|), vordefinierte Funktionen (`name(<Argument>)` bzw. `name(<Argument>, <Argument>)`), sowie verschiedene Zahlendarstellungen erlaubt (Exponentialschreibweise: `3e-4`, HEX-Zahlen: `$FF`, Zahl mit Fehler: `{3, 0.5}`). Die einzelnen Elemente werden im folgenden erklärt.

Gültige Anweisungen wären etwa:

```
n=(3! * 5!)^2%4; n^2/5
a={3, 0.5}; b={4e-3, 0.4e-4}; a+b
e^(-7/(2*pi))
{2, 0.5}-{1, 0.4}
r={4, 0.2}; 1/2*pi*r^2
alpha=pi/2; sin(alpha)
pi1=2*arccos(0);
```

## 3.2 Zahlen, Konstanten

Es gibt zwei Möglichkeiten Zahlen einzugeben:

- *Exponentialdarstellung*: Zahlen können ganz normal mit dem Punkt „.“ als Dezimaltrennzeichen eingegeben werden (Bsp: 3 3.1415 ...). Zusätzlich kann durch E abgetrennt ein Exponent angegeben werden. So entspricht etwa der Zahl  $3.1415 \cdot 10^{-5}$  die Darstellung 3.1415E-5. Vor negativen Zahlen wird einfach ein Minus angegeben (Bsp: -3.5E3).
- *HEX-Darstellung*: Ganzzahlige Werte können zusätzlich in hexadezimaler Schreibweise angegeben werden. Vor die Zahl muss dann ein \$ gestellt werden. Bsp: \$3FF, \$A2
- *fehlerbehaftete Zahlen*: Zu jeder Zahl kann auch ein Fehler angegeben werden. Dann muss die Zahl in geschweifte Klammern {} gesetzt werden. Zahl und fehler werden durch ein Komma getrennt. Bsp: {3.1415, 0.01}, {3.1415, 1e-4}, {\$123, 0.01} (Es sind also auch HEX-Zahlen in fehlerangaben erlaubt).  
Um etwa eine Geschwindigkeit zu berechnen, wenn man  $x = (10 \pm 0.05)\text{cm}$  und  $t = (10\text{s} \pm 10\text{ms})$  bestimmt hat gibt man folgendes ein:

```
> {10e-2, 0.05e-2}/{10, 10e-3}
= (0,01 +/-5,09901951359278E-5)
```

Und erhält also als Ergebnis:  $(0.01000 \pm 5 \cdot 10^{-5}) \frac{\text{m}}{\text{s}}$ .

Es ist möglich neben den vordefinierten Konstanten eigene einzuführen. Diesen können beliebige Namen zugewiesen werden, sofern die Namen mit einem Buchstaben ('a'..'z') beginnen und sich danach nur noch aus den folgenden Zeichen zusammensetzen: ['a'..'z', '0'..'9', '\_'].

Eine Konstante wird durch eine Zuweisung definiert, oder überschrieben. danach darf sie in Formeln, wie eine Zahl verwendet werden. Bei einer Zuweisung steht zuerst der Konstantenname, danach eine '='-Zeichen und dann der Ausdruck, dessen Wert der Konstante zugewiesen werden soll.

Obiges Beispiel kann man auch mit Konstanten lösen. Wenn man noch zusätzlich die Anfangsgeschwindigkeit  $v_0 = 0$  kennt, kann man dann auch die Beschleunigung  $a$  berechnen:

```
> x={10e-2, 0.05e-2}; t={10, 10e-3}; v=x/t
= (0,01 +/-5,09901951359278E-5)
> a=v/t
= (0,001 +/-5,19615242270663E-6)
```

man erhält also:  $v = (0.01000 \pm 5 \cdot 10^{-5}) \frac{\text{m}}{\text{s}}$  und  $a = (0.001000 \pm 5 \cdot 10^{-6}) \frac{\text{m}}{\text{s}^2}$ .

**ACHTUNG:** Vordefinierte Konstanten können überschrieben werden! So liefert folgender Ausdruck nicht mehr die Kreisfläche  $\pi r^2$  mit  $\pi = 3.1415\dots$ , da jetzt  $pi = 5$  gilt:

```
> pi=5; r=2; pi*r^2
= 20
```

Auf diese Weise können natürlich auch genauere Werte berechnet werden.

Es sind eine ganze Reihe von mathematischen und physikalischen Konstanten vordefiniert.

### 3.2.1 mathematische Konstanten

Konstante	Wert	Erklärung
Pi	3.1415926535897932385	Kreiszahl $\pi$
e	2.71828182845905	Euler'sche Zahl $e$
Random, Rnd		Zufallszahl aus [0..1]

## 3.2.2 physikalische Konstanten

Konstante	Wert	Einheit	Erklärung
m_alpha	$6.6446616 \cdot 10^{27}$	[kg]	Ruhemasse eines He <sup>2+</sup>
m_alpha_u	4.0015065	[u]	– ” –
at_unit	$1.660540 \cdot 10^{27}$	[kg]	1/12 der Masse des <sup>12</sup> C (=1 u)
avogadro	$6.0221 \cdot 10^{23}$	[1/mol]	Avogadro-Zahl
boltzmann	$1.3807 \cdot 10^{23}$	[J/K]	Boltzmann-Konstante
boltzmann_ev	$8.617386 \cdot 10^5$	[eV/K]	– ” –
efield	$8.8542 \cdot 10^{12}$	[C/V m]	elektrische Feldkonstante
m_e	$9.1094 \cdot 10^{31}$	[kg]	Ruhemasse: Elektron
m_e_u	$5.48580 \cdot 10^4$	[u]	– ” –
elem_charge	$1.6022 \cdot 10^{19}$	[C]	Elementarladung
g_norm	9.80665	[m/s <sup>2</sup> ]	Fallbeschleunigung, Norm
g_hd	9.8101	[m/s <sup>2</sup> ]	Fallbeschleunigung, Heidelberg
faraday	96485.309	[C/mol]	Faraday-Konstante
gasconst	8.3145	[J/K mol]	allg. Gaskonstante
gravity	$6.673 \cdot 10^{11}$	[m <sup>3</sup> /kg s <sup>2</sup> ]	Gravitationskonstante
c_light	$2.99792458 \cdot 10^8$	[m/s <sup>2</sup> ]	Lichtgeschwindigkeit im Vakuum
m_neutron	$1.67493 \cdot 10^{27}$	[kg]	Ruhemasse des Neutrons
m_neutron_u	1.008665	[u]	– ” –
v_norm	22.4	[l/mol]	molares Vol. eines idealen Gases im Normzustand (1013 mbar, 273 K)
planck	$6.6261 \cdot 10^{34}$	[J s]	Planck-Konstante
planck_ev	$4.1357 \cdot 10^{15}$	[eV s]	– ” –
m_proton	$1.67262 \cdot 10^{27}$	[kg]	Ruhemasse: Proton
m_proton_u	1.007276	[u]	– ” –
stefan_boltzmann	$5.6705 \cdot 10^8$	[W/m <sup>2</sup> K <sup>4</sup> ]	Stefan-Boltzmann-Konstante
m_earth	$5.977 \cdot 10^{24}$	[kg]	Erdmasse
r_earth	$6268 \cdot 10^3$	[m]	Erdradius
m_sun	$1.98 \cdot 10^{30}$	[kg]	Sonnenmasse
r_sun	$6.96 \cdot 10^8$	[m]	Sonnenradius
m_moon	$1.23 \cdot 10^{-2} \cdot 5.977 \cdot 10^{24}$	[kg]	Mondmasse
r_moon	$0.273 \cdot 6268 \cdot 10^3$	[m]	Mondradius
density_h2o	0.998	[kg/l]	Dichte von Wasser (bei 1013 mbar, 293 K)
density_air	1.293	[kg/m <sup>3</sup> ]	Dichte von Luft (bei 1013 mbar, 273 K)

Die Werte für die physikalischen Konstanten entstammen dem Buch:  
H. Hammer, K. Hammer: **“Physikalische Formeln und Tabellen“**, J. Lindauer verlag (Schaefer), 1994<sup>6</sup>.

## 3.3 Die Operatoren

Die folgenden Operatoren sind erlaubt. Sie sind in der reihenfolge ihrer Interpretation aufgeführt (zuerst Potenz, dann Multiplikation, dann Addition ...):

- Fakultät n!: n darf dabei ein beliebiger Ausdruck sein, der einen ganzzahligen Wert ergibt. Bsp: n=3, n=(3+3) usw. Die Fehlerbehandlung für Fakultäten ist nicht implementiert. Falls n Fehlerbehaftet war, so wird der Fehler

auf 0 gesetzt. Der größte Wert, von dem eine Fakultät berechnet werden kann ist 25. Definition der Fakultät:  
 $n! = n \cdot (n - 1) \cdot \dots \cdot 2 \cdot 1$

- Potenz  $x^y$ : x und y dürfen dabei beliebige Ausdrücke sein. Die Fehlerbehandlung geschieht nach der Formel:

$$f = x^y \Rightarrow \Delta f = \sqrt{(\Delta x \cdot y \cdot x^{y-1})^2 + (\Delta y \cdot x^y \cdot \ln(y))^2}$$

- Modulo  $x\%y$ : x und y dürfen dabei Ausdrücke sein, die ein ganzzahliges Ergebnis haben. Die Fehlerbehandlung ist für die Modulo-Operation nicht implementiert. Daher wird der Fehler auf 0 gesetzt.
- Multiplikation  $x*y$ : x und y dürfen beliebige Ausdrücke sein. Der Fehler berechnet sich zu:

$$f = x \cdot y \Rightarrow \Delta f = \sqrt{(y \cdot \Delta x)^2 + (x \cdot \Delta y)^2}$$

- Division  $x/y$ : x und y dürfen beliebige Ausdrücke sein. Der Fehler berechnet sich zu:

$$f = x/y \Rightarrow \Delta f = \sqrt{\left(\frac{\Delta x}{y}\right)^2 + \left(\frac{x \cdot \Delta y}{y^2}\right)^2}$$

- Addition/Subtraktion  $x+y$ ,  $x-y$ : x und y dürfen beliebige Ausdrücke sein. Der Fehler berechnet sich zu:

$$f = x \pm y \Rightarrow \Delta f = \sqrt{(\Delta x)^2 + (\Delta y)^2}$$

### 3.4 Funktionen

Funktionen haben immer einen, oder zwei Parameter, die beliebige Ausdrücke sein dürfen und durch ein Komma getrennt werden. Der allgemeine Syntax ist also:

`name(x)` bzw. `name(x, y)`

Folgende Funktionen sind definiert:

#### 3.4.1 Diverses

Funktion	Erklärung	Fehler $\Delta f =$
<code>sqrt(x)</code>	Quadratwurzel ( $= \sqrt{x}$ )	$\left  \frac{\Delta x \cdot \frac{1}{2\sqrt{x}}}{1} \right $
<code>sqr(x)</code>	Quadrat ( $= x^2$ )	siehe Mult.
<code>abs(x)</code> , <code> x </code>	Absolutbetrag ( $=  x $ )	$ \Delta x $
<code>round(x)</code>	gerundeter Wert	$round(\Delta x)$
<code>ceil(x)</code>	kleinster Ganzzahl-Wert, größer oder gleich x	$ceil(\Delta x)$
<code>floor(x)</code>	Abrunden	$floor(\Delta x)$
<code>int(x)</code>	ganzzahliger Anteil	$int(\Delta x)$
<code>frac(x)</code>	Nachkommaanteil	$frac(\Delta x)$
<code>sgn(x)</code>	Vorzeichenfunktion $sgn(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$	0
<code>min(x, y)</code>	Minimum der zwei Zahlen	zugehörig
<code>max(x, y)</code>	Maximum der zwei Zahlen	zugehörig

#### 3.4.2 Potenzen und Logarithmen

Funktion	Erklärung	Fehler $\Delta f =$
<code>exp(x)</code>	Exponentialfunktion ( $= e^x$ )	$ \Delta x \cdot e^x $
<code>log(b, x)</code>	Logarithmus zur Basis b von x ( $= \log_b x$ )	$\left  \frac{\Delta x}{x \cdot \ln b} \right $
<code>ln(x)</code>	natürlicher Logarithmus ( $= \log_e x$ )	$\left  \frac{\Delta x}{x} \right $
<code>lb(x)</code> , <code>log2(x)</code>	binärer Logarithmus ( $= \log_2 x$ )	$\left  \frac{\Delta x}{x \cdot \ln 2} \right $
<code>lg(x)</code> , <code>log10(x)</code>	dekadischer Logarithmus ( $= \log_{10} x$ )	$\left  \frac{\Delta x}{x \cdot \ln 10} \right $

## 3.4.3 trigonometrische Funktionen

Funktion	Erklärung	Fehler $\Delta f =$
$\sin(x)$	Sinus	$ \Delta x \cdot \cos(x) $
$\cos(x)$	Kosinus	$ \Delta x \cdot \sin(x) $
$\tan(x)$	Tangens	$ \Delta x \cdot \frac{1}{\cos^2 x} $
$\cot(x)$	Kotangens	$ \Delta x \cdot \frac{1}{\sin^2 x} $
$\arcsin(x)$	Arcussinus ( $= \sin^{-1} x$ )	$ \frac{\Delta x}{\sqrt{1-x^2}} $
$\arccos(x)$	Arcuskosinus ( $= \cos^{-1} x$ )	$ \frac{\Delta x}{\sqrt{1-x^2}} $
$\arctan(x)$	Arcustangens ( $= \tan^{-1} x$ )	$ \frac{\Delta x}{1+x^2} $
$\sinh(x)$	Sinus hyperbolicus	$ \Delta x \cdot \cosh(x) $
$\cosh(x)$	Kosinus hyperbolicus	$ \Delta x \cdot \sinh(x) $
$\tanh(x)$	Tangens hyperbolicus	$ \Delta x \cdot \frac{1}{\cosh^2 x} $
$\operatorname{arcsinh}(x)$	Arcussinus hyperbolicus	$ \frac{\Delta x}{\sqrt{1+x^2}} $
$\operatorname{arccosh}(x)$	Arcuskosinus hyperbolicus	$ \frac{\Delta x}{\sqrt{x^2-1}} $
$\operatorname{arctanh}(x)$	Arcustangens hyperbolicus	$ \frac{\Delta x}{1-x^2} $
$\operatorname{hypot}(x, y)$	gibt die Hypotenuse eines rechtwinkligen Dreiecks (Katheten $x, y$ ) zurück ( $= \sqrt{x^2 + y^2}$ )	aus Formel

## 3.4.4 Zufallszahlen

Funktion	Erklärung	Fehler $\Delta f =$
$\operatorname{random}(x), \operatorname{rnd}(x)$	Zufallszahl aus $[0..x]$	0
$\operatorname{randomint}(x), \operatorname{rndint}(x)$	ganzzahlige Zufallszahl aus $[0..x]$	0
$\operatorname{randomG}(\text{mean}, \text{std}), \operatorname{rndG}(\text{mean}, \text{std})$	Zufallszahlen nach Gauß-Verteilung (mean: Mittelwert, std: Standardabweichung)	0

## 3.4.5 Fehlerrechnung

Funktion	Erklärung	Fehler $\Delta f =$
$\operatorname{error}(x)$	gibt den Fehler des Arguments zurück ( $\operatorname{error}(20 \pm 0.5) = 0.5$ )	0
$\operatorname{relError}(x), \operatorname{relativeError}(x)$	gibt den relativen Fehler des Arguments zurück ( $\operatorname{relerror}(x \pm \Delta x) = \frac{\Delta x}{x}$ )	0

## A Syntaxdefinition

```

program:
  END                                     // END ist Eingabeende
  expression_list

expression_list:
                                     // es knne mehrere Ausdrcke, getrennt durch
                                     // PRINT=';' eingegeben werden. Ein zeilenumbruch
                                     // wirkt ebenfalls wie PRINT. Der Rckgabewert ent-
                                     // spricht dem letzten ausgewerteten Ausdruck
  expression [PRINT]                   // das letzte PRINT ist optional (wird von der
                                     // Komponente angefgt!)
  expression PRINT expression_list

expression:
  expression + term                     // Addition
  expression - term                     // Subtraktion
  term

term:
  term / primary                        // Division
  term * primary                        // Multiplikation
  term % primary                        // Modulo (nur fr Ganzzahlwerte, Fehler=0 !!!)
  primary

primary:
  NUMBER                                // ein Zahl (Formate: 5.3 -5.3 5.3e-4 5.3e4,
                                     // e/E trennt den Exponenten ab)
                                     // '.' ist Dezimaltrennzeichen, ',' hat eine andere
                                     // Bedeutung !!!
  NAME                                  // eine Konstante/Variable
  NAME = expression                    // Zuweisung eines Wertes an eine Konstante/Variable
                                     // Bsp: e = exp(1);
  - primary
  ( expression )
  | expression |                       // Absolut-Betrag
  NAME( expression )                  // Funktion mit einem Parameter, Bsp: exp(1);
  NAME( expression, expression )      // Funktion mit zwei Parameter, Bsp: logN(10, 1000);
  primary ^ primary                   // Potenz (0^x=0; x^0=1)
  primary !                           // Fakultt (nur fr Ganzzahlwerte aus [0..25], Fehler=0 !!!)
  {X, deltaX}                         // Wert mit Fehler

```