software entwicklung GmbH

# Funktionsparser

Hinweise zur Verwendung

Durch einfügen der beiden Dateien '**fnkt.h**' sowie '**fxypc.cpp**' in ein Projekt, erhält man eine neue Klasse 'UserFunction' mit der Stringausdrücke wie y=sin(x)/cos(y) oder y=x^2 berechnet werden können. Hierzu stellt die Klasse '**UserFunction**' zwei Methoden zur Verfügung: **UINT f_def(char *inbuf)** und
**long double calc(long double x, long double y)**.
Mit Hilfe der Funktions **f_def(char *inbuf)** muss zunächst der gewünschte String in Tokens umgewandelt werden. Diese verbleiben im internen Buffer der Klasse, solange die Klasse referenziert ist oder dies Funktion erneut aufgerufen wird. Mit der Methode **long double calc(long double x, long double y)** kann dann der Funktionswert berechnet werden.

## Class Members:

### UINT f_def(char *inbuf)

**Rückgabewerte:**

*TRUE:* String wurde korrekt verarbeitet.

*NO_MEM:* Nicht genügend RAM frei.

*SYNTAX_ERROR:* Der String hat nicht das richtige Eingabeformat.

**Übergabeparameter:**

*char *inbuf:* Zeiger auf den String, der die Funktionsdefinition enthält.

### long double UserFunction::calc(long double x, long double y)

**Rückgabewert:**

Der Berechnete Funktionswert an der Stelle x,y bzw. x, falls nur ein Wert übergeben wurde.

**Übergabeparameter:**

*long double x, long double y:* Die Funktionswerte aus dem Wertebereich.

## BOOL is_definded;

**is_defined** hat den Wert *TRUE,* Wen der interne Tokensbuffer gültige Werte beinhaltet, anderenfalls ist der Wert *FALSE*. Nur wenn **is_defined** den Wert *TRUE* hat, ist ein sinnvoller Aufruf von **calc(x,y)** möglich.

## Beispiel:

```
#include <fnkt.h>

....

 UserFunction myFnkt;

 char str[50] = {"2^x*(sin(x)/cos(y)-2*x*y)"};
 double z;

 myFnkt.f_def(str);

 if(myFnkt.is_defined)
{
  for(double y=0.0; x < 3.0; x += 0.01)
  {
   for(double x=0.0; x < 2.0; x += 0.01)
   {
    z=myFnkt.calc(x,y);
    ....
   }
  }
 }

....
```

## Verfügbare Operatoren:

| | |
|---|---|
| **+, -, *, / :** | Addition, Subtraktion, Multiplikation, Division. |
| **() :** | Klammern von Ausdrücken, Klammern haben die höchste Priorität. |
| **^ :** | Exponentialfunktion. |
| **sin(), cos(), tan() :** | Sinus, Cosinus, Tangens, Werte in Bogenmass. |
| **exp():** | e-Funktion. |

**sinh(), cosh(), tanh():** hyperbolische Funktionen.

**ln():** natürlicher Logarithums.

**abs():** Betrag.

**asin(), acos(), atan():** ArcusSinus, ArcusCosinus, ArcusTangens.

**sqrt():** Quadratwurzel.