

Parser/TP

$$\sqrt[5]{\sin(\alpha * \log \lambda)}$$

Parser/TP is a recursive decent expression parser unit for Turbo-Pascal, this unit includes the 5 basic math. operators (+ - * / ^), and some basic mathematical functions (sin, exp, cos, ln, log10, log2, abs, arctan, sqr, random, sqrt, tan, cotan, arcsin, arccos) and 2 constants (pi, e). Parser can be used to receive formula input from the user, and perform the necessary calculations during program runtime. Parser/TP supports a dynamic symbol table, that is created and maintained during the parser's execution. New variables are created when they are referenced for the first time, and receive the initial value of 0.0 . Variables values can be set by using the = operator (e.g. myvar = sin(anotherVar)*35), and used in user formulas.

Related Topics :

[Registration](#)

[File List](#)

[Operation](#)

[Warranty](#)

[Contact](#)

Credits :

Parser/TP was created by Ron Loewy.
Turbo Pascal is a Trademark/Copyright of Borland International.

Why Register and Registration Instructions

Parser/TP is a shareware product, if you find this product valuable, please register it. This section describes the reasons you should register.

By registering Parser/TP you will receive the complete source code to the parser unit and DLL, parser symbol table manager, and will be able to compile parser for inclusion in your protected mode and windows programs.

Your registration will help us to create the next versions of Parser/TP, that will include more options, and features, some of them might even be your enhancement requests!.

Registered Parser/TP users get full no-rotality usage permission.

Related Topics :

[Contact](#)

File List

This package contains the following files :

PARSER .INT - Parser/TP Turbo Pascal Interface Source file.
PARSER .TPU - Parser/TP Turbo Pascal Real Mode Unit.
PARSER .DOC - Parser/TP documentation file.
PARSER .REG - Parser/TP Registration File.
PARSER .DLL - Parser/TP DLL.
PARSLIB .PAS - Parser/TP import unit for DLL.
PROGRAMS.TXT - HyperAct Inc shareware products description.
CALC .PAS - Simple Parser Turbo Pascal test sample.
CALCW .PAS - Simple Parser Turbo Pascal test sample - uses DLL.
PARSEDB .TPU - Parser Symbol Table manager Module.
DBENGBAS.TPU - Parser Symbol Table manager Module.
DBENGMEM.TPU - Parser Symbol Table manager Module.
TBLBASE .TPU - Parser Symbol Table manager Module.
QUICKSRT.TPU - Parser Symbol Table manager Module.
COMBSRT .TPU - Parser Symbol Table manager Module.
MERGSORT.TPU - Parser Symbol Table manager Module.
DATEUNIT.TPU - Parser Symbol Table manager Module.
MEMTBOBJ.TPU - Parser Symbol Table manager Module.
PARSER .TPH - Parser/TP online help file in THELP/IDE format.
PARSER .HLP - Parser/TP online help file in WINHELP format.
PARSER .LIB - Parser/TP Library for use with BC++/4
PARSLIB .H - Header File for C++/DLL
CALCCW.CPP - BC++/4 C++ source to the calc program.

Related Topics :

[Contact](#)

Operation - Usage of Parser/TP in your programs.

Simply use parser unit in your program using the USES clause, when ever you want to parse a string formula do :

```
myResult := getExpr(formulaString, validity);  
if (validity = false) then  
    tellError { the formula has an error at the errAt byte }  
else  
    tellResult(myResult);
```

Please notice - the source code for the CALC program is provided, so you will probably find it easy to understand that example after you will look into the code.

Related Topics :

[Calc Example](#)

Calc Example

```
(*****
*
*                               calc                               *
* HyperAct Inc. CALC - Parser demo program.                       *
* (c) 1992, 1994 HyperAct Inc, Written By Ron Loewy.              *
* Version 2.5, Feb. 1994.                                         *
*****)
program calc;

uses
  parser
  ;

var
  exitFlag,
  validity : boolean;
  s        : string;
  result   : real;

(*****
*
*                               makeUpper                          *
* receive a string, and convert it to upper-case                  *
*****)
function makeUpper(s : string) : string;
var
  i : byte;
begin
  for i := 1 to length(s) do
    if (s[i] in ['a' .. 'z']) then
      s[i] := upCase(s[i]);
  makeUpper := s;
end; {makeUpper}

begin
  writeln('CALC V2.5, HyperAct Inc, Parser demo program');
  writeln;
  writeln('Enter mathematical expression, or QUIT/EXIT to DOS');
  writeln;
  exitFlag := false;
  repeat
    write('CALC> ');
    readln(s);
    s := makeUpper(s);
    if ((copy(s, 1, 4) = 'QUIT') or (copy(s, 1, 4) = 'EXIT')) then
      exitFlag := true
    else begin
      result := getExpr(s, validity);
      if (not validity) then
        writeln('Error in expression, at position ', errAt)
      else
        writeln('RSLT> ', result);
    end; { not a QUIT command }
```

```
until (exitFlag);  
writeln;  
writeln('Thank you for using CALC, HyperAct Inc.');
```

writeln;
end.

Related Topics :

[Operation](#)

Warranty

There is no warranty what so ever, This software package is supplied as is, The distributor (HyperAct Inc.), or the author (Loewy Ron), are not, and will not be responsible for any damages, lost profits, or inconveniences caused by the use, or inability to use this package. The use of the package is at your own risk. By using (or attempting to use) the package you agree to this.

Contact

Please contact :

HyperAct Inc,
P.O.Box 5517
Coralville IA 52241,
U.S.A

E-Mail : Compuserve - 76350,333
Phone : (319) 351-8413

To contact the author directly :

E-Mail : Compuserve - 100274,162

Related Topics :

[Registration](#)

