

Beispiel (Double Erweiterter Taschenrechner)

doubleErweiterterTaschenrechner.lex

```
%{
#include "y.tab.h"
%}
dreel    ([0-9]*[.] [0-9]+)
ereel    ([0-9]*[.] [0-9]+[eE] [+]?[0-9]+)
%%
[ \t]+   ;
[0-9]+   { sscanf(yytext,"%d",&yyval.ganzzahl); return(INTEGER); }
{dreel}  |
{ereel}  { sscanf(yytext,"%lf",&yyval.reel); return(REEL); }
[A-Za-z] { yyval.zeich=tolower(yytext[0])-'a'; return(BUCHSTABE); }
\n       |
.         return(yytext[0]);
```

doubleErweiterterTaschenrechner.yacc

```
%{
#include <stdio.h>
#include <math.h>
double reg[26];
%}
%start zeilen

%union {
    char    zeich;
    int      ganzzahl;
    double   reel;
}

%token <zeich> BUCHSTABE
%token <ganzzahl> INTEGER
%token <reel> REEL

%left '|'
%left '&'
%left '+' '-'
%left '*' '/' '%'
%left '^'
%left VORZEICHEN
```

```
%type <reel> eingabe ausdruck
%%
```

```
zeilen : eingabe '\n'
        | zeilen eingabe '\n'
        | error '\n' {yyerrok;}
        ;
```

```
eingabe : ausdruck {printf("    = %f\n",$1);}
        | BUCHSTABE '=' eingabe {$$=reg[$1]=$3;}
        ;
```

```
ausdruck: '(' ausdruck ')' {$$=$2;}
        | ausdruck '+' ausdruck {$$=$1+$3;}
        | ausdruck '-' ausdruck {$$=$1-$3;}
        | ausdruck '*' ausdruck {$$=$1*$3;}
        | ausdruck '/' ausdruck {$$=($3==0)?null():$1/$3;}
        | ausdruck '%' ausdruck {$$=fmod($1,$3);}
        | ausdruck '&' ausdruck {warnung($1,$3,'&');$$=(int)$1&(int)$3;}
        | ausdruck '|' ausdruck {warnung($1,$3,'|');$$=(int)$1|(int)$3;}
        | ausdruck '^' ausdruck {$$=pow($1,$3);}
        | '-' ausdruck %prec VORZEICHEN {$$=-$2;}
        | BUCHSTABE {$$=reg[$1);}
        | INTEGER {$$=(double)$1;}
        | REEL
        ;
```

```
%%
```

```
double null(void)
{
    printf("Fehler: Division durch 0\n");
    return(0);
}
```

```
void warnung(double op1, double op2, char z)
{
    /* fmod bestimmt von op1 und op2 den Nachkomma-Teil */
    if (fmod(op1,1)!=0 || fmod(op2,1)!=0)
        printf("Warnung: Beide Operanden zu '%c' wurden zur Berechnung "
               "in Ganzzahlen konvertiert\n", z);
}
```

```
int main(int argc, char *argv[])
{
    yyparse();
    return(0);
}
```