

acm.mipt.ru

олимпиады по программированию на Физтехе

Раздел «Язык Си» . StackBisonCalculator :

Калькулятор на Bison

Файл с правилами bison

```
/*
  file:  rpn.yy
  title: Reverse polish notation calculator rules.
*/

%{
    #define YYSTYPE double
    #include <math.h>
    #include <stdio.h>
}%

%token NUM

%% /* Grammar rules and actions follow */

input:      /* empty */
           | input line
           ;

line:       '\n'
           | exp '\n' { printf ("\t%.10g\n", $1); }
           ;

exp:        NUM                { $$ = $1; }
           | exp exp '+'       { $$ = $1 + $2; }
           | exp exp '-'       { $$ = $1 - $2; }
           | exp exp '*'       { $$ = $1 * $2; }
           | exp exp '/'       { $$ = $1 / $2; }
           /* Exponentiation */
           | exp exp '^'       { $$ = pow ($1, $2); }
           /* Unary minus */
           | exp 'n'           { $$ = -$1; }
           ;

%%
```

Файл на языке Си с функциями main и yylex

```
/*
  file:  rpn.cc
  title: Tokenizer functions
        yylex -- функция разбиения входного потока на токены.
        yyerror -- функция обработки ошибок
        main -- главная функция
*/
#include <stdio.h>
#include <ctype.h>
#include "rpn.tab.hh"
int yyparse(void);

int yyerror (const char *s) /* Called by yyparse on error */
{
    printf ("%s\n", s);
}
```

Поиск

Поиск

Раздел «Язык Си»

Главная
Зачем учить C?
Определения

Инструменты:

Поиск
Изменения
Index
Статистика

Разделы

Информация
Алгоритмы
Язык Си
Язык Ruby
Язык
Ассемблера
El Judge
Парадигмы
Образование
Сети
Objective C

Logon>>

```
/* Lexical analyzer returns a double floating point
   number on the stack and the token NUM, or the ASCII
   character read if not a number. Skips all blanks
   and tabs, returns 0 for EOF. */
int yylex (void)
{
    int c;

    /* skip white space */
    while ((c = getchar ()) == ' ' || c == '\t')
        ;
    /* process numbers */
    if (c == '.' || isdigit (c))
    {
        ungetc (c, stdin);
        scanf ("%lf", &yylval);
        return NUM;
    }
    /* return end-of-file */
    if (c == EOF)
        return 0;
    /* return single chars */
    return c;
}

int main (void)
{
    return yyparse ();
}
```

Команды компиляции

```
$ bison -d rpn.yy      # -> produce files rpn.tab.cc and rpn.tab.hh
$ gcc rpn.tab.cc rpn.cc -lm -o rpn # produce executable file rpn
```

Пример использования калькулятора:

```
$
./rpn
1 2 +
3
1 2 3 * + 100 +
107
<Ctrl+D>
$
```

-- ArtemVoroztsov - 16 Sep 2005

(с) Материалы раздела "Язык Си" публикуются под лицензией GNU Free Documentation License.