Εργαστηριακή Άσκηση 2012-2013

Υλοποίηση: Parser της γλώσσας Simon

*Αρχεία: **byson.y**
flex.l
test.txt
test2.txt

Δεν έχει υλοποιηθεί το 3ο Ερώτημα.

Ξανθόπουλος Νικήτας	4325
Γιατρά Παναγιώτα	4704
Λύκος Κάρολος	4758

bison.y

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


 int yylex(void);
 int line;
 FILE *yyin;
 void yyerror(char *errorinfo);
  int errors;
%}


%debug

%token NEW CLASS IF ELSE WHILE RETURN VOID
%token ADD DIV EGREATER EQUALS ESMALLER GREATER MOD MUL SMALLER NEQUALS
MINUS
%token LPAR RPAR LBLK RBLK LHOOK RHOOK COMMA
%token QUESTIONM /* ';' */
%token BECOMES /* '=' */
%token THEN OR AND NOT
%token PUBLIC STATIC PROTECTED PRIVATE ABSTRACT FINAL
%token ID DIGIT ARRAY
%token INTEGER CHAR  MINTEGER MCHAR NLINE

%%

eclass : CLASS ID LBLK block RBLK ;
block : | var_decl constructor meth_declaration  ;

var_decl :  var_decl ID BECOMES NEW type QUESTIONM | ID BECOMES NEW type
QUESTIONM  ;

type : CHAR | INTEGER | carray | iarray | ARRAY ;
iarray : INTEGER LHOOK MINTEGER RHOOK ;
carray : CHAR LHOOK MINTEGER RHOOK ;



constructor : scope ID LPAR parameters RPAR LBLK var_decl RBLK ;
/*public AB(int/char/intArray[]/charArray[]/,../) {var def} */


parameters : | parameter ID parameters  | COMMA parameter ID parameters ;
parameter : CHAR | INTEGER | iarray | carray | ARRAY ;

scope : PUBLIC | PROTECTED | PRIVATE | STATIC | FINAL ;
meth_declaration : meth_decl | meth_decla ;
meth_decl : scope meth_type ;
meth_type : VOID ID LPAR parameters RPAR LBLK vbody RBLK | type ID LPAR parameters
RPAR LBLK tbody RBLK ;
```

```
vbody :  statement;
tbody :  statement RETURN ID QUESTIONM ;

meth_decla : ABSTRACT VOID ID LPAR parameters RPAR LBLK RBLK | ABSTRACT type ID LPAR
parameters RPAR LBLK RBLK ;


statement : | loop statement | meth_var statement | expression oper expression statement ;
meth_var : ID BECOMES MINTEGER QUESTIONM | ID BECOMES MCHAR QUESTIONM ;
/* var_def | var_decl | meth_decl */
loop : if_express | while_express ;

if_express : IF LPAR condition RPAR LBLK statement RBLK | IF LPAR condition RPAR LBLK
statement RBLK ELSE LBLK statement RBLK ;
while_express : WHILE LPAR condition RPAR LBLK statement RBLK ;

condition : expression oper expression | expression oper expression oper condition |
expression  ;
expression :  ID  | MINTEGER | iarray | carray | ARRAY  ;

oper : boper | loper | aroper | expression ;
aroper : ADD | DIV | MINUS | MOD | MUL ;
boper : AND | NOT | OR ;
loper : EQUALS | GREATER | SMALLER | ESMALLER | EGREATER | NEQUALS ;



%%



main(int argc, char *argv[])
        {
                ++argv;
                --argc;
                errors=0;
                if (argv>0)
                {
                printf("\n\n");
                        yyin=fopen(argv[0],"r");
                        yydebug=0;
                        yyparse();
                printf("\n\n");
                }

                if(errors==0)
                {
                        printf("\n");

        printf("************************************************\n");
                        printf("                    No Errors\n");
                        printf("************************************************
\n\n");
                }
        }
```

```
void yyerror(char *msg)
        {       errors++;
                printf("\n**************************************************\n");
                printf("Error at line %d: %s\n",line, msg);
                printf("**************************************************\n\n");
        }
```

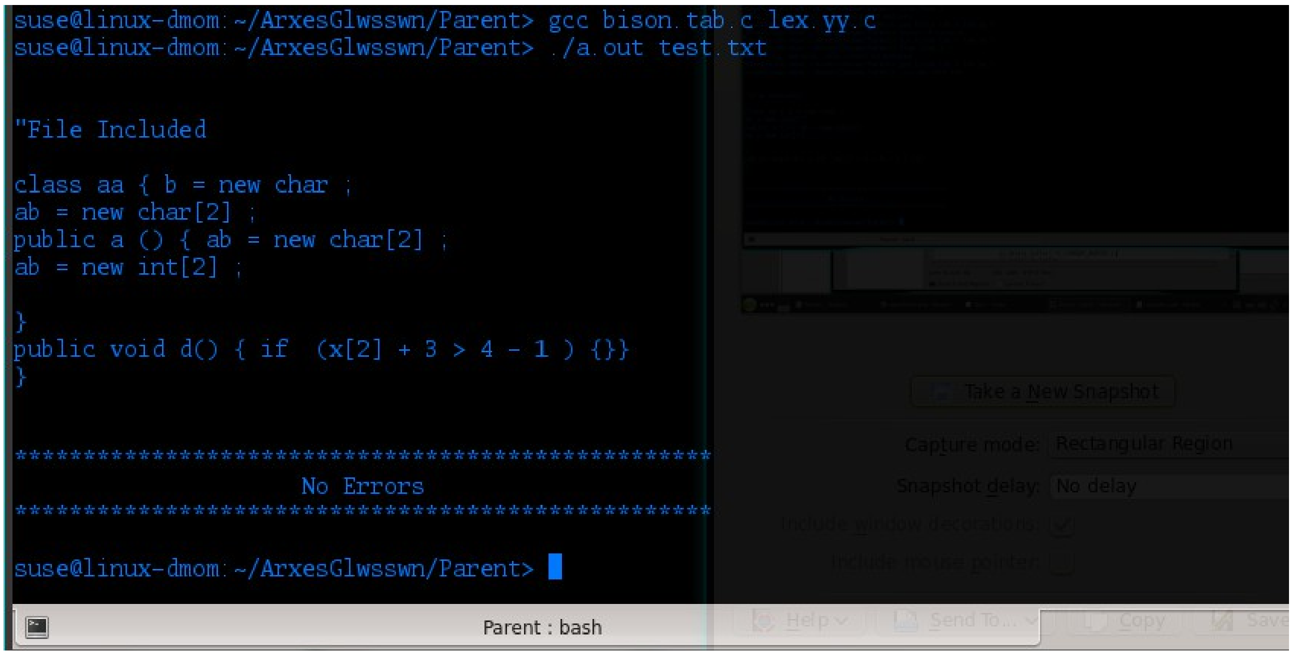
flex.l

```
%x incl
%{
#include <math.h> /*atof() */
#include "bison.tab.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int line;
int line_num;

 #define MAX_INCLUDE_DEPTH 10
        YY_BUFFER_STATE include_stack[MAX_INCLUDE_DEPTH];
        int include_stack_ptr = 0;

%}
%option noyywrap
%option yylineno
%option stack

DIGIT  [0-9]
SCHAR [\"\'\0\\\t]+
MCHAR {SCHAR}?[a-zA-Z_]


%x comment


%% /* http://flex.sourceforge.net/manual/Multiple-Input-Buffers.html */
#include BEGIN(incl);


        <incl>[ \t ]*      /* eat the whitespace */
        <incl>[^ \"\t\n]+   { /* got the include file name */
               if ( include_stack_ptr >= MAX_INCLUDE_DEPTH )
                   {
                   fprintf( stderr, "Includes nested too deeply" );
                   exit( 1 );
                   }

               include_stack[include_stack_ptr++] =
                   YY_CURRENT_BUFFER;

               yyin = fopen( yytext, "r" );

               if ( ! yyin )
                   printf("Error opening include file\n");
                    else printf("File Included\n\n");
               yy_switch_to_buffer(
                   yy_create_buffer( yyin, YY_BUF_SIZE ) );
 BEGIN(INITIAL);
              }

       <<EOF>> {
              if ( --include_stack_ptr == 0 )
                 {
                 yyterminate();
                 }

              else
                 {
                 yy_delete_buffer( YY_CURRENT_BUFFER );
                 yy_switch_to_buffer(
                      include_stack[include_stack_ptr] );
                 }
              }



int line_num = 1; /*
http://www.softlab.ntua.gr/facilities/documentation/unix/gnu/flex/flex_11.html */
"/*"         BEGIN(comment);

<comment>[^*\n]*        /* eat anything that's not a '*' */
<comment>"*"+[^*/\n]*   /* eat up '*'s not followed by '/'s */
<comment>\n             ++line_num;
<comment>"*"+"/"        BEGIN(INITIAL);

"\n"              {printf("%s", yytext); ++line; }
("+"|"-")?{DIGIT}+ { printf("%s", yytext); return MINTEGER ;}

"char"            { printf("%s", yytext); return CHAR ;}
"integer"                  { printf("%s", yytext); return INTEGER ;}
"new"           {  printf( "%s", yytext); return NEW ;}
"class"         {  printf( "%s", yytext); return CLASS ;}
"if"            {  printf( "%s", yytext); return IF ;}
"else"          {  printf( "%s", yytext); return ELSE ;}
"while"         {  printf( "%s", yytext); return WHILE ;}
"return"        {  printf( "%s", yytext);  return RETURN ;}
"void"          {  printf( "%s", yytext); return VOID ;}

"public"        {  printf("%s", yytext); return PUBLIC ;}
"private"       {  printf("%s", yytext); return PRIVATE ;}
"protected"     {  printf("%s", yytext); return PROTECTED ;}
"static"        {  printf("%s", yytext); return STATIC ;}
"abstract"      {  printf("%s", yytext); return ABSTRACT ;}
"final"         {  printf("%s", yytext); return FINAL ;}

[a-zA-Z_][a-z_A-Z0-9]*"["{DIGIT}+"]"    {printf("%s", yytext); return ARRAY;}

{MCHAR}[a-z_A-Z0-9]*               { printf("%s" , yytext); return ID ;}

";"             {  printf("%s", yytext); return QUESTIONM ;}
"="             {  printf("%s", yytext); return BECOMES ;}
","             {  printf("%s", yytext); return COMMA ;}
"+"             {  printf("%s", yytext); return ADD ;}
"-"             {  printf("%s", yytext); return MINUS ;}
"*"             {  printf("%s", yytext); return MUL ;}
```

```
"/"           {  printf("%s", yytext); return DIV ;}
"%"            {  printf("%s", yytext); return MOD ;}
"("           {  printf("%s", yytext); return LPAR ;}
")"           {  printf("%s", yytext); return RPAR ;}
"["           {  printf("%s", yytext); return LHOOK ;}
"]"           {  printf("%s", yytext); return RHOOK ;}
"{"           {  printf("%s", yytext); return LBLK ;}
"}"           {  printf("%s", yytext); return RBLK ;}
"=="            {  printf("%s", yytext); return EQUALS ;}
"!="           {  printf("%s", yytext); return NEQUALS ;}
">"           {  printf("%s", yytext); return GREATER ;}
"<"           {  printf("%s", yytext); return SMALLER ;}
">="            {  printf("%s", yytext); return EGREATER ;}
"<="            {  printf("%s", yytext); return ESMALLER ;}
"||"          {  printf("%s", yytext); return OR ;}
"&&"           {  printf("%s", yytext); return AND ;}
"!"           {  printf("%s", yytext); return NOT ;}


%%
```

Αποτελέσματα

Με αρχείο εισόδου ένα txt με κώδικα που αναμένουμε να αναγνωρίζεται από την υποθετική γλώσσα:



```
suse@linux-dmom:~/ArxesGlwsswn/Parent> gcc bison.tab.c lex.yy.c
suse@linux-dmom:~/ArxesGlwsswn/Parent> ./a.out test.txt


"File Included

class aa { b = new char ;
ab = new char[2] ;
public a () { ab = new char[2] ;
ab = new int[2] ;

}
public void d() { if  (x[2] + 3 > 4 - 1 ) {}}
}


*****************************************************
                      No Errors
*****************************************************

suse@linux-dmom:~/ArxesGlwsswn/Parent>
```

Αρχείο test.txt

```
#include "test2.txt"                                @συμπεριλαμβάνουμε ένα αρχείο test2.txt
 class aa { b = new char ;                          @δημιουργεία κεντρικής κλάσης
               ab = new char[2] ;                   @δήλωση μεταβλητών (και array)
               public a () { ab = new char[2] ; @constructor και
               ab = new int[2] ;                    @δήλωση μεταβλητών μέσα στον constructor
               /* s */                              @σχόλια C
        }
        public void d() { if  ( x[2] + 3 > 4 - 1 ) {}}  @δήλωση μεθόδων με χρήση υποθετικού
                                                         @scope καθώς και χρήση if
                                                         @με λογικες και αριθμητικές πράξεις
        }
```

Αρχείο test2.txt

```
 class aa { b = new char ;
        ab = new char[2] ;
        public a () { ab = new char[2] ;
        ab = new int[2] ;
        /* s */
        }
        public void d() { if  (x[2] + 3 > 4 - 1 ) {}}
}
```