

Modelo Análise Sintática + Semântica - prog(), decl_list_var() e decl_var()

terça-feira, 3 de dezembro de 2024 19:20

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include "Analex.h"

void prog() {
    escopo_atual = GBL;
    tk = analex();
    while ((tk.cat == PAL_RESERV) &&
           ((tk.codigo == CONST) || (tk.codigo == INT) || (tk.codigo == REAL) ||
            (tk.codigo == CHAR) || (tk.codigo == BOOL))) {
        decl_list_var();
    }
    while ((tk.cat == PAL_RESERV) &&
           ((tk.codigo == PROT) || (tk.codigo == DEF))) {
        decl_def_prot();
    }
    if (tk.cat != END_FILE) erro("Declaração ou definição de procedimento esperado!");
}

void decl_list_var() {

    bool eh_const;
    int tipo;

    eh_const = ((tk.cat == PAL_RESERV) && (tk.codigo == CONST));
    if (eh_const) tk = analex();
    if (!((tk.cat == PAL_RESERV) && ((tk.codigo == INT) || (tk.codigo == REAL) ||
        (tk.codigo == CHAR) || (tk.codigo == BOOL)))) erro("Tipo esperado.");

    tipo = tk.codigo;
    decl_var(tipo);
    tk = analex();
    while ((tk.cat == SINAL) && (tk.codigo == VIRGULA)) {
        tk = analex();
        decl_var(tipo, eh_const);
    }
}

void decl_var(int tipo, bool eh_const) {

    int k, i;
    bool eh_array;
    int cont_dim = 0;
    int tam_dims[MAX_ARRAY_DIM];
    int catTk;

    if (tk.cat != IDENT) erro("Identificador esperado");
    k = consulta_tab_simb(tk.lexema);
    if ((k != -1) && (ts[k].linha.escopo == escopo_atual)) erro ("Redeclaração de identificador");
    k = insere_tab_simb(tk.lexema, escopo_atual); // parte fixa
    tk = analex();
    eh_array = ((tk.cat == SINAL) && (tk.codigo == ABRE_COLCH));
    if (eh_array) {
        while ((tk.cat == SINAL) && (tk.codigo == ABRE_COLCH)) {
            cont_dim++;
            if (cont_dim > MAX_ARRAY_DIM) erro("Número de dimensões do array maior que limite!");
            tk = analex();
            if (tk.cat == CT_I) {
                tam_dims[cont_dims-1] = tk.valor;
                tk=analex();
            }
        }
        else if (tk.cat == IDENT) {
            w = consulta_tab_simb(tk.lexema);
            if (w == -1) erro("Identificado constante dimensão do array não declarado");
        }
    }
}
```

```
if (ts[w].linha.eh_const == NAO) erro ("Identificador dimensão array deve ser constante");
if (ts[w].linha.tipo != INT) erro ("Identificador dimensão array deve ser do tipo inteiro");
tam_dims[cont_dims-1] = ts[w].linha.valor_const.inteiro;
tk=analex();
}
else erro("Dimensão de array identificador constante inteiro ou constante inteira esperada");
if ((tk.cat != SINAL) && (tk.codigo == FECHA_COLCH)) erro("Fecha colchete esperado");
tk = analex();
}
insere_tab_simb_decl_var_array(k, tipo, (escopo_atual == GBL ? VG : VL), cont_dim, tam_dims, eh_const); // parte variável
}
else {
    insere_tab_simb_decl_var_escalar(k, tipo, (escopo_atual == GBL ? VG : VL), VS, eh_const);
}
if (eh_const)
    if (!((tk.cat == SINAL) && (tk.codigo == ATRIB))) erro("Declaração de identificador constante necessita inicialização");
if ((tk.cat == SINAL) && (tk.codigo == ATRIB)) {
    // processa sintaxe de inicialização de variável simples ou de array (atualiza tab. simb para valor de ident. constante)
    switch (tipo) {
        case INT: catTk = CT_I; break;
        case REAL: catTk = CT_R; break;
        case CHAR: catTk = CT_C; break;
        default: ;
    }
    if (eh_array) {
        tk = analex();
        if (!(tk.cat == SINAL) && (tk.codigo == ABRE_CHAVE)) erro("Incialização de array inválida");
        tk = analex();
        while (1) {
            if (tk.cat != catTk) erro("Tipo de dado de inicialização de array inválido");
            tk = analex();
            if ((tk.cat == SINAL) && (tk.codigo == VIRGULA)) {
                tk = analex();
                continue;
            }
            else if ((tk.cat == SINAL) && (tk.codigo == FECHA_CHAVE)) {
                break;
            }
            else erro("Vírgula ou fecha parênteses esperado");
        }
    }
    else {
        tk = analex();
        if (tk.cat != catTk) erro("Tipo de dado de inicialização de varável escalar inválido");
    }
}
tk = analex();
}
```

