

## Modelo Análise Sintática - prog(), decl\_list\_var() e decl\_var()

terça-feira, 3 de dezembro de 2024 19:17

```
#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, eh_const);
    tk = analex();
    while ((tk.cat == SINHAL) && (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];

    if (tk.cat != IDENT) erro("Identificador esperado");
    k = insere_tab_simb(tk.lexema, escopo_atual); // parte fixa
    tk = analex();
    eh_array = ((tk.cat == SINHAL) && (tk.codigo == ABRE_COLCH));
    if (eh_array) {
        while ((tk.cat == SINHAL) && (tk.codigo == ABRE_COLCH)) {
            cont_dim++;
            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);
                tam_dims[cont_dims-1] = ts[w].linha.valor_const.inteiro;
                tk=analex();
            }
            else erro("Dimensão de array identificador constante inteiro ou constante inteira espe:");
            if ((tk.cat != SINHAL) && (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_dir
    }
}
```

```

else {
    insere_tab_simb_decl_var_escalar(k, tipo, (escopo_atual == GBL ? VG : VL), VS, eh_const);
}
if ((tk.cat == SINAL) && (tk.codigo == ATRIB)) {
    if (eh_array) {
        tk = analex();
        if (!(tk.cat == SINAL) && (tk.codigo == ABRE_CHAVE)) erro("Incialização de array invál:");
        tk = analex();
        while (1) {
            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();
    }
}
}

```