CATEGORIES

def ,expr, sentence, type

NODES

program -> AST*;

func:def-> name:string parameter* retorno:type defVar* sentence*;

defVar:def -> name:string type;

parameter:def -> name:string type;

defStruct:def -> name:string parameter*;

intType:type -> ;

realType:type -> ;

charType:type -> ;

arrayType:type -> index:int type;

structType:type -> name:string;

voidType:type -> ;

print:sentence -> string expr ;

read:sentence ->expr ;

assignment:sentence -> left:expr right:expr;

ifSentence:sentence -> condition :expr iftrue:sentence*;

ifElseSentence:sentence -> condition:expr iftrue:sentence* else1:sentence*;

whileSentence:sentence -> condition:expr sentence*;

returnNode:sentence -> expr;

funcCall:sentence -> name:string args:* ;

exprAritmetica:expr -> left:expr op:string right:expr ;

exprLogica:expr -> left:expr op:string right:expr ;

exprLogicaNe:expr -> expr;

acces:expr -> left:expr right:string ;

arrayAcces:expr -> left:expr right:expr ;

cast:expr -> typeToConvert:type expr ;

litEnt:expr -> string ;

litReal:expr -> string ;

litChar:expr -> string ;

variable:expr -> string ;

methodCallExpr:expr -> name:string args:expr* ;