

Caspal v0.0.1α

Palavras Reservadas

A linguagem Caspal é baseada na estrutura do Pascal com algumas modificações, onde as palavras reservadas são descritas de forma com que a primeira sílaba vá para o final da palavra. Por exemplo, a palavra integer em Caspal é tegerin.

São palavras reservadas do Caspal:

grampro, rav, ginbe, ned, rof, od, fi, enth, seel, ot, adre e tewri.

Tipos de Variáveis

Existem três tipos de variáveis, inteiros, booleanos e strings, e são representados como: tegerin, leanboo e ingstr.

É possível também construir arrays uni e multidimensionais em cima de cada tipo de variável, adicionando '[' e ']' no final da declaração:

tegerin[], leanboo[][] e ingstr[][]...[].

O acesso aos arrays é dado por `var_arr[0]`, por exemplo.

Variáveis do tipo tegerin devem receber valores inteiros, note, não é possível atribuir valores que começam com 0 e que sejam diferentes de 0. Por exemplo é possível fazer `var_a = 0`; mas não `var_b = 01`;

Variáveis do tipo leanboo devem ser *Etru* e *Sefal*.

Variáveis do tipo ingstr devem estar entre aspas duplas. Por exemplo, `var_a = "Hello"`;

Operações

É possível realizar operações lógicas e aritméticas entre constantes e variáveis, além de realizar atribuições.

Como operadores aritméticos, temos:

+, -, * e /.

Como operadores lógicos, temos:

>, <, !=, si, ro, nad e ton.

A Atribuição de variáveis pode ser feita com o símbolo =.

Identificadores

É possível nomear variáveis, nomes de métodos e o nome do programa com qualquer letra do alfabeto, minúscula ou maiúscula, além do símbolo `_` seguido dos mesmos caracteres, com adição de números. Ou seja, é impossível iniciar um identificador com um dígito.

Separadores

Em Cascal, os seguintes separadores são utilizados:

`., : ; e ,.`

O separador 'ponto-e-virgula' é utilizado no final de cada linha de operação (atribuição, chamada de função, etc).

O Separador 'dois pontos' é utilizado na etapa de declaração de variável, tanto após o comando *var*: quanto após cada identificação de variável do tipo *nome: <tipo>*.

O Separador 'vírgula' é utilizado na etapa de identificação de variável, quando se deseja que mais de uma variável seja do mesmo tipo. Por exemplo:

var:

var_1, var_2, ..., var_n: tegerin

Por fim, o separador 'ponto' é utilizado apenas para identificar o final do programa, logo após a última instrução *ned*.

Note: Nas versões iniciais do compilador, a etapa de análise léxica considera espaço e fim de linha como um separador, então é obrigatório separar cada símbolo da linguagem por um espaço.

Gramática Oficial do Caspal v.0.0.1α

S → grampro <ID>; ginbe <PROGRAM> ned.
grampro <ID>; rav '\n'<RAV> ginbe <PROGRAM> ned.
<RAV> → <ID_LIST>: <DECLVAR> | <ID_LIST>: <DECLVAR> '\n' <RAV>
<ID_LIST> → <ID> | <ID>, <ID_LIST>
<DECLVAR> → tegerin<ARR> | leanboo<ARR> | ingstr<ARR>
<ARR> → []<ARR> | ε
<OFFSET_ARR> → [<EXP>]
<ID> → <ID_PREFIX> <OFFSET_ARR> | <ID_PREFIX>
<ID_PREFIX> → a<ID_SUFFIX> | b<ID_SUFFIX> | ... | z<ID_SUFFIX> | A<ID_SUFFIX> | ... |
Z<ID_SUFFIX> | _<ID_SUFFIX>
<ID_SUFFIX> → a<ID_SUFFIX> | b<ID_SUFFIX> | ... | z<ID_SUFFIX> | A<ID_SUFFIX> | ... |
Z<ID_SUFFIX> | _<ID_SUFFIX> | 0<ID_SUFFIX> | ... | 9<ID_SUFFIX> | ε
<PROGRAM> → <ATTR> | <LOOP> | <COND> | <IO> | ε
<ATTR> → <ID> := <EXP>; '\n' <PROGRAM>
<EXP> → <EXP> <OP> <EXP> | <ID> | <CONST> | <NOT_OP><EXP>
<OP> → + | - | * | / | > | < | si | != | nad | ro
<CONST> → <INTEGER> | <BOOLEAN> | "<STRING>"
<INTEGER> → <NUMBERS> | -<NUMBERS>
<NUMBERS> → 1<NUMBERS_SUFFIX> | ... | 9<NUMBERS_SUFFIX>
<NUMBERS_SUFFIX> → 0<NUMBERS_SUFFIX> | ... | 9<NUMBERS_SUFFIX> | ε
<BOOLEAN> → Sefal | Etru
<STRING> → a<STRING> | b<STRING> | ... | z<STRING> | A<STRING> | ... |
Z<STRING> | _<STRING> | 0<STRING> | ... | 9<STRING> | <STRING> | ε
<LOOP> → rof <ATTR> ot <INTEGER> od '\n' ginbe <PROGRAM> ned; '\n' <PROGRAM>
<COND> → fi <EXP> enth <PROGRAM><ELSE> '\n' <PROGRAM>
<ELSE> → seel <PROGRAM> ned; | ned;
<IO> → tewri <EXP>; '\n' <PROGRAM> | <ID> := adre <STRING>; '\n' <PROGRAM>
<NOT_OP> → ton