Declarações - Return

Professor:
Christiano Braga
Aluno:
Allan Patrick De Freitas Santana

Bnf -

⟨start⟩ ::= [ds:decSeq] [cs:cmd_seq]

⟨exp⟩ ::= ⟨arr⟩ | ⟨bin_exp⟩ | ⟨un_exp⟩ | ⟨paren_exp⟩ | ⟨call⟩ | ⟨atom⟩

⟨arr⟩ ::= '[' ',' % ⟨exp⟩ * ']' | ⟨identifier⟩ '[' ⟨exp⟩ ']' | 'concat' '(' ⟨exp⟩ ',' ⟨exp⟩ ')' | 'append' '(' ⟨exp⟩ ',' ⟨ exp⟩ ')'

⟨paren_exp⟩ ::= '(' ⟨exp⟩ ')'

⟨bin_exp⟩ ::= ⟨exp⟩ ⟨binop⟩ ⟨exp⟩

⟨un_exp⟩ ::= 'len' '(' ⟨exp⟩ ')' | 'not' ⟨exp⟩

⟨atom⟩ ::= ⟨num_atom⟩ | ⟨truth_atom⟩ | ⟨id_atom⟩

⟨num_atom⟩ ::= ⟨number⟩

⟨truth_atom⟩ ::= ⟨truth⟩

⟨id_atom⟩ ::= ⟨identifier⟩

⟨binop⟩ ::= 'and' | 'or' | '==' | '<=' | '>=' | '<' | '>' | '+' | '-' | '*' | '/'

⟨truth⟩ ::= 'True' | 'False'

⟨identifier⟩ ::= /(?!\d)\w+/

⟨number⟩ ::= /\d+/

⟨blk⟩ ::= '{' ⟨decSeq⟩ ⟨cmd_seq⟩ '}'

⟨cmd_seq⟩ ::= ',' % ⟨atom_cmd⟩ +

⟨atom_cmd⟩ ::= ⟨ret⟩ | ⟨cond⟩ | ⟨loop⟩ | ⟨assign⟩ | ⟨print⟩ | ⟨call⟩ | ⟨skip⟩

⟨ret⟩ ::= 'return' ⟨exp⟩

⟨skip⟩ ::= ‘skip’

⟨assign⟩ ::= ⟨identifier⟩ ’:=’ ⟨exp⟩ | ⟨identifier⟩ ’[‘ ⟨exp⟩ ’]’ ‘:=’ ⟨exp⟩

⟨print⟩ ::= ‘print’ ⟨exp⟩

⟨cond⟩ ::= ‘if’ ⟨paren_exp⟩ ⟨blk⟩ [‘else’ ⟨blk⟩]

⟨loop⟩ ::= ‘while’ ⟨paren_exp⟩ ⟨blk⟩

⟨dec⟩ ::= ⟨fn⟩ | ⟨rec⟩ | ⟨var⟩ | ⟨const⟩

⟨decSeq⟩ ::= ⟨dec⟩ *

⟨var⟩ ::= ‘var’ ‘,’%{ ⟨identifier⟩ ‘=’ ⟨exp⟩ }⁺ ‘;’

⟨const⟩ ::= ‘const’ ⟨identifier⟩ ‘=’ ⟨exp⟩ ’;’

⟨fn⟩ ::= ‘def’ ⟨identifier⟩ '(' ⟨formal⟩ ')' ⟨blk⟩

⟨rec⟩ ::= ‘rec’ ⟨identifier⟩ '(' ⟨formal⟩ ')' ⟨blk⟩

⟨formal⟩ ::= ','% ⟨identifier⟩ *

⟨call⟩ ::= ⟨identifier⟩ '(' ⟨actual⟩ ')'

⟨actual⟩ ::= ','% ⟨exp⟩ *

Especificações ebnf

1. Foram feitas as seguintes alterações

- Foi adicionado a keyword return.
- A seguinte produção foi alterada exp e atom_cmd.
- AS produções alteradas ficaram da seguinte forma:
    - exp = arr | bin_exp | un_exp | paren_exp | call | @:atom ;
    - atom_cmd = ret | cond | loop | assign | print | call | skip;
- Foi adicionado uma nova produção atom_cmd.
- A nova produção tem o seguinte formato:
    - ret = “return” exp;

BNF -> Impiler:

Conforme o impiler percorre a bnf, as expressões correspondentes terão suas classes no impiler chamadas. As funções a serem chamadas no impiler são mapeadas pelo nome, utilizando o termo a esquerda da expressão no BNF

ret- Quando ocorre a aparição de um return no programa a função ret é chamada e em seguida é feita a chamada da função de return do pi.

Especificações impiler

1. Foram feitas as seguintes alterações

- Criação da função ret para tratar do caso de return;

Impiler-> pi:

- **As funções a seguir são chamadas pelo impiler:**
- A classe return trata os valores passados ao return;

Especificações pi

1. Foram feitas as seguintes alterações

- Criação da classe Return para tratar o caso do return;
- Criação das funções de avaliação para empilhar e desempilhar as novas instruções, as funções estão listadas a seguir:
    - __evalReturn
    - __evalReturnKw