

Legenda para o uso da BNF

1. **Símbolos não-terminais:** Estão entre **<** e **>**. Exemplo: **<programa>**, **<comando>**.
2. **Símbolos terminais:** Estão em **negrito** e representam tokens da linguagem, ou seja, palavras-chave, operadores, e outros símbolos literais. Exemplo: **program**, **procedure**, **while**, **+**, **;**.
3. **Produções:** Cada produção é indicada pelo símbolo **::=**, que significa "pode ser substituído por". Exemplo: **<programa> ::= program <identificador> ; <corpo>**.
4. **Conteúdo opcional:** Colocado entre colchetes **[]**, indica elementos que podem aparecer zero ou uma vez. Exemplo: **[<declaração de variáveis>]**.
5. **Conteúdo repetido:** Colocado entre chaves **{ }**, indica elementos que podem aparecer zero ou várias vezes. Exemplo: **{ <comando> }**.
6. **Operadores e Expressões:** Os operadores aritméticos e relacionais, como **+**, **-**, *****, **/**, **==**, **!=**, **<**, **<=**, **>**, **>=**, estão representados na gramática e indicam operações permitidas nas expressões.
7. **Identificadores e constantes:** **<identificador>** representa nomes de variáveis, funções e procedimentos definidos pelo usuário, enquanto **<número>** representa valores numéricos.

Programa e Corpo

<programa> ::= programa <identificador> ; <corpo>

<corpo> ::= [<declaração de variáveis>] [<declaração de sub-rotinas>] <comandos>

Declarações

<declaração de variáveis> ::= **<tipo>** <lista de identificadores> ;

<tipo> ::= **int** | **bool**

<lista de identificadores> ::= <identificador> { , <identificador> }

<declaração de sub-rotinas> ::= <declaração de procedimento> | <declaração de função>

<declaração de procedimento> ::= **procedure** <identificador> ([<lista de parâmetros>]) { <corpo> }

<declaração de função> ::= **function** <identificador> ([<lista de parâmetros>]) : <tipo> { <corpo> <comando retorno> }

<lista de parâmetros> ::= <parâmetro> { , <parâmetro> }

<parâmetro> ::= <tipo> <identificador>

Comandos

<comandos> ::= <comando> { <comando> }

<comando> ::= <comando atribuição> | <chamada de procedimento> | <comando condicional> | <comando laço> | <comando retorno> | <comando impressão>

<comando atribuição> ::= <identificador> = <expressão> ;

<comando laço> ::= **while** (<expressão>) { <comandos> { <comando desvio> } }

<comando condicional> ::= **if** (<expressão>) { <comandos> } [**else** { <comandos> }]

<comando laço> ::= **while** (<expressão>) { <comandos> }

<comando retorno> ::= **return** <expressão> ;

<comando desvio> ::= **break** ; | **continue** ;

<comando impressão> ::= **print** (<expressão>) ;

Expressões

<expressão> ::= <expressão simples> [<operador relacional> <expressão simples>]

<operador relacional> ::= (== | != | < | <= | > | >=)

<expressão simples> ::= [+ | -] <termo> { (+ | -) <termo> }

<termo> ::= <fator> { (* | /) <fator> }

<fator> ::= <identificador> | <número> | <chamada de função> | (<expressão>) |

true | **false**

<chamada de função> ::= <identificador> ([<expressão> { , <expressão> }])

Números e identificadores

<identificador> ::= <letra> { <letra> | <dígito> | _ }

<número> ::= <dígito> { <dígito> }

<dígito> ::= **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9**

<letra> ::= **a** | **b** | **c** | ... | **z** | **A** | **B** | ... | **Z**