Dicionario gramatica compiladores

Main

O programa irá iniciar desta maneira:

void main{
    <declaração de variavel>
    <declaração de função>

inicio;
    <corpo>

fim
}

```
void main{
    inicio;

    fim
}
```

Comentarios:

Comentarios são para descrever a situação ou lembrar algo que tem que ser feito
Comentarios de linhas unicas serão feitos com //
Comentarios de blocos serão feitos com /* e fechados com */

```
// comentario de linha
/*
```

```
comentarios de linhas
*/
```

Tipos de dados:

São aceitos os seguintes tipos de dados: integer para inteiros (1,2,3), char para caracteres (a,b,c), string para cadeia de caracteres (soma, resultado), real para números reais (1.2, 3.4, 5.7).

Os tipos são:

- integer
- float (ponto flutuante, até duas casas após o ponto)
- string
- char

Declaração de variável

Primeiro é declarado o nome da variável e depois o tipo, separados por dois pontos e finalizando com ponto e virgula, exemplo: <nome da variavel> : <tipo>;

```
a : float;
b, c : char;
d : integer; e : string;
```

Atribuição de variável

Para atribuir uma variável é necessário declara-lá e depois atribuir o valor a ela, exemplo:/*

<ome da variavel> : <tipo>;

<nome da variavel> = <valor da variavel>

```
a: integer;
```

```
a = 20;
```

Operações aritméticas

Adição: +
Subtração: -
Multiplicação: *
Divisão: /

```
a = 1 + 2; //adição
b = 1 - 2; //subtração
a = 1 * 2; //multiplicação
b = 1 / 2; //divisão
```

Operadores de comparação e atribuição

Operadores de comparação comparam termos. Os operadores de comparação são:
Igual: ==.
Diferente: !=.
Maior que: >.
Menor que: <.
Maior ou igual que: >=.
Menor ou igual que: <=.
Exemplos:
<nome da variavel> <operador> <nome da variavel>

```
a == b
//Verdadeiro (TRUE) se a é igual a b.
a != b
//Verdadeiro se a não é igual a b.
a > b
//Verdadeiro se a é maior que b.
a < b
```

```
//Verdadeiro se a é menor que b.
a >= b
//Verdadeiro se a é maior ou igual a b.
a <= b
//Verdadeiro se a é menor ou igual a b.
```

Estruturas de controle

Estruturas de controle são usadas para controlar o fluxo de execução do programa.
As estruturas de controle são:

- if/else
- do/while
- while
- for

Exemplo if/else:

if (<nome da variavel> <operador> <nome da variavel>){
  <comando>
}else{
  <comando>
}

```
if (a > b){
    cout << "A maior que B";
}
else{
    cout << "A menor que B";
}

//Exemplo do if sem o else
if(a > b){
```

```
    cout << "A maior que B";
}
```

Exemplo do/while

do{
  <comando>
}while(<nome da variavel> <operador> <nome da variavel>)

```
do{
    cout << "teste";
}while(a < b)
```

Exemplo while

while(<nome da variavel> <operador> <nome da variavel>)

```
while(a < b){
    cout << "teste";
}
```

Exemplo for

for (<nome da variavel> = <valor inicial>; <nome da variavel> <operador> <valor final>; <incrementador>){
  <comando>
}

```
for (i = 0; i < 10; ++1){
    cout << "teste";
}
```

Declaração de função

Funções são usadas para separar blocos de código e torna-lo mais legivel, para declarar uma função, usamos este formato:

<tipo de retorno> <nome da função> (<tipo da variavel>: <nome da variavel>){

inicio;

<comando>

fim

return(<valor>)

}

```
integer funcaoTeste( integer var) {
    inicio;
    fim
    return(1)
}

integer funcaoTesteDoisParametros( integer var; integer varDois)
    inicio;
    fim
    return(1)
}

integer funcaoTesteSemParametro {
    inicio;
    fim
    return(1)
}
```

Chamada de função

Agora para chamar as funções declaradas, usamos o seguinte formato:

callfuncao <nome da variavel> (<parametros>)

```
//funcao com parametro
callfuncao soma(1,2);
//funcao sem parametros
callfuncao retornaVazio;
```

Incremento e decremento

Para incrementar ou decrementar uma variável, usamos o ++ <numerointeiro> ou - - <numerointeiro>

Pode apenas ser usado dentro de um for loop

```
++ 3
-- 4
```

Saída e entrada de dados

Para mostrar no terminal ou atribuir uma variável no terminal, usamos o cin e o cout

```
cout << "printando"

cin >> <nome da variavel>
```

Regras Léxicas

1. Números inteiros podem ir de 0 até 2000
2. Números com ponto flutuante vão de 0 até 2000
3. Strings podem ter no máximo 20 caracteres
4. O nome das variáveis contém apenas caracteres, com no máximo 20 letras, não podem ser usados caracteres especiais e nem números

5. Char e string devem ser dentro de aspas simples
6. Literais devem ser dentro de aspas duplas

Erros Léxicos

1. Números inteiros ou com ponto flutuante fora da faixa definida
2. Se o número com ponto flutuante conter outro carácter para separar o inteiro do decimal se não o ponto
3. Se a variável conter mais de 20 letras, carácter especial, número
4. Se algum dos tipo char, string forem dentro de aspas duplas ou literais dentro de aspas simples
5. Não fechar as aspas que foram abertas