Universidade do Extremo Sul Catarinense - UNESC

Curso: Ciência da Computação

Disciplina: Compiladores

Manual:

Linguagem de programação

LJ++

Equipe: Lucas Orestes Fabris e Jean Carlos Ricken

Criciúma, Agosto de 2022

**Terminais**

1. while
2. void
3. string
4. return
5. integer
6. float
7. var
8. char
9. cin
10. main
11. cout
12. If
13. else
14. for
15. do
16. !=
17. =>
18. =
19. >
20. <
21. >=
22. <=
23. +
24. –
25. \*
26. /
27. \
28. ;
29. :
30. (
31. )
32. {
33. }
34. [
35. ]
36. |
37. &
38. $
39. `
40. Î
41. >>
42. <<
43. ++
44. –
45. ,
46. .

**Tipos de dados**

São aceitos os tipos de dados:

Integer para declaração de números inteiros (1, 2, 3), Sem limite de números.

Float para números quebrados, seu separador é ‘.’ e o limite de casas decimais são 4 após o ponto (1.25, 4.00, 10.50), sem limite de números antes e depois do ponto.

String para palavras ou números (%%palavra%%), Sem limite de caracteres.

Char para caracteres separados por hashtag (‘a’, ‘b’, ‘c’). Limite de 1 caractere.

Literal são strings e caracteres que iniciam com % e fecham% (%%palavra%%), Sem limite de caracteres

**Estrutura**

O projeto se inicia com void main, será aberto com chaves {e dentro da chaves declaramos uma variável, suas respectivas funções, o seu corpo e finalizamos com} terminando com a declaração ’; ‘.

Exemplo de estrutura:

Void main{

A1, A2: integer; N2, N3: float;

DCLFUNC{ };

CORPO{ };

}

**Declaração de variável**

Pode-se declamar deve se usar $ antes para poder declarar uma ou mais de uma variável na mesma declaração separando por vírgula, e depois de declarar as variáveis deve declarar o seu tipo.

Para declarar uma variável usamos a sintaxe: nomeDaVariavel:TipoDeVariavel;

Lembre-se de sempre por ***;*** para encerrar o código.

t1, t2, t3 : integer;

t3: float; nome1, nome2 : string ;

a, b, c: char;

**Declaração de Funções**

Para declarar uma função devemos usar $ antes informar primeiro o seu tipo para retorno, seu nome, seus parâmetros, podendo informar mais de um tipo de função.

Após isso abrimos chaves {onde podemos declarar variáveis, funções e seu corpo (Podendo iniciar uma nova variável que pode ter uma expressão com seus respectivos parâmetros, conter comandos if/else e o fim desse corpo). Retornando o valor da nossa função e fechando as chaves} podemos declarar outra função.

Exemplo de declaração de função:

Integer f1{

N1= 10;

If(N1 == 10){

cout << “É igual a 10””;

}

}

**cout**

O objeto cout imprime os dados na tela, deve iniciar com cout contes << o dado a ser informado e impressão de variáveis. E também tem a opção de separar as variáveis para impressão por .

Exemplo:

cout <<“Lendo o primeiro numero...”.

cout << numero1, numero2;

**cin**

O objeto cin realiza a leitura de dados informadas pelo usuário, e armazena na variável desejada. Ele inicia com cin e aponta >> para a variável em que será armazenada

Exemplo:

cin>> var1;

**Estruturas de controle**

**If/else**

Os comandos if/else são as estruturas condicionais. Elas são utilizadas para verificar se uma determinada expressão é verdadeira ou não. Se existir o comando else e a expressão no if for falsa, ele será executado.

Após digitar o comando if deve abrir (informar o nome da variável para comparação, informar o tipo de comparação) e fechar parênteses e abrir chaves {dentro das chaves executar novos comandos caso for necessário ao fechar chaves} podemos chamar o comando else aberto com {inserindo novos comando para a comparação e fechando com}. Else também pode ser vazio, contendo apenas o if.

Exemplo:

If(variavel1 >= 10){

variavel2 = 5;

}else{

Variavel3 = 9;

};

**While**

O comando while se trata de um laço de repetição, vai ser utilizado quando não sabemos a quantidade de vezes que a função deverá ser repetida.

Para utilizarmos devemos chamar o comando while seguido de parênteses (onde devemos chamar a variável e iniciar a comparação fechando com parênteses) e abrindo chaves {para iniciar o nosso laço de repetição, podendo chamar novos comandos e repeti-los e fechando com}.

Exemplo:

While(numero < 10){

numero++;

};

**Do/while**

Tem quase o mesmo funcionamento que o while, a diferença é que dele teremos comando executados apenas uma única vez

Exemplo:

Do{

cout<< %%Teste 1%%

}

While(comandos)

**for**

O comando para se tratar de um laço de repetição onde nós temos a informação de quantas vezes ela deverá ser repetida.

Para utilizarmos devemos chamar o comando for seguido de parênteses (informando a variável de início; o nome da variável e a sua comparação; e o incremento podendo ser ++ ou – fechando com parênteses) e abrindo chaves { para iniciar o laço de repetição, podendo chamar novos comando e repeti-los e fechando com }.

Exemplo:

For(n1=0; n2 <= 10; i++){

cont++;

};

**Tipos de comparação**

Tipos de comparação podem ser utilizados em comandos if/else, while e for comparando seus dados.

Igual: ==

Diferente: !=

Maior: >

Menor: <

Menor ou igual: <=

Maior ou igual: >=

Exemplos:

x == y

Verdadeiro se x for igual a y.

x != y

Verdadeiro se x for diferente de y

x > y

Verdadeiro se x for maior que y.

x < y

Verdadeiro se x for menor que y

x <= y

Verdadeiro se x for menor ou igual a y

x >= y

Verdadeiro se x for maior ou igual a y

**Operadores Aritméticos**

São expressões matemáticas onde os fatores receberam números para serem utilizadas, com soma de variáveis, divisão, multiplicação e subtração.

Adição: +

Subtração: -

Divisão: /

Multiplicação:  \*

Exemplos:

soma = a + b;

mult =  a \* b;

div = a / b;

sub = a - b;

**Incremento e Decremento**

Incremento a cada vez que for chamado adiciona mais 1 para a contagem, é representado por ++

Decremento a cada vez que for chamado remove menos 1 da contagem, é representado por –

Exemplo:

Incremento:

A = 2;

A++;

cout A;

3

Decremento:

A = 2;

A--;

cout A;

1

**Comentários**

Comentários servem para fazer observações no código, instruções, ou explicar o que foi programado, os comentários não fazem nenhuma ação no código.

Para comentar uma linha basta colocar // no começo da linha.

Para comentar um bloco basta colocar /\* Para iniciar o campo de comentário e finalizar o campo com \*/

Exemplo comentário de linha:

//N1: integer; N2: integer;

Exemplo comentário de bloco:

-->//

Nome: string;

N1, N2, N3: integer;

}//<--

**Regras léxicas**

* Nome de variáveis não podem começar com números e nem caracteres especiais.
* Strings devem estar dentro de aspas simples ‘ ‘.
* Literal devem estar dentro de aspas duplas ‘’ ‘’.
* Números reais devem ser armazenados em float.
* Literal deve estar dentro de aspas simples ‘teste’
* String deve estar dentro de aspas duplas “teste2”
* Char deve está dentro de aspas triplas #s#
* Integer não tem limite de números
* Float não tem limite de números antes e após o ponto

**Erro Léxico**

* Abrir um bloco de comentário e não fechar.
* Caso literal não estiver dentro de aspas simples.

**Erro Semântico**

* Dados do tipo string podem receber tanto caracteres quanto palavras
* Variáveis ambíguas

**Erro Sintático**

* Se abrir um comando e não fechar.
* Identificadores não podem ser palavras reservadas da linguagem
* Palavras reservadas incompletas