

ALGORITMOS I

Professores:

Adilso Nunes de Souza

Maikon Cismoski dos Santos



ROTEIRO DA AULA

- Comando Para / For
 - Exemplos do comando Para
 - Exemplos do comando For
 - Exemplo de teste de mesa



COMANDO PARA

- Sintaxe:

para <variável de controle> de <valor inicial>
ate <valor final> [passo <incremento>] faça
 <lista de comandos>

Fimpara

OBS: O <valor inicial> e o <valor final> podem
ser valores estáticos ou definido por variáveis



COMANDO PARA

- A “variável de controle” é uma variável do tipo inteiro.
- É automaticamente controlada pelo laço e incrementada a cada execução, quando definido o valor do incremento na instrução “passo” ele segue este valor caso contrário incrementa 1.
- O bloco de comandos vai ser executado tantas vezes quantas forem necessárias, enquanto a “variável de controle” mantenha a condição de parada verdadeira.



COMANDO PARA

- A execução do laço “para” ocorre da seguinte forma:
 - No início da execução do comando, o valor inicial é atribuído a variável controladora;
 - Em seguida a condição de parada é avaliada:
 - Se for verdadeira o bloco de comandos inicia sua execução;
 - Caso contrário o comando se encerra;
 - Ao final do bloco de comandos, o incremento/decremento é executado;
 - Em seguida a condição de parada é avaliada.
 - Se for verdadeira o bloco de comandos reinicia sua execução;
 - caso contrário o comando se encerra;



COMANDO PARA

```
algoritmo "repeticao_1"
```

```
var
```

```
    x, num, soma : inteiro
```

```
inicio
```

```
    soma <- 0
```

```
    para x de 1 ate 10 passo 1 faca
```

```
        escreva("Digite o valor ", x, ": ")
```

```
        leia (num)
```

```
        soma <- soma + num
```

```
    fimpara
```

```
    escreval("Soma: ", soma)
```

```
fimalgoritmo
```



COMANDO PARA

```
algoritmo "repeticao_1"  
  
var  
    x, num, soma : inteiro  
inicio  
    soma <- 0  
    para x de 1 ate 10 passo 2 faca  
        escreva("Digite o valor ", x, ": ")  
        leia (num)  
        soma <- soma + num  
    fimpara  
    escreval("Soma: ", soma)  
fimalgoritmo
```



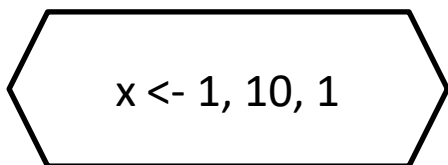
COMANDO PARA

```
algoritmo "repeticao_1"  
  
var  
    x, num, soma : inteiro  
inicio  
    soma <- 0  
    para x de 10 ate 1 passo -1 faca  
        escreva("Digite o valor ", x, ": ")  
        leia (num)  
        soma <- soma + num  
    fimpara  
    escreval("Soma: ", soma)  
fimalgoritmo
```




DIAGRAMA DE BLOCOS

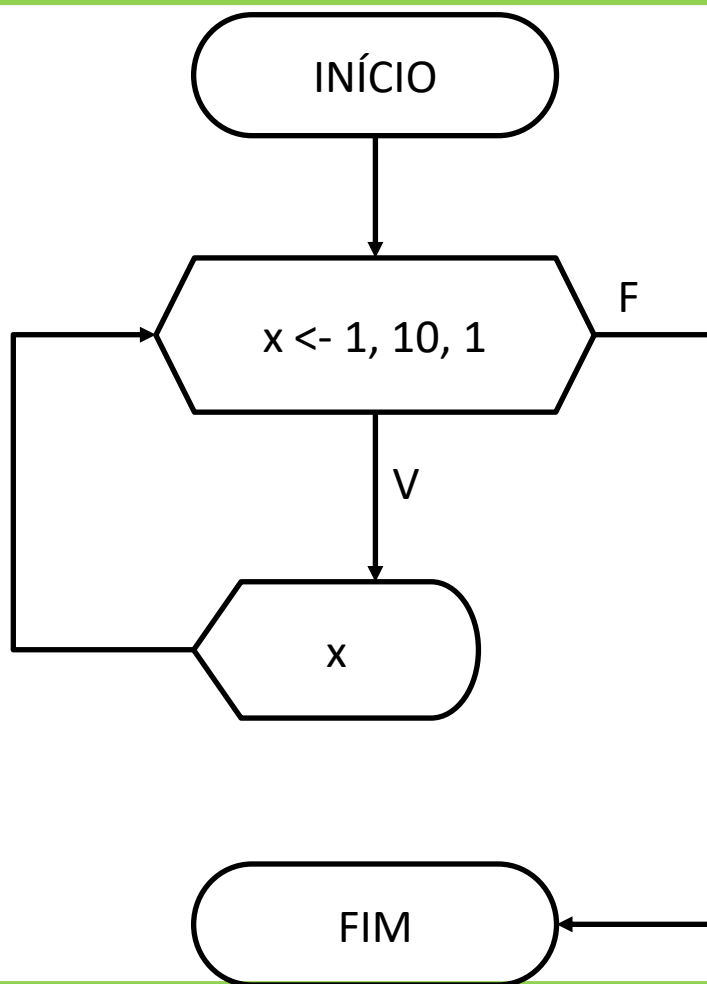
- O controle da execução de um comando de repetição é avaliado em um teste condicional, por esta razão utiliza-se o losango para expressarmos este trecho do algoritmo no diagrama de blocos.
- Porém o laço “para” possui uma representação diferenciada, pois apresenta um grupo de ações existentes em relação a sua atividade subsequencial, neste caso usamos o símbolo:



Dentro é colocado a variável de controle, seu valor inicial, valor final e o incremento.



DIAGRAMA DE BLOCOS





FOR

- Laço for (para):
 - No Algoritmo:
para <variável de controle> de <valor inicial> ate <valor final> [passo
<incremento>] faça
 <lista de comandos>
fimpara
 - Na linguagem C/C++:
for (<variavel> = <valor inicial> ; <condição parada> ; <incremento>)
{
 Bloco de Comandos ou instruções;
}



FOR

- Exemplo:

```
for (x = 1; x <= 10 ; x++)  
{  
    cout << "Digite o número: ";  
    cin >> num;  
    fflush(stdin);  
    soma = soma + num;  
}  
cout << soma << endl;
```



FOR

```
1  #include <iostream>
2  using namespace std;
3
4  main()
5  {
6      int x, ini, fim;
7      cout << "Informe inicio: ";
8      cin >> ini;
9
10     cout << "Informe fim: ";
11     cin >> fim;
12
13     for (x=ini; x<=fim; x++)
14     {
15         cout << "x: " << x << endl;
16     }
17 }
```



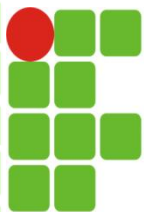
FOR

- A “variável” é uma variável do tipo inteiro, que é automaticamente controlada pelo laço e incrementada a cada execução.
- O bloco de comandos vai ser executado tantas vezes quantas forem necessárias, enquanto a “variável” mantenha a condição de parada verdadeira.
- No momento em que a condição de parada se tornar falsa, o comando de repetição se encerra.



FOR

- A execução do laço “for” ocorre da seguinte forma:
 - No início da execução do comando, o valor inicial é atribuído a variável controladora;
 - Em seguida a condição de parada é avaliada:
 - Se for verdadeira o bloco de comandos inicia sua execução;
 - Caso contrário o comando se encerra;
 - Ao final do bloco de comandos, o incremento é executado;
 - Em seguida a condição de parada é avaliada.
 - Se for verdadeira o bloco de comandos reinicia sua execução;
 - caso contrário o comando se encerra;



TESTE DE MESA

```
1  #include <iostream>
2  using namespace std;
3
4  main()
5  {
6      int i, n, soma=0;
7      for (i=0; i<3; i++)
8      {
9          cout << "Informe um numero: ";
10         cin >> n;
11         soma += n;
12     }
13     cout << "soma: " << soma << endl;
14 }
15
```

Linha	i	n	soma	i<3
6	?	?	0	?
7	0	?		TRUE
10		(2)		
11			2	
12	1			
7				TRUE
10		(5)		
11			7	
12	2			
7				TRUE
10		(1)		
11			8	
12	3			
7				FALSE
13			{8}	



INTERROMPER A EXECUÇÃO

- Em muitos casos não é necessário executar todas as iterações do laço, tendo em vista uma condição qualquer ser encontrada.
- Nestes casos pode-se interromper a execução do laço antes de que a condição de parada seja alcançada.
- O comando **break** é responsável por esta ação.



COMANDO BREAK

```
#include <iostream>
using namespace std;

main()
[ {
    int i, n, soma=0;
    for (i=1; i<=10; i++)
    {
        cout << "Informe um numero: ";
        cin >> n;
        soma += n;

        if(soma>=15)
            break;
    }
    cout << "soma: " << soma << endl;
}
```



FOR

- Exemplo:

```
for(x = 1; (x == 5 or x <= 10); x++)  
{  
    cout << x << endl;  
    x += 2;  
}
```

- Quantas vezes o laço será executado?



FOR

- Exemplo:

```
int x = 1, soma = 0, i;  
for(i = 1; (soma <= 10); x++)  
{  
    cout << "Valor x: " << x << endl;  
    soma += x;  
}
```

- Qual é a variável controladora do laço?



FOR

- Exemplo:

```
for(i=1; i<=10; i++)  
{  
    cout<<"i: " <<i << endl;  
}
```

```
for(i=1; i<=10; i+=2)  
{  
    cout<<"i: " <<i << endl;  
}
```

```
for(int i=10; i>=1; i--)  
{  
    cout<<"i: " <<i << endl;  
}
```



REFERÊNCIAS

- FORBELLONE, André Luiz Villar. Lógica de programação: a construção de algoritmos e estruturas de dados. 3 ed. São Paulo: Prentice Hall, 2005.
- VILARIN, Gilvan. Algoritmos Programação para Iniciantes. Editora Ciência Moderna. Rio de Janeiro, 2004.
- MORAES, Paulo Sérgio. Curso Básico de Lógica de Programação. Centro de Computação – Unicamp, 2000.
- STEINMETZ, Ernesto H. R.; FONTES, Roberto Duarte Cartilha Lógica de Programação. Editora IFB, Brasília - DF, 2013.