

Curso: Análise e Desenvolvimento de Sistemas

Período: 2º Semestre

Disciplina: Linguagem de Programação

Professor: Profa. Dra. Ligia Rodrigues Prete

E-mail: ligia.prete@fatec.sp.gov.br

4 – Estruturas de Repetição

Estrutura de Repetição: while

- A instrução **enquanto** (algoritmo) e **while** (Linguagem C) executa uma instrução ou bloco de instruções enquanto uma determinada condição for verdadeira. A sua sintaxe é:

Algoritmo

```
contador <- 0
enquanto (condicao) faca
    instrucao1
    instrucao2
...
    instrucaoN
    contador <- contador + 1
fimenquanto
```

Linguagem C

```
1      contador = 0;
2      while (condicao1) {
3          instrucaoA;
4          instrucaoB;
5          ...
6          instrucaoN;
7          contador++;
8      }
```

O seu funcionamento segue os passos:

1. A condição é avaliada;
2. Se o resultado da avaliação for verdadeiro, é executada a instrução ou bloco de instruções ali presente, associada ao while.
3. Volta ao ponto 2.
4. Se o resultado da avaliação for falso, o laço termina e o programa continua na instrução imediatamente após o while.

Exemplo com “while”

```
1  #include <stdio.h>
2
3  void main()
4  {
5      int contador;
6
7      contador = 1;
8      while (contador <= 10)
9      {
10         printf("%d \n", contador);
11         contador++;
12     }
13 }
14
```

```
1
2
3
4
5
6
7
8
9
10
```

Estrutura de Repetição: do .. while

- A instrução **repita (algoritmo)** e **do .. while (Linguagem C)** executa uma instrução ou bloco de instruções enquanto uma determinada condição for verdadeira. A sua sintaxe é:

Algoritmo

```
contador <- 0
repita
    instracao1
    instracao2
    ...
    instracao
    contador <- contador + 1
ate (condicao)
```

Linguagem C

```
1      contador = 0;
      do {
2          instracaoA;
3          instracaoB;
4          ...
5          instracaoN;
6          contador++;
7      } while (condicao);
```

O seu funcionamento seguem os passos:

1. A instrução (ou bloco de instruções é executada);
2. A condição é avaliada;
3. Se o resultado da condição for verdadeiro, volta ao ponto 1.
4. Se o resultado da condição for falso, termina o laço e o programa continua na instrução seguinte ao laço.

Exemplo com “do .. while”

```
1  #include <stdio.h>
2
3  void main()
4  {
5      int contador;
6
7      contador = 1;
8      do
9      {
10         printf("%d \n", contador);
11         contador++;
12     }
13     while (contador <= 10);
14 }
15
```

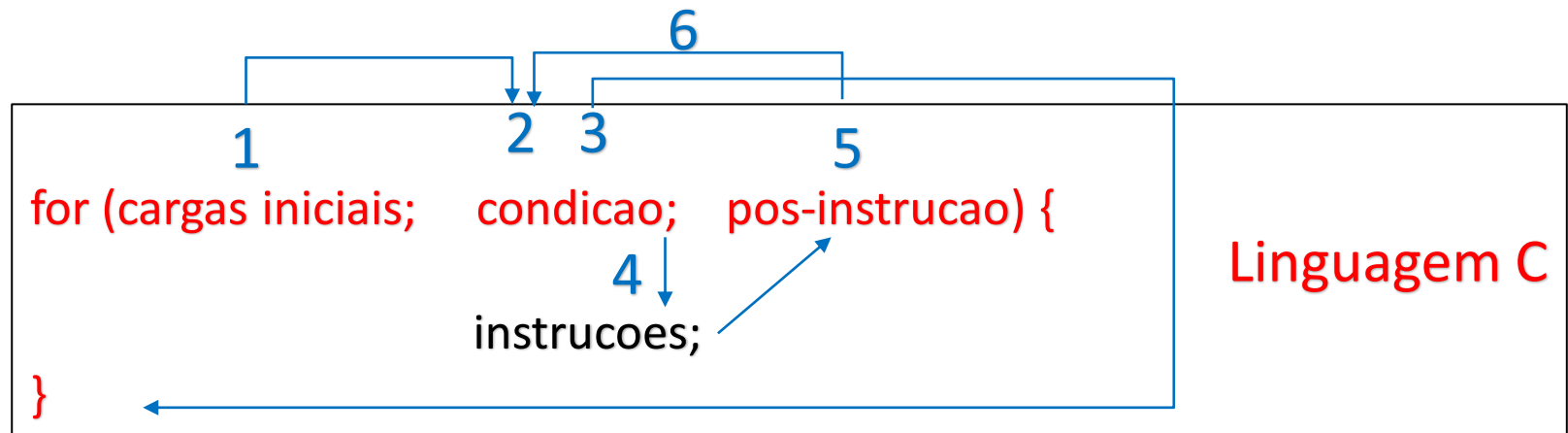
```
1
2
3
4
5
6
7
8
9
10
```

Estrutura de Repetição: for

- A instrução **para .. faça (algoritmo)** e **for (Linguagem C)** executa uma instrução ou bloco de instruções enquanto uma determinada condição for verdadeira. A sua sintaxe é:

```
para <variável> de <valor inicial> ate <valor final> faça
    instruções
fimpara
```

Algoritmo



O seu funcionamento seguem os passos:

1. A variável é inicializada com um valor em **cargas iniciais**.
2. A **condição** é avaliada;
3. Se o resultado da condição for **falso**, então o laço for termina e o programa continua na instrução imediatamente a seguir.
4. Se o resultado da condição for **verdadeiro**, então é executada a instrução (ou bloco de instruções) do laço.
5. Depois de executada a instrução presente no laço, é executada a **pós-instrução**. Nesse componente do laço for são realizadas as alterações para passar à próxima iteração do laço.
6. Volta ao ponto 2.

Exemplo com “for”

```
1  #include <stdio.h>
2
3  void main()
4  {
5      int contador;
6
7      for(contador = 1; contador <= 10; contador++)
8      {
9          printf("%d \n", contador);
10     }
11 }
12
```

```
1
2
3
4
5
6
7
8
9
10
```