

## **Conteúdo**

- Estruturas de repetição.

## **Competências**

- (3) Implementar programas para a solução de problemas computacionais usando estruturas de controle de fluxo, por meio de repetições.

## **1 Estruturas de controle – repetição/laços**

Uma estrutura de repetição permite que uma sequência de comandos seja executada repetidamente até que uma determinada condição de parada seja satisfeita. A condição de parada deve ser representada por uma expressão lógica.

As estruturas de repetição se dividem em:

- Repetição com teste no início: WHILE
- Repetição com teste no fim: DO ... WHILE
- Repetição com variável de controle: FOR

### **1.1 Repetição com teste no início**

A condição de parada é checada no início do bloco a ser repetido. Enquanto a condição é verdadeira, o bloco é executado. No momento em que a condição se torna falsa, a repetição é finalizada.

Comportamento: PRIMEIRO TESTA, DEPOIS EXECUTA.

Se na primeira tentativa de execução do bloco a condição for falsa, o bloco não é executado nenhuma vez.

O modelo genérico em Portugol é:

```
ENQUANTO (condição) FAÇA
    bloco de ação a ser repetido
FIM_ENQUANTO
```

O modelo genérico em Linguagem C é:

```
while (condição){
    bloco de ação a ser repetido
}
```

**Exemplo 1:** Elaborar um algoritmo que leia um número inteiro positivo e mostre na tela os números de 1 até o número lido.

## PORTUGOL

```
ALGORITMO mostraNumeros
VAR
    INTEIRO: maximo, contador;
INÍCIO
    ESCRIVA("Informe o número máximo:");
    LEIA(maximo);
    contador ← 1;

    ENQUANTO (contador <= maximo) FAÇA
        ESCRIVA(contador);
        contador ← contador + 1;
    FIM_ENQUANTO
FIM
```

## LINGUAGEM C

```
#include <stdio.h>

int main() {

    int maximo, contador;

    printf("Informe o número máximo: ");
    scanf("%d", &maximo);

    contador = 1;

    while (contador <= maximo){
        printf("%d ", contador);
        contador++;
    }

    return 0;
}
```

### 1.2 Repetição com teste no fim

Passa-se a primeira vez pelo bloco a ser repetido e a condição de parada é checada no final deste bloco. Enquanto a condição é verdadeira, o bloco é executado. No momento em que a condição se torna falsa, a repetição é finalizada.

Comportamento: PRIMEIRO EXECUTA, DEPOIS TESTA.

Mesmo que na primeira tentativa a condição seja falsa, o bloco é executado uma vez, visto que o teste é feito apenas no final do bloco a ser repetido. Ou seja, pelo menos uma vez o bloco é executado.

O modelo genérico em Portugol é:

```
REPITA
    bloco de ação a ser repetido
ENQUANTO (condição);
```

O modelo genérico em Linguagem C é:

```
do{
    bloco de ação a ser repetido
}while (condição);
```

**Exemplo 2:** Elaborar um algoritmo que leia um número inteiro positivo e mostre na tela os números de 1 até o número lido.

#### PORTUGOL

```
ALGORITMO mostraNumeros
VAR
    INTEIRO: maximo, contador;
INÍCIO
    ESCREVA("Informe o número máximo:");
    LEIA(maximo);
    contador ← 1;

    REPITA
        ESCREVA(contador);
        contador ← contador + 1;
    ENQUANTO (contador <= maximo);
FIM
```

#### LINGAGEM C

```
#include <stdio.h>

int main() {

    int maximo, contador;

    printf("Informe o número máximo: ");
    scanf("%d", &maximo);

    contador = 1;

    do{
        printf("%d ", contador);
        contador++;
    } while (contador <= maximo);

    return 0;
}
```

### 1.3 Repetição com variável de controle

Determina-se claramente uma variável (contador) que irá controlar o número de vezes que o bloco de comando é executado. A variável deve ser conduzida gradativamente de forma que, a partir do seu valor inicial, atinja um valor final a fim de encerrar a execução do bloco.

A variável é inicializada e, enquanto não atingir o valor final, repete-se o bloco de comandos. O valor da variável é atualizado de acordo com o valor do passo estipulado (Ex.: PASSO 1: a variável tem seu valor acrescido de 1 em 1; PASSO 2: a variável tem seu valor acrescido de 2 em 2, ...).

Comportamento: PRIMEIRO TESTA, DEPOIS EXECUTA.

O modelo genérico em Portugol é:

```
PARA (inicialização; condição; atualização) FAÇA
    bloco de ação a ser repetido
FIM_PARA
```

O modelo genérico em Linguagem C é:

```
for (inicialização; condição; atualização){
    bloco de ação a ser repetido;
}
```

**Exemplo 3:** Elaborar um algoritmo que leia um número inteiro positivo e mostre na tela os números de 1 até o número lido.

#### PORTUGOL

```
ALGORITMO mostraNumeros
VAR
    INTEIRO: maximo, contador;
INÍCIO
    ESCREVA("Informe o número máximo:");
    LEIA(maximo);

    PARA(contador ← 1; contador <= maximo; contador ← contador + 1) FAÇA
        ESCREVA(contador);
    FIM_PARA
FIM
```

#### LINGUAGEM C

```
#include <stdio.h>

int main() {

    int maximo, contador;

    printf("Informe o número máximo: ");
    scanf("%d", &maximo);

    for(contador = 1; contador <= maximo; contador++){
        printf("%d ", contador);
    }

    return 0;
}
```

## 2 Comparação das diferentes estruturas de repetição

### Semelhanças:

- As três estruturas têm o critério de parada determinado por uma expressão lógica.
- Enquanto a condição for verdadeira, o bloco de comandos é executado. Quando a condição passa a ser falsa, abandona-se este bloco de comandos.
- Com relação à variável de define o critério de parada:
  - Deve ser declarada normalmente.
  - Deve ser devidamente inicializada.
  - Deve ter seu valor alterado gradativamente de forma que atinja o critério de parada.

### Diferenças:

#### WHILE e DO

- Em geral, são utilizados quando não se sabe ao certo quantas vezes um bloco de comando será repetido

#### FOR

- Utilizado quando se sabe quantas vezes o bloco de comando será repetido.
- Escrita mais condensada.

## Exercícios

1. Faça um programa que escreva todos os números de 1 até 10.
2. Faça um programa que escreva todos os números de 10 até 1.
3. Faça um programa que escreva todos os números pares de 1 até 10.
4. Faça um programa que escreva todos os números pares existentes entre 1 e um número lido do teclado bem como a quantidade de números apresentados.
5. Faça um programa que escreva os números múltiplos de 5, no intervalo de 1 até 50.
6. Faça um programa que apresenta na tela a soma obtida dos 100 primeiros números inteiros. Ex.:  $1 + 2 + 3 + 4 + 5 + \dots + 97 + 98 + 99 + 100$ .
7. Dados 2 números inteiros, escreva um programa que mostre os números do menor número até o maior número informado.
8. Faça um programa que some os números fornecidos pelo usuário até que o número lido seja igual a zero. Ao final, mostre a soma.
9. Faça um programa que calcule e apresente a média aritmética de diversos números lidos do teclado. Serão digitados valores até que o usuário digite 0.
10. Construa um programa em que leia vários números e informe quantos números entre 10 e 20 foram digitados. Quando o valor 0 (zero) for lido, o programa deve ser encerrado.
11. Construa um programa que leia 10 números do teclado e apresente na tela o maior e o menor valor lido.
12. Elabore um programa que apresente na tela a tabuada de um número lido do teclado. Ex.: Considerando que o número lido é 2, o formato da apresentação deve ser o seguinte:  
  
 $2 \times 0 = 0$   
 $2 \times 1 = 2$   
 $2 \times 2 = 4$   
 $2 \times 3 = 6$   
 $2 \times 4 = 8$   
 $2 \times 5 = 10$   
 $2 \times 6 = 12$   
 $2 \times 7 = 14$   
 $2 \times 8 = 16$   
 $2 \times 9 = 18$   
 $2 \times 10 = 20$
13. Faça um programa que calcule o fatorial de um número positivo qualquer. Ex.:  $5! = 5 * 4 * 3 * 2 * 1 = 120$  e  $0! = 1$ .
14. Elabore um programa que receba um número inteiro positivo e escreva a sequência de números de Fibonacci. Por exemplo, se o número for 10, escreva 1, 1, 2, 3, 5, 8, 13, 21, 34, 55.
15. Francisco tem 1.50 metro e cresce 2 centímetros por ano, enquanto José tem 1.40 metros e cresce 3 centímetros por ano. Escreva um programa mostre quantos anos serão necessários para que José tenha uma altura maior que a altura de Francisco.

16. Construa um programa que leia uma sequência de números do teclado até que o usuário digite 0. Sobre os números lidos, apresente na tela:
- O menor valor.
  - O maior valor.
  - A soma de todos os valores.
  - A média dos valores.
  - A quantidade de números ímpares.
  - A quantidade de números negativos.

17. Fazer um programa que calcule e escreva o valor de S, de acordo com a seguinte fórmula:

$$s = \frac{1}{1} + \frac{3}{2} + \frac{5}{3} + \frac{7}{4} + \dots + \frac{99}{50}$$

18. Crie um programa que calcule e escreva a soma dos 50 primeiros termos da série:

$$\frac{1000}{1} - \frac{997}{2} + \frac{994}{3} - \frac{991}{4} + \dots$$

19. Elabore um programa que leia uma sequência de números do teclado até que o usuário digite um número negativo. Sobre os números lidos, apresente:

- O percentual de números pares.
- A média dos números.
- O percentual de números acima de 50.

20. Faça um algoritmo que apresente na tela o seguinte menu:

1. Digitar um número ímpar
2. Digitar um número par
3. Sair

Caso o usuário escolha a opção 1, ele deve digitar um número ímpar para voltar ao menu. O mesmo deve ser feito na opção 2, mas para um número par. Caso escolha a opção 3, o algoritmo deve apresentar a quantidade de números ímpares digitados na opção 1, a quantidade de números pares digitados na opção 2 e encerrar o algoritmo.

21. Escreva um programa que leia um número inteiro positivo n e em seguida imprima n linhas do chamado Triângulo de Floyd. Para n = 6, temos:

```
1
2  3
4  5  6
7  8  9  10
11 12 13 14 15
16 17 18 19 20 21
```