



INTRODUÇÃO À PROGRAMAÇÃO

Estrutura de Repetição

Para

Prof^o. Sérgio Roberto Costa Vieira, M.Sc.

Cursos de Computação

1^o. Período

Introdução à Programação

Roteiro

- **Estruturas de Repetição**
 - **Variável Contadora**
 - **Variável Acumuladora**
 - **Laços Contados**
 - Para .. Faça
 - **Exercícios de Fixação**
 - **Laços Condicionais**
 - Enquanto .. Faça
 - Repita .. Até que
 - **Exercícios de Fixação**

Introdução à Programação

Estrutura de Repetição

- A estrutura de repetição permite que uma sequência de comandos seja executada um certo número de vezes até que uma determinada condição seja satisfeita ou tenha um fim de repetições determinado.
- Estas estruturas possibilitam que nosso algoritmo seja muito mais enxuto e fácil de se programar.

Introdução à Programação

Estrutura de Repetição

- Uma estrutura de repetição é utilizada quando um trecho do algoritmo, ou até mesmo o algoritmo inteiro, **precisa ser repetido**.
- O número de repetições **pode ser fixo ou estar atrelado a uma condição**.
- Assim, existem estruturas de repetições (laços):
 - ✓ Condicionais (teste no início ou teste no fim)
 - ✓ Contados

Introdução à Programação

Estrutura de Repetição

- As estruturas de repetição são muitas vezes chamadas de Laços ou Loops e se dividem em:
 - ✓ **Laços Condicionais:** quando não se conhece o número de vezes que um conjunto de comandos no interior do laço será repetido.

A repetição ou não dos comandos dependerá do resultado de uma condição.

Também são conhecidas como: repetição com teste no início e repetição com teste no final do laço.

Introdução à Programação

Estrutura de Repetição

- ✓ **Laços Contados:** quando se conhece previamente quantas vezes o conjunto de comandos será executado.

Esse tipo de estrutura também é conhecida como repetição com variável de controle.

Todo algoritmo que possui um ou mais de seus passos repetidos um determinado número de vezes denomina-se **algoritmo com repetição**.

Introdução à Programação

Estrutura de Repetição

- Com a utilização de estruturas de repetição para a elaboração de algoritmos, torna-se necessário o uso de dois tipos de variáveis para a resolução de diversos tipos de problemas:
 - ✓ Variáveis contadoras e,
 - ✓ Variáveis acumuladoras

Introdução à Programação

Variável Contadora

- Uma **VARIÁVEL CONTADORA** é uma variável que recebe um valor inicial, geralmente 0 (zero) antes do início de uma estrutura de repetição, e é incrementada no interior da estrutura de um valor constante, geralmente 1 (um), conforme o exemplo abaixo:

```
...  
cont ← 0;  
<estrutura_de_repetição>  
  
...  
cont ← cont + 1;  
  
...  
<fim_da_estrutura_de_repetição>  
...
```


Introdução à Programação

Variável Acumuladora

- Uma **VARIÁVEL ACUMULADORA** é uma variável que recebe um valor inicial, geralmente 0 (zero) antes do início de uma estrutura de repetição, e é incrementada no interior da estrutura de um valor variável, geralmente a variável usada na estrutura de controle, conforme o exemplo abaixo:

```
...  
soma ← 0;  
<estrutura_de_repetição>  
...  
soma ← soma + x;  
...  
<fim_da_estrutura_de_repetição>  
...
```

Estrutura de Repetição com número definido de repetições

Laços Contados (PARA)

Introdução à Programação

Laços Contados

Os laços contados são úteis quando se conhece previamente o número exato de vezes que se deseja executar um determinado conjunto de comandos.

Então, este tipo de laço nada mais é que uma estrutura dotada de mecanismos para contar o número de vezes que o corpo do laço (ou seja, o comando composto em seu interior) é executado.

Introdução à Programação

Laços Contados

Sintaxe em Portugol:

PARA *<variável>* \leftarrow *<início>* **ATÉ** *<final>* **FAÇA**
<instruções e comandos>

FIMPARA

O comando **PARA** executa um bloco de comandos, gerenciando esta **execução em cima de uma contagem**.

Introdução à Programação

Laços Contados

Enquanto ela não for satisfeita, ou seja, **o valor final não for *ultrapassado***, os comandos do bloco serão repetidos a cada nova verificação da condição e obtenção de resultado falso.

Esta estrutura tem o seu **funcionamento controlado por uma variável denominada contador**.

Sendo assim, poderá executar um determinado conjunto de instruções um determinado número de vezes.

Introdução à Programação

Laços Contados

Existe uma condição especial em **que a contagem deve ser de forma decrescente**, onde o valor da variável é **decrementado** em uma unidade.

A sintaxe deste laço é a seguinte:

```
PARA <variável> ← <início> ATÉ <final> PASSO -1 FAÇA  
    <comando_composto>  
FIMPARA
```

Exemplo 1

Portugol

Algoritmo exemplo1

Declare c : inteiro;

Declare num : real;

para c \leftarrow 1 até 5 faça

 escreva(“Informe o número”);

 leia(num);

 escreva(“O quadrado do número é”, num**2);

fimpara

Fim.

*Criar um algoritmo que
entre com cinco números e
imprimir o quadrado de
cada número:*

Exemplo 1

Linguagem C

```
#include<stdio.h>
#include<math.h>
int main() {
```

```
    int c;
```

```
    float num;
```

```
    for (c=1; c<=5; c++) {
```

```
        printf("Informe o número");
```

```
        scanf("%f", &num);
```

```
        printf("O quadrado do número é %f", pow(num,2));
```

```
    }
```

```
    return 0; }
```

Criar um algoritmo que entre com cinco números e imprimir o quadrado de cada número:

Exemplo 2

Portugol

*Criar um algoritmo que
possa imprimir todos os
números pares no
intervalo de 1 – 20;*

Algoritmo exemplo2

Declare c : inteiro;

para c \leftarrow 2 até 20 passo 2 faça

escreva(“O número é: ”, c);

fimpara

Fim.

Exemplo 2

Linguagem C

*Criar um algoritmo que
possa imprimir todos os
números pares no
intervalo de 1 – 20;*

```
#include<stdio.h>

int main() {
    int c ;
    for ( c=2; c <= 20; c +=2 ) {

        printf("O número é: %d", c);

    }
    return 0;
}
```


Exemplo 2.1

Portugol

*Criar um algoritmo que
possa imprimir todos os
números pares no
intervalo de 1 – 20;*

Algoritmo exemplo2.1

Declare c : inteiro;

para c \leftarrow 1 até 20 faça

se (c % 2 = 0) então

escreva(“O número é: ”, c);

fimse

fimpara

Fim.

Exemplo 2.1

Linguagem C

```
#include<stdio.h>
```

```
int main() {
```

```
    int c;
```

```
    for (c=1; c<=20; c++) {
```

```
        if ( c % 2 == 0 ) {
```

```
            printf("O número é: %d", c);
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

Criar um algoritmo que possa imprimir todos os números pares no intervalo de 1 – 20;

Exemplo 3

Portugol

*Criar um algoritmo que
possa imprimir todos os
números ímpares no
intervalo de 19 – 1;*

Algoritmo exemplo3

Declare c : inteiro;

para c \leftarrow 19 até 1 passo -2 faça

escreva(“O número é: ”, c);

fimpara

Fim.

Exemplo 3

Linguagem C

Criar um algoritmo que possa imprimir todos os números ímpares no intervalo de 19 – 1;

```
#include<stdio.h>

int main() {
    int c;
    for (c=19; c>=1; c -= 2) {
        printf("O número é: %d", c);
    }
    return 0;
}
```

Exemplo 3.1

Portugol

*Criar um algoritmo que
possa imprimir todos os
números ímpares no
intervalo de 19 – 1;*

Algoritmo exemplo3.1

Declare c : inteiro;

para c \leftarrow 1 até 19 faça

se (c mod 2 \neq 0) então

escreva(“O número é: ”, c);

fimse

fimpara

Fim.

Exemplo 3.1

Linguagem C

```
#include<stdio.h>
```

```
int main() {
```

```
    int c;
```

```
    for (c=1; c<=19; c++) {
```

```
        if ( c % 2 != 0 ) {
```

```
            printf("O número é: %d", c);
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

Criar um algoritmo que possa imprimir todos os números ímpares no intervalo de 19 – 1;

Exemplo 4

Portugol

Algoritmo exemplo4

Declare c, idade : inteiro;

Declare nome : string;

Declare sexo : caracter;

sexo \leftarrow 'M';

para c \leftarrow 1 até 10 faça

 escreva ("Informe o nome");

 leia (nome);

 escreva ("Informe o sexo: M – Masculino ou F - Feminino");

 leia (sexo);

 escreva ("Informe a idade");

 leia (idade);

 se (sexo = 'M') ou (sexo = 'm') então

 esceva ("O nome é:", nome, "e a idade", idade);

 fimse

fimpara

Fim.

Criar um algoritmo que leia o nome, sexo e idade de 10 pessoas. Imprimir o nome e idade se a pessoa for do sexo masculino:

Exemplo 4

Linguagem C

```
#include<stdio.h>
```

```
int c, idade;
```

```
char nome[10], sexo;
```

```
sexo = 'M';
```

```
for ( c = 1; c <= 10; c++ ) {
```

```
    printf ("Informe o nome");
```

```
    gets (nome);
```

```
    printf ("Informe o sexo: M – Masculino ou F - Feminino");
```

```
    scanf ("%c", &sexo);
```

```
    printf ("Informe a idade");
```

```
    scanf ("%d", &idade);
```

```
    if ( sexo == 'M' || sexo == 'm' ) {
```

```
        printf ("O nome é: %s e a idade é %d", nome, idade);
```

```
    }
```

```
}
```

```
return 0;
```

```
}
```

Criar um algoritmo que leia o nome, sexo e idade de 10 pessoas. Imprimir o nome e idade se a pessoa for do sexo masculino:

Exemplo 5

Portugol

Algoritmo exemplo5

Declare a : inteiro;

Declare numero, soma : real;

soma \leftarrow 0;

para a \leftarrow 1 até 20 faça

 escreva (“Informe o numero”);

 leia (numero);

 se (numero $**$ 2 < 225) então

 soma \leftarrow soma + numero;

 fimse

 escreva (“A soma é: ”, soma);

fimpara

Fim.

Criar um algoritmo que leia 20 números e imprimir a soma dos números cujo o quadrados são menores que 225:

Exemplo 5

Linguagem C

```
#include<stdio.h>
#include<math.h>
int main() {
    int a;
    float numero, soma=0;
    for ( a = 1; a <= 20; a++) {
        printf ("Informe o numero");
        scanf ("%f", &numero);
        if (pow(numero, 2) < 225) {
            soma = soma + numero;
        }
        printf ("A soma é: %f", soma);
    }
    return 0; }
```

Criar um algoritmo que leia 20 números e imprimir a soma dos números cujo o quadrados são menores que 225:

Introdução à Programação

Exercícios de Fixação

P1 – Faça um algoritmo para escrever os números de 1 a 30 em ordem crescente:

P2 – Faça um algoritmo para escrever os números de 1 a 20 em ordem decrescente:

P3 – Escreva um algoritmo que imprima os 10 primeiros números maiores que 100;

P4 – Faça um algoritmo que apresente a tabuada de multiplicar de um número qualquer:

Introdução à Programação

Exercícios de Fixação

P5 – Faça um algoritmo para ler 10 números e escrever quantos são pares:

P6 – Faça um algoritmo para ler 20 números e escrever a média dos números que são ímpares:

P7 – Faça um algoritmo que leia 12 números e escrever quantos estão no intervalo de 10 a 20 e quantos estão fora do intervalo:

P8 – Escreva um algoritmo para ler 10 valores e imprimir a média aritmética destes números;

Introdução à Programação

Exercícios de Fixação

P9 – Faça um algoritmo que leia 20 números e imprimir a soma dos números cujo o valor do cubo dos números são menores que 300:

P10 – Faça um algoritmo para ler a nota de três provas (p1, p2, p3) de 10 alunos e escrever a média aritmética de cada aluno, se a média for maior que 6 escrever “Parabéns Aprovado!”, senão “Você foi Reprovado!”:

Introdução à Programação

Exercícios de Fixação

P11 – Faça um algoritmo que leia 20 números e imprimir quais são os números ímpares:

P12 – Faça um algoritmo para ler dois números e imprimir todos os números entre eles. Seu algoritmo deve tratar tanto se o primeiro for ou se o segundo for maior:

P13 – Faça um algoritmo para ler X e Y, calcule e imprima a soma dos valores entre eles:



INTRODUÇÃO À PROGRAMAÇÃO

Estrutura de Repetição

Para

Prof^o. Sérgio Roberto Costa Vieira, M.Sc.

Cursos de Computação

1^o. Período