

LPG0002 – Linguagem de Programação

Estruturas de Repetição (parte 1)

Prof^a Luciana Rita Guedes
Departamento de Ciência da Computação
UDESC / Joinville

Material elaborado por: Prof. Rui Jorge Tramontin Junior

Introdução

- Problemas mais complexos demandam por soluções que utilizem estruturas de repetição;

Introdução

- Problemas mais complexos demandam por soluções que utilizem estruturas de repetição;
- Uma linguagem de programação deve oferecer meios para permitir que determinadas partes de um código possam executar repetidas vezes;

Introdução

- Problemas mais complexos demandam por soluções que utilizem estruturas de repetição;
- Uma linguagem de programação deve oferecer meios para permitir que determinadas partes de um código possam executar repetidas vezes;
- Estruturas de repetição são também chamadas de **estruturas de iteração**.

Exemplos Típicos

1. Processamento de múltiplos valores de entrada;
 - Cálculo da média de N valores;

Exemplos Típicos

1. Processamento de múltiplos valores de entrada;
 - Cálculo da média de N valores;
2. Geração de múltiplos valores na saída;
 - Sequência de Fibonacci, séries em geral;

Exemplos Típicos

1. Processamento de múltiplos valores de entrada;
 - Cálculo da média de N valores;
2. Geração de múltiplos valores na saída;
 - Sequência de Fibonacci, séries em geral;
3. Processamento pode ter um número variável de repetições;
 - Fatorial, potenciação, verificar se um número é primo;

Exemplos Típicos

1. Processamento de múltiplos valores de entrada;
 - Cálculo da média de N valores;
2. Geração de múltiplos valores na saída;
 - Sequência de Fibonacci, séries em geral;
3. Processamento pode ter um número variável de repetições;
 - Fatorial, potenciação, verificar se um número é primo;
4. Rotinas que se repetem periodicamente;
 - Menu de opções, validação da entrada de dados.

ESTRUTURA DE REPETIÇÃO *WHILE*

Estrutura de Repetição *while*

- Sintaxe:

```
while( condição ){  
    // Bloco de comandos  
}
```

Estrutura de Repetição *while*

- Sintaxe:

```
while( condição ){  
    // Bloco de comandos  
}
```

Estrutura de Repetição *while*

- Sintaxe:

```
while( condição ){  
    // Bloco de comandos  
}
```



- A *condição* é testada;
 - Caso seja verdadeira, o bloco de comandos é executado;

Estrutura de Repetição *while*

- Sintaxe:

```
while( condição ) {  
    // Bloco de comandos  
}
```



- A *condição* é testada;
 - Caso seja verdadeira, o bloco de comandos é executado;
 - Caso contrário, a execução continua fora do bloco;

Estrutura de Repetição *while*

- Sintaxe:

```
while( condição ){  
    // Bloco de comandos  
}
```

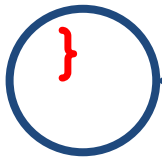


- A *condição* é testada;
 - Caso seja verdadeira, o bloco de comandos é executado;
 - Caso contrário, a execução continua fora do bloco;
- Inicialmente, parece semelhante à estrutura *if*.

Estrutura de Repetição *while*

- Sintaxe:

```
while( condição ){  
    // Bloco de comandos
```



- Porém, quando o bloco de comandos chega ao fim, o fluxo de execução retorna ao início do bloco;

Estrutura de Repetição *while*

- Sintaxe:

```
while( condição ){  
    // Bloco de comandos  
}
```



- A *condição* é testada novamente, para verificar se será feita uma nova repetição (*iteração*).

EXEMPLO 1: CÁLCULO DO FATORIAL

Exemplo 1:

Cálculo do Fatorial

- Usuário deve informar um valor N , sendo um inteiro não negativo;

Exemplo 1:

Cálculo do Fatorial

- Usuário deve informar um valor N , sendo um inteiro não negativo;
- O programa realiza uma série de multiplicações, com valores de N até 1;

Exemplo 1:

Cálculo do Fatorial

- Usuário deve informar um valor N , sendo um inteiro não negativo;
- O programa realiza uma série de multiplicações, com valores de N até 1;
- Exemplo: $N = 5$

Exemplo 1:

Cálculo do Fatorial

- Usuário deve informar um valor N, sendo um inteiro não negativo;
- O programa realiza uma série de multiplicações, com valores de N até 1;
- Exemplo: $N = 5$

$$5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$$

Exemplo 1:

Cálculo do Fatorial

- O acumulador deve ser inicializado com 1, pois temos uma sequência de multiplicações;

Exemplo 1:

Cálculo do Fatorial

- O acumulador deve ser inicializado com 1, pois temos uma sequência de multiplicações;
- A cada iteração, N é acumulado, e depois N é decrementado;

Exemplo 1:

Cálculo do Fatorial

- O acumulador deve ser inicializado com 1, pois temos uma sequência de multiplicações;
- A cada iteração, N é acumulado, e depois N é decrementado;
- O processo se encerra quando $N = 0$.

Exemplo 1:

Cálculo do Fatorial

- Analisando a memória a cada iteração:

Iteração:	N	Acumulador
Após entrada de dados	5	1

Exemplo 1:

Cálculo do Fatorial

- Analisando a memória a cada iteração:

Iteração:	N	Acumulador
Após entrada de dados	5	1
1ª iteração	5	5

Exemplo 1:

Cálculo do Fatorial

- Analisando a memória a cada iteração:

Iteração:	N	Acumulador
Após entrada de dados	5	1
1ª iteração	5	5
2ª iteração	4	20

Exemplo 1:

Cálculo do Fatorial

- Analisando a memória a cada iteração:

Iteração:	N	Acumulador
Após entrada de dados	5	1
1ª iteração	5	5
2ª iteração	4	20
3ª iteração	3	60

Exemplo 1:

Cálculo do Fatorial

- Analisando a memória a cada iteração:

Iteração:	N	Acumulador
Após entrada de dados	5	1
1ª iteração	5	5
2ª iteração	4	20
3ª iteração	3	60
4ª iteração	2	120

Exemplo 1:

Cálculo do Fatorial

- Analisando a memória a cada iteração:

Iteração:	N	Acumulador
Após entrada de dados	5	1
1ª iteração	5	5
2ª iteração	4	20
3ª iteração	3	60
4ª iteração	2	120
5ª iteração	1	120

Exemplo 1:

Cálculo do Fatorial

- Analisando a memória a cada iteração:

Iteração:	N	Acumulador
Após entrada de dados	5	1
1ª iteração	5	5
2ª iteração	4	20
3ª iteração	3	60
4ª iteração	2	120
5ª iteração	1	120
Sai do laço (fim)	0	120

Exemplo 1:

Cálculo do Fatorial

```
int n;  
printf("Digite um número: ");  
scanf("%i", &n);
```


Exemplo 1:

Cálculo do Fatorial

```
int n;  
printf("Digite um número: ");  
scanf("%i", &n);  
int fat = 1; // Acumulador
```

Exemplo 1:

Cálculo do Fatorial

```
int n;  
printf("Digite um número: ");  
scanf("%i", &n);  
int fat = 1; // Acumulador  
while ( n > 0 ){  
  
}
```

Exemplo 1:

Cálculo do Fatorial

```
int n;  
printf("Digite um número: ");  
scanf("%i", &n);  
int fat = 1; // Acumulador  
while ( n > 0 ){  
    fat = fat * n; // Acumula n em fat  
}
```

Exemplo 1:

Cálculo do Fatorial

```
int n;  
printf("Digite um número: ");  
scanf("%i", &n);  
int fat = 1; // Acumulador  
while ( n > 0 ){  
    fat = fat * n; // Acumula n em fat  
    n = n - 1;    // Decrementa n  
}
```

Exemplo 1:

Cálculo do Fatorial

```
int n;  
printf("Digite um número: ");  
scanf("%i", &n);  
int fat = 1; // Acumulador  
while ( n > 0 ){  
    fat = fat * n; // Acumula n em fat  
    n = n - 1;    // Decrementa n  
}  
printf("Fatorial: %i\n", fat);
```

Considerações

- O fatorial poderia ser calculado de outra forma;

Considerações

- O fatorial poderia ser calculado de outra forma;
- O programa realiza uma série de multiplicações, com valores de variando de 1 até N (incremental);

Considerações

- O fatorial poderia ser calculado de outra forma;
- O programa realiza uma série de multiplicações, com valores de variando de 1 até N (incremental);
- Exemplo: $N = 5$

Considerações

- O fatorial poderia ser calculado de outra forma;
- O programa realiza uma série de multiplicações, com valores de variando de 1 até N (incremental);
- Exemplo: $N = 5$

$$1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$$

Cálculo do Fatorial (v.2)

- Vamos precisar de um contador:

Iteração:	N	Cont	Acumulador
Após entrada de dados	5	1	1

Cálculo do Fatorial (v.2)

- Vamos precisar de um contador:

Iteração:	N	Cont	Acumulador
Após entrada de dados	5	1	1
1ª iteração	5	1	1

Cálculo do Fatorial (v.2)

- Vamos precisar de um contador:

Iteração:	N	Cont	Acumulador
Após entrada de dados	5	1	1
1ª iteração	5	1	1
2ª iteração	5	2	2

Cálculo do Fatorial (v.2)

- Vamos precisar de um contador:

Iteração:	N	Cont	Acumulador
Após entrada de dados	5	1	1
1ª iteração	5	1	1
2ª iteração	5	2	2
3ª iteração	5	3	6

Cálculo do Fatorial (v.2)

- Vamos precisar de um contador:

Iteração:	N	Cont	Acumulador
Após entrada de dados	5	1	1
1ª iteração	5	1	1
2ª iteração	5	2	2
3ª iteração	5	3	6
4ª iteração	5	4	24

Cálculo do Fatorial (v.2)

- Vamos precisar de um contador:

Iteração:	N	Cont	Acumulador
Após entrada de dados	5	1	1
1ª iteração	5	1	1
2ª iteração	5	2	2
3ª iteração	5	3	6
4ª iteração	5	4	24
5ª iteração	5	5	120

Cálculo do Fatorial (v.2)

- Vamos precisar de um contador:

Iteração:	N	Cont	Acumulador
Após entrada de dados	5	1	1
1ª iteração	5	1	1
2ª iteração	5	2	2
3ª iteração	5	3	6
4ª iteração	5	4	24
5ª iteração	5	5	120
Sai do laço (fim)	5	6	120

Cálculo do Fatorial (v.2)

```
int n;  
printf("Digite um número: ");  
scanf("%i", &n);
```

Cálculo do Fatorial (v.2)

```
int n;  
printf("Digite um número: ");  
scanf("%i", &n);  
int fat = 1;  // Acumulador
```

Cálculo do Fatorial (v.2)

```
int n;  
printf("Digite um número: ");  
scanf("%i", &n);  
int fat = 1; // Acumulador  
int cont = 1; // Contador
```

Cálculo do Fatorial (v.2)

```
int n;  
printf("Digite um número: ");  
scanf("%i", &n);  
int fat = 1; // Acumulador  
int cont = 1; // Contador  
while ( cont <= n ){  
  
}
```

Cálculo do Fatorial (v.2)

```
int n;  
printf("Digite um número: ");  
scanf("%i", &n);  
int fat = 1; // Acumulador  
int cont = 1; // Contador  
while ( cont <= n ){  
    fat = fat * cont; // Acumula cont em fat  
}
```

Cálculo do Fatorial (v.2)

```
int n;  
printf("Digite um número: ");  
scanf("%i", &n);  
int fat = 1; // Acumulador  
int cont = 1; // Contador  
while ( cont <= n ){  
    fat = fat * cont; // Acumula cont em fat  
    cont = cont + 1; // Incrementa cont  
}
```

Cálculo do Fatorial (v.2)

```
int n;  
printf("Digite um número: ");  
scanf("%i", &n);  
int fat = 1; // Acumulador  
int cont = 1; // Contador  
while ( cont <= n ){  
    fat = fat * cont; // Acumula cont em fat  
    cont = cont + 1; // Incrementa cont  
}  
printf("Fatorial de %i: %i\n", n, fat);
```

ESTRUTURA DE REPETIÇÃO *FOR*

Estrutura de Repetição *for*

- A estrutura de repetição *for* (PARA ... FAÇA) é adequada quando se tem uma repetição que depende de um contador;

Estrutura de Repetição *for*

- A estrutura de repetição *for* (PARA ... FAÇA) é adequada quando se tem uma repetição que depende de um contador;
- No exemplo do fatorial, tínhamos um contador de variava de 1 até N.

Estrutura de Repetição *for*

```
int cont = 1;           // Inicialização
while ( cont <= n ){    // Teste
    fat = fat * cont;
    cont = cont + 1;     // Incremento
}
```

Estrutura de Repetição *for*

```
int cont = 1;           // Inicialização
while ( cont <= n ){    // Teste
    fat = fat * cont;
    cont = cont + 1;     // Incremento
}
```

Usando o *for*:

```
int cont;
for ( cont = 1 ; cont <= n ; cont = cont + 1 ){
    fat = fat * cont;
}
```

Estrutura de Repetição *for*

```
int cont = 1; // Inicialização
while ( cont <= n ) { // Teste
    fat = fat * cont;
    cont = cont + 1; // Incremento
}
```

- A inicialização ocorre somente uma vez, antes do laço.

Usando o *for*:

```
int cont:
for ( cont = 1 cont <= n ; cont = cont + 1 ) {
    fat = fat * cont;
}
```

Estrutura de Repetição *for*

```
int cont = 1;           // Inicialização
while ( cont <= n ) {  // Teste
    fat = fat * cont;
    cont = cont + 1;    // Incremento
}
```

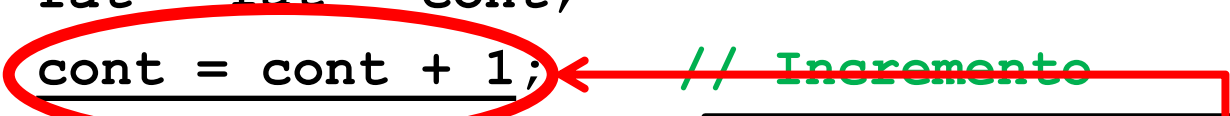
- O teste ocorre a cada iteração (*condição* para repetição).

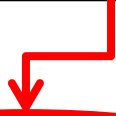
Usando o *for*:

```
int cont;
for ( cont = 1 ; cont <= n ; cont = cont + 1 ) {
    fat = fat * cont;
}
```

Estrutura de Repetição *for*

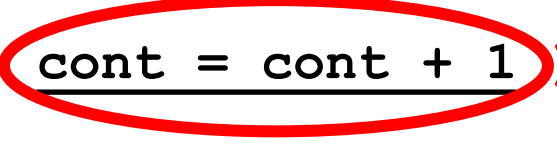
```
int cont = 1;           // Inicialização
while ( cont <= n ){    // Teste
    fat = fat * cont;
    cont = cont + 1;    // Incremento
}
```



- O incremento ocorre ao final de cada iteração.
- 

Usando o *for*:

```
int cont;
for ( cont = 1 ; cont <= n ; cont = cont + 1 ) {
    fat = fat * cont;
}
```



Estrutura de Repetição *for*

```
int cont = 1;           // Inicialização
while ( cont <= n ){    // Teste
    fat = fat * cont;
    cont = cont + 1;     // Incremento
}
```

Os códigos são **equivalentes!**

Usando o *for*:

```
int cont;
for ( cont = 1 ; cont <= n ; cont = cont + 1 ){
    fat = fat * cont;
}
```


Sintaxe da estrutura *for*

```
for( inicialização ; teste ; incremento ) {  
    // Bloco de comandos  
}
```

Sintaxe da estrutura *for*

```
for( inicialização ; teste ; incremento ) {  
    // Bloco de comandos  
}
```

- Incremento pode ser um decremento, dependendo do caso:

Sintaxe da estrutura *for*

```
for( inicialização ; teste ; incremento ) {  
    // Bloco de comandos  
}
```

- Incremento pode ser um decremento, dependendo do caso:
 - Exemplo: i variando de n até 1.

Sintaxe da estrutura *for*

```
for( inicialização ; teste ; incremento ){  
    // Bloco de comandos  
}
```

- Incremento pode ser um decremento, dependendo do caso:
 - Exemplo: i variando de n até 1.

```
int i;  
for ( i = n ; i > 0 ; i = i - 1 ){  
    // Bloco de comandos  
}
```

Fatorial usando o *for* (v.3)

```
int n;  
printf("Digite um número: ");  
scanf("%i", &n);  
int fat = 1;    // Acumulador  
int cont;       // Contador  
for ( cont = 1; cont <= n ; cont = cont + 1 ){  
    fat = fat * cont; // Acumula cont em fat  
}  
printf("Fatorial de %i: %i\n", n, fat);
```

Fatorial usando o *for* (v.3)

```
int n;  
printf("Digite um número: ");  
scanf("%i", &n);  
int fat = 1;    // Acumulador  
int cont;       // Contador  
for ( cont = 1; cont <= n ; cont = cont + 1 ) {  
    fat = fat * cont; // Acumula cont em fat  
}  
printf("Fatorial de %i: %i\n", n, fat);
```

Fatorial usando o *for* (v.3)

```
int n;  
printf("Digite um número: ");  
scanf("%i", &n);  
int fat = 1;    // Acumulador  
int cont;       // Contador  
for ( cont = 1; cont <= n ; cont++) {  
    fat = fat * cont; // Acumula cont em fat  
}  
printf("Fatorial de %i: %i\n", n, fat);
```

Fatorial usando o *for* (v.3)

```
int n;  
printf("Digite um número: ");  
scanf("%i", &n);  
int fat = 1;    // Acumulador  
int cont;       // Contador  
for ( cont = 1; cont <= n ; cont++ ){  
    fat = fat * cont; // Acumula cont em fat  
}  
printf("Fatorial de %i: %i\n", n, fat);
```


EXEMPLO 2: VALIDAÇÃO DA ENTRADA DE DADOS

Exemplo 2:

Validação da entrada de dados

- Considere que o usuário tenha que digitar uma nota de 0 a 10;

Exemplo 2:

Validação da entrada de dados

- Considere que o usuário tenha que digitar uma nota de 0 a 10;
- O programa deve garantir que a entrada esteja nesse intervalo;

Exemplo 2:

Validação da entrada de dados

- Considere que o usuário tenha que digitar uma nota de 0 a 10;
- O programa deve garantir que a entrada esteja nesse intervalo;
- O programa deve solicitar repetidas vezes a entrada da dados, até que o usuário tenha digitado corretamente.

Exemplo 2:

Validação da entrada de dados

- Exemplo em execução (1/9):

`Digite a nota (0-10) :`

Exemplo 2:

Validação da entrada de dados

- Exemplo em execução (2/9):

`Digite a nota (0-10) : 11`

Exemplo 2:

Validação da entrada de dados

- Exemplo em execução (3/9):

Digite a nota (0-10): 11

Entrada Inválida! Digite a nota (0-10):

Exemplo 2:

Validação da entrada de dados

- Exemplo em execução (4/9):

Digite a nota (0-10): **11**

Entrada Inválida! Digite a nota (0-10): **12**

Exemplo 2:

Validação da entrada de dados

- Exemplo em execução (5/9):

Digite a nota (0-10): 11

Entrada Inválida! Digite a nota (0-10): 12

Entrada Inválida! Digite a nota (0-10):

Exemplo 2:

Validação da entrada de dados

- Exemplo em execução (6/9):

Digite a nota (0-10): **11**

Entrada Inválida! Digite a nota (0-10): **12**

Entrada Inválida! Digite a nota (0-10): **-1**

Exemplo 2:

Validação da entrada de dados

- Exemplo em execução (7/9):

Digite a nota (0-10): 11

Entrada Inválida! Digite a nota (0-10): 12

Entrada Inválida! Digite a nota (0-10): -1

Entrada Inválida! Digite a nota (0-10):

Exemplo 2:

Validação da entrada de dados

- Exemplo em execução (8/9):

Digite a nota (0-10): 11

Entrada Inválida! Digite a nota (0-10): 12

Entrada Inválida! Digite a nota (0-10): -1

Entrada Inválida! Digite a nota (0-10): 8.5

Exemplo 2:

Validação da entrada de dados

- Exemplo em execução (9/9):

Digite a nota (0-10): **11**

Entrada Inválida! Digite a nota (0-10): **12**

Entrada Inválida! Digite a nota (0-10): **-1**

Entrada Inválida! Digite a nota (0-10): **8.5**

...

(programa continua a execução)

Exemplo 2:

Validação da entrada de dados

```
printf("Digite a nota (0-10): ");  
scanf("%f", &nota);
```

Exemplo 2:

Validação da entrada de dados

```
printf("Digite a nota (0-10): ");  
scanf("%f", &nota);  
while ( nota < 0 || nota > 10 ){  
  
}
```

Exemplo 2:

Validação da entrada de dados

```
printf("Digite a nota (0-10): ");  
scanf("%f", &nota);  
while ( nota < 0 || nota > 10 ){  
    printf("E. Inválida! Digite a nota (0-10): ");  
    scanf("%f", &nota);  
}
```


Exemplo 2:

Validação da entrada de dados

```
printf("Digite a nota (0-10): ");  
scanf("%f", &nota);  
while ( nota < 0 || nota > 10 ){  
    printf("E. Inválida! Digite a nota (0-10): ");  
    scanf("%f", &nota);  
}  
// Programa continua a execução
```

EXERCÍCIOS

Exercícios

1. Faça um algoritmo que leia dois valores, N e X. Mostre na tela os números de 1 a N e, a cada múltiplo de X, emita uma mensagem: “Múltiplo de X”.
2. Faça um algoritmo para calcular a **somatória**, a **soma dos quadrados** e a **média** dos N primeiros números naturais.
 - Exemplo: N=3
 - Soma: $1 + 2 + 3 = 6$
 - Soma dos quadrados: $1^2 + 2^2 + 3^2 = 14$
 - Média: $\text{Soma} / N = 2$