

Estruturas de Controle - estrutura de repetição – comandos ENQUANTO e REPITA

**Algoritmos I
2014/1**

Profa: Daniela Scherer dos Santos
daniela.santos37@ulbra.edu.br



Estruturas de Controle

- ♦ A estrutura de controle de um algoritmo indica o fluxo de execução que deverá ser seguido pelo algoritmo para que ele processe a saída desejada;
- ♦ Estruturas básicas de controle:
 - ♦ sequencial;
 - ♦ de seleção (condicional);
 - ♦ de repetição.

Estruturas de Controle

- ♦ A estrutura de controle de um algoritmo indica o fluxo de execução que deverá ser seguido pelo algoritmo para que ele processe a saída desejada;
- ♦ Estruturas básicas de controle:
 - ♦ ~~sequencial;~~
 - ♦ ~~de seleção (condicional);~~
 - ♦ de repetição.

Estrutura de Repetição

- uma estrutura de repetição permite a repetição de uma instrução, ou bloco de instruções, por um número determinado (ou indeterminado) de vezes.

Estrutura de Repetição

- uma estrutura de repetição permite a repetição de uma instrução, ou bloco de instruções, por um número **determinado** (ou indeterminado) de vezes.

quando se conhece a quantidade de repetições necessárias

Estrutura de Repetição

- uma estrutura de repetição permite a repetição de uma instrução, ou bloco de instruções, por um número determinado (ou **indeterminado**) de vezes.

quando NÃO se conhece a quantidade de repetições necessárias

Estrutura de Repetição

- ♦ Repetição determinada:
 - ♦ Comando para;
- ♦ Repetição indeterminada:
 - ♦ Comando enquanto;
 - ♦ Comando repita;

Estrutura de Repetição

- ◆ Repetição determinada:
 - ◆ Comando para;
- ◆ Repetição indeterminada (também são usados para repetição determinada):
 - ◆ Comando enquanto;
 - ◆ Comando repita;

PROBLEMA

Precisamos escrever um algoritmo para validar uma senha digitada pelo usuário.

Se a senha digitada estiver correta (123) o algoritmo deverá imprimir a mensagem "OK" e encerrar sua execução, caso contrário, deverá imprimir a mensagem "SENHA INVÁLIDA! TENDE NOVAMENTE." e possibilitar ao usuário uma nova tentativa.

Como você escreveria este algoritmo?

PROBLEMA

```
algoritmo senha
var
    inteiro senha, senhaValida;
inicio
    senhaValida = 123;
    escreva("Informe sua senha: ");
    leia(senha);
    se (senha==senhaValida) então
        inicio
            escreva("OK!")
        fim
    senao
        inicio
            escreva("SENHA INVÁLIDA! TENDE NOVAMENTE!");
        fim
    fim
```

PROBLEMA

```
algoritmo senha
var
    inteiro senha, senhaValida;
inicio
    senhaValida = 123;
    escreva("Informe sua senha: ");
    leia(senha);
    se (senha==senhaValida) então
        inicio
            escreva("OK!")
        fim
    senao
        inicio
            escreva("SENHA INVÁLIDA! TENDE NOVAMENTE!");
        fim
    fim
```

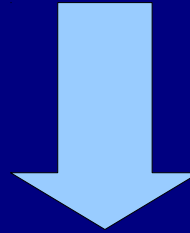
Se o usuário digitar a senha incorreta, o algoritmo deste exemplo não permitirá que ele tente digitá-la novamente.

PROBLEMA

Como fazer para que o algoritmo repita a solicitação de leitura de senha enquanto o usuário não digitar a senha correta?

PROBLEMA

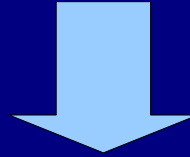
Como fazer para que o algoritmo repita a solicitação de leitura de senha enquanto o usuário não digitar a senha correta?



Podemos utilizar uma estrutura de REPETIÇÃO
→ ENQUANTO

Comando *Enquanto*

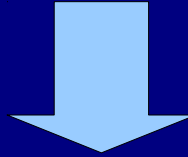
comando ENQUANTO



Repete um trecho de algoritmo enquanto uma determinada condição for verdadeira.

Comando *Enquanto*

comando ENQUANTO



Repete um trecho de algoritmo enquanto uma determinada condição for verdadeira.

Utilizado quando o número de repetições for desconhecido e também para a situação em que o número de repetições é previamente conhecido

Comando *Enquanto*

- ◆ SINTAXE:

```
enquanto (condição) faça  
início  
    <bloco de comandos>;  
fim
```


Comando *Enquanto*

```
enquanto (condição) faça  
início  
    <bloco de comandos>;  
fim
```

- *enquanto* e *faça* → são palavras reservadas do comando;
- *condição* é uma expressão lógica.

Comando *Enquanto*

1) Faça um algoritmo que calcule as médias de 20 alunos de uma turma, considerando que cada aluno tem duas notas e que estará aprovado aquele aluno cuja média aritmética das notas for igual ou superior a 7. Utilizar como dados de entrada o nome do aluno e suas notas. O algoritmo deverá fornecer como resultado o nome do aluno, sua média e a informação “Aprovado” ou “Reprovado”.

Comando *Enquanto*

algoritmo resultadoTurma

var

caracter nome;

real nota1, nota2, media;

inteiro x;

inicio

x=0;

enquanto (x<20) faça

inicio

leia (nome);

leia (nota1, nota2);

media = (nota1 + nota2)/2;

se (media >= 7) então

inicio

escreva(nome, "Aprovado com média:", media);

fim

senão

inicio

escreva(nome, "Reprovado e sua média é:", media);

fim

x = x + 1;

fim

fim

Comando *Enquanto*

algoritmo resultadoTurma

var

caracter nome;

real nota1, nota2, media;

inteiro x;

inicio

x=0;

enquanto (x<20) faça

inicio

leia (nome);

leia (nota1, nota2);

media = (nota1 + nota2)/2;

se (media >= 7) então

inicio

escreva(nome, "Aprovado com média:", media);

fim

senão

inicio

escreva(nome, "Reprovado e sua média é:", media);

fim

x = x + 1;

fim

fim

Bloco de comandos que será repetido
20 vezes

Comando *Enquanto*

```
algoritmo resultadoTurma
```

```
var
```

```
    caracter nome;
```

```
    real nota1, nota2, media;
```

```
    inteiro x;
```

```
inicio
```

```
    x=0;
```

```
    enquanto (x<20) faça
```

```
        inicio
```

```
            leia (nome);
```

```
            leia (nota1, nota2);
```

```
            media = (nota1 + nota2)/2;
```

```
            se (media >= 7) então
```

```
                inicio
```

```
                    escreva(nome, "Aprovado com média:", media);
```

```
                fim
```

```
            senão
```

```
                inicio
```

```
                    escreva(nome, "Reprovado e sua média é:", media);
```

```
                fim
```

```
            x = x + 1;
```

```
        fim
```

```
    fim
```

x é uma variável contadora → conta o número de repetições que deverão ser executadas. x inicia com valor zero e é incrementada em uma unidade após cada repetição .

Comando *Enquanto*

algoritmo resultadoTurma

var

caracter nome;

real nota1, nota2, media;

inteiro x;

inicio

x=0;

enquanto (x<20) faça

inicio

leia (nome);

leia (nota1, nota2);

media = (nota1 + nota2)/2;

se (media >= 7) então

inicio

escreva(nome, "Aprovado com média:", media);

fim

senão

inicio

escreva(nome, "Reprovado e sua média é:", media);

fim

x = x + 1;

fim

fim

O que precisaríamos alterar no algoritmo se o valor de x fosse inicializado com 1?

Comando *Enquanto*

```
algoritmo resultadoTurma
```

```
var
```

```
    caracter nome;
```

```
    real nota1, nota2, media;
```

```
    inteiro x;
```

```
inicio
```

```
    x=1;
```

```
    enquanto (x<=20) faça
```

```
        inicio
```

```
            leia (nome);
```

```
            leia (nota1, nota2);
```

```
            media = (nota1 + nota2)/2;
```

```
            se (media >= 7) então
```

```
                inicio
```

```
                    escreva(nome, "Aprovado com média:", media);
```

```
                fim
```

```
            senão
```

```
                inicio
```

```
                    escreva(nome, "Reprovado e sua média é:", media);
```

```
                fim
```

```
            x = x + 1;
```

```
        fim
```

```
    fim
```

Precisaríamos alterar a condição para garantir que o algoritmo efetue a leitura dos dados de 20 alunos

Comando *Enquanto*

2. Faça um algoritmo que deverá calcular e informar a média de “n” números inteiros lidos, encerrando a leitura destes números com a entrada de um valor negativo.

Comando *Enquanto*

2. Faça um algoritmo que deverá calcular e informar a média de “n” números inteiros lidos, encerrando a leitura destes números com a entrada de um valor negativo.

Para calcular a média precisamos:

- contar quantos números foram lidos e
- somar todos os números lidos

$$\text{media} = \text{soma} / \text{quantidadeNumerosLidos}$$

Comando *Enquanto*

algoritmo mediaDeValores

var

inteiro contador, valor;

real soma, media;

inicio

contador = 0;

soma = 0;

leia (valor);

enquanto (valor >= 0) faça

inicio

soma = soma + valor;

contador = contador + 1;

leia (valor);

fim

se (contador > 0) então

inicio

media = soma / contador;

escreva ("A média entre os valores lidos é:", media);

fim

fim

Comando *Enquanto*

```
algoritmo mediaDeValores  
var
```

```
    inteiro contador, valor;  
    real soma, media;
```

```
início
```

```
    contador = 0;
```

```
    soma = 0;
```

```
    leia (valor);
```

```
    enquanto (valor >= 0) faça
```

```
        início
```

```
            soma = soma + valor;
```

```
            contador = contador + 1;
```

```
            leia (valor);
```

```
        fim
```

```
    se (contador > 0) então
```

```
        início
```

```
            media = soma / contador;
```

```
            escreva ("A média entre os valores lidos é:", media);
```

```
        fim
```

```
    fim
```

contador: variável para contar a quantidade de valores lidos
soma: variável para acumular o somatório dos valores lidos
Devem ser inicializadas com zero!

Comando *Enquanto*

```
algoritmo mediaDeValores  
var
```

```
    inteiro contador, valor;  
    real soma, media;
```

```
inicio
```

```
    contador = 0;  
    soma = 0;  
    leia (valor);
```

```
    enquanto (valor >= 0) faça  
        inicio
```

```
            soma = soma + valor;  
            contador = contador + 1;  
            leia (valor);
```

```
        fim
```

```
    se (contador > 0) então
```

```
        inicio
```

```
            media = soma / contador;  
            escreva ("A média entre os valores lidos é:", media);
```

```
        fim
```

```
    fim
```

Trecho que será repetido por um número indeterminado de vezes → enquanto o valor informado pelo usuário for positivo

Comando *Enquanto*

```
algoritmo mediaDeValores  
var
```

```
    inteiro contador, valor;  
    real soma, media;
```

```
inicio
```

```
    contador = 0;
```

```
    soma = 0;
```

```
    leia (valor);
```

```
    enquanto (valor >= 0) faça  
        inicio
```

```
            soma = soma + valor;
```

```
            contador = contador + 1;
```

```
            leia (valor);
```

```
        fim
```

```
    se (contador > 0) então
```

```
        inicio
```

```
            media = soma / contador;
```

```
            escreva ("A média entre os valores lidos é:", media);
```

```
        fim
```

```
    fim
```

Para que seja possível a realização do primeiro teste do comando enquanto, é necessária a leitura prévia de *valor* e a variável *valor* deverá ser lida novamente antes do final do bloco de repetição, para permitir novo teste no comando enquanto.

Comando *Enquanto*

```
algoritmo mediaDeValores  
var
```

```
    inteiro contador, valor;  
    real soma, media;
```

```
inicio
```

```
    contador = 0;
```

```
    soma = 0;
```

```
    leia (valor);
```

```
    enquanto (valor >= 0) faça
```

```
        inicio
```

```
            soma = soma + valor;
```

```
            contador = contador + 1;
```

```
            leia (valor);
```

```
        fim
```

```
        se (contador > 0) então
```

```
            inicio
```

```
                media = soma / contador;
```

```
                escreva ("A média entre os valores lidos é:", media);
```

```
            fim
```

```
fim
```

Evita que o algoritmo efetue uma divisão por zero

Comando *Enquanto*

Exercício: Precisamos escrever um algoritmo para validar uma senha digitada pelo usuário.

Se a senha digitada estiver correta (123) o algoritmo deverá imprimir a mensagem "OK" e encerrar sua execução, caso contrário, deverá imprimir a mensagem "SENHA INVÁLIDA! TENTE NOVAMENTE." e possibilitar ao usuário uma nova tentativa.

Como você escreveria este algoritmo?

Comando *Repita*

- serve para determinarmos que um ou vários comandos sejam repetidamente executados até que uma determinada condição seja verdadeira.
- SINTAXE:

```
repita  
    <bloco de comandos>;  
até (condição);
```


Comando *Repita*

repita

<bloco de comandos>;
até (condição);

- *repita* e *até* são palavras reservadas do comando;
- condição é uma expressão lógica.

Comando *Repita*

repita

**<bloco de comandos>;
até (condição);**

- *repita* e *até* são palavras reservadas do comando;
- condição é uma expressão lógica.

O teste da condição de repetição ocorre após a execução de todos os comandos!

Comando *Repita*

1) Faça um algoritmo que calcule as médias de 20 alunos de uma turma, considerando que cada aluno tem duas notas e que estará aprovado aquele aluno cuja média aritmética das notas for igual ou superior a 7. Utilizar como dados de entrada o nome do aluno e suas notas. O algoritmo deverá fornecer como resultado o nome do aluno, sua média e a informação “Aprovado” ou “Reprovado”.

Comando *Repita*

algoritmo resultadoTurma

var

caracter nome;

real nota1, nota2, media;

inteiro x;

inicio

x=0;

repita

leia (nome);

leia (nota1, nota2);

media = (nota1 + nota2)/2;

se (media >= 7) então

inicio

escreva(nome, "Aprovado com média:", media);

fim

senão

inicio

escreva(nome, "Reprovado e sua média é:", media);

fim

x = x + 1;

até (x==20);

fim

Comando *Repita*

algoritmo resultadoTurma

var

caracter nome;

real nota1, nota2, media;

inteiro x;

inicio

x=0;

repita

leia (nome);

leia (nota1, nota2);

media = (nota1 + nota2)/2;

se (media >= 7) então

inicio

escreva(nome, "Aprovado com média:", media);

fim

senão

inicio

escreva(nome, "Reprovado e sua média é:", media);

fim

x = x + 1;

até (x==20);

fim

x é uma variável contadora → conta o número de repetições que deverão ser executadas. x inicia com valor zero e é incrementada em uma unidade após cada repetição .

Comando *Repita*

2. Faça um algoritmo que deverá calcular e informar a média de “n” números inteiros lidos, encerrando a leitura destes números com a entrada de um valor negativo.

Para calcular a média precisamos:

- contar quantos números foram lidos e
- somar todos os números lidos

$$\text{media} = \text{soma} / \text{quantidadeNumerosLidos}$$

Comando *Repita*

```
algoritmo mediaDeValores  
var
```

```
    inteiro contador, valor;  
    real soma, media;
```

```
inicio
```

```
    contador = 0;
```

```
    soma = 0;
```

```
    repita
```

```
        leia(valor);
```

```
        soma = soma + valor;
```

```
        contador = contador + 1;
```

```
    até (valor < 0);
```

```
    se (contador > 0) então
```

```
        inicio
```

```
            media = soma / contador;
```

```
            escreva ("A média entre os valores lidos é:", media);
```

```
        fim
```

```
    fim
```

EFETUE O TESTE DE
MESA CONSIDERANDO -1
COMO CONTEÚDO DA
VARIÁVEL valor!

Comando *Repita*

```
algoritmo mediaDeValores  
var
```

```
    inteiro contador, valor;  
    real soma, media;
```

```
início
```

```
    contador = 0;
```

```
    soma = 0;
```

```
    repita
```

```
        leia(valor);
```

```
        se(valor >= 0) então
```

```
            início
```

```
                soma = soma + valor;
```

```
                contador = contador + 1;
```

```
            fim
```

```
    até (valor < 0);
```

```
    se (contador > 0) então
```

```
        início
```

```
            media = soma / contador;
```

```
            escreva ("A média entre os valores lidos é:", media);
```

```
        fim
```

```
    fim
```

EFETUE O TESTE DE MESA
CONSIDERANDO -1 COMO
CONTEÚDO DA VARIÁVEL *valor*!

- é necessário incluir um teste para impedir de acrescentar ao acumulador *soma* um valor negativo e para que o mesmo não fosse contado como valor válido.

Comando *Repita*

```
algoritmo mediaDeValores  
var
```

```
    inteiro contador, valor;  
    real soma, media;
```

```
início
```

```
    contador = 0;
```

```
    soma = 0;
```

```
    repita
```

```
        leia(valor);
```

```
        se(valor >= 0) então
```

```
            início
```

```
                soma = soma + valor;
```

```
                contador = contador + 1;
```

```
            fim
```

```
    até (valor < 0);
```

```
    se (contador > 0) então
```

```
        início
```

```
            media = soma / contador;
```

```
            escreva ("A média entre os valores lidos é: ", media);
```

```
        fim
```

```
    fim
```

EFETUE O TESTE DE MESA
CONSIDERANDO -1 COMO
CONTEÚDO DA VARIÁVEL *valor*!

- é necessário incluir um teste para impedir de acrescentar ao acumulador *soma* um valor negativo e para que o mesmo não fosse contado como valor válido.

PORTANTO, DEVIDO AO TESTE NO
FINAL, NEM SEMPRE O COMANDO
repita É UMA BOA ESCOLHA

Referências Utilizadas

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F.. Lógica de Programação: a construção de algoritmos e estruturas de dados. São Paulo: Makron Books, 2000.