



EJA IV – Ensino Fundamental
Qualificação Profissional – Informática Básica
Análise e Lógica da Programação

Elementos Básicos: Comandos e funções

Rildo Oliveira



26/03/2024

ROTEIRO DE AULA

OBJETO DO CONHECIMENTO: Elementos Básicos: Comandos e funções

HABILIDADE: (EMIFFTP02) Levantar e testar hipóteses para resolver problemas do cotidiano pessoal, da escola e do trabalho, utilizando procedimentos e linguagens adequados à investigação científica.

OBJETIVOS:

- Apresentar a estrutura de um algoritmo como são escritas as expressões em um código;
- Qual a estrutura de escrita de um algoritmo;

DA TEORIA À PRÁTICA: Uso de imagens, texto e conceitos para um melhor entendimento do tema abordado.

Pasta Compartilhada EJA



<https://github.com/rildexter/eja2024/tree/main>

Para casa - Solução

1. Elabore um algoritmo que peça um valor em graus Fahrenheit de uma temperatura e transforme em graus Celsius ($F = 9/5 * C + 32$)
2. Elabore um algoritmo que recebe como entrada o raio (não negativo) de um círculo. Calcule a área e o perímetro deste círculo. Imprima o raio, a área e o perímetro. (área = $\pi * r^2$, perímetro = $2 * \pi * r$)



Para casa - Solução

```
9
10 Var
11   celsius, fahrenheit : real
12
13
14 Inicio
15 //Seção de comandos
16   escreva ("Fahrenheit? ")
17   leia(fahrenheit)
18   celsius <- 5 * (fahrenheit - 32) / 9
19   escreva ("celsius = ", celsius)
20 Fimalgoritmo
```

Para casa - Solução

```
3
4
5 Var
6   area, raio, perimetro: Real
7
8
9 Inicio
10   escreva("Informe o raio de uma circunferência.")
11   leia(raio)
12
13   escreva ("Imprima o perimetro da circunferência.")
14   leia(perimetro)
15
16   escreva ("Informe a área da circunferência.")
17   leia(area)
18
19   area := 3.14 * (raio * raio) * (perimetro * perimetro) * (area * area)
20
21   escreva("A Área da circunferência é: ",area)
22
23
24
25 Fimalgoritmo
```

Atividades 1

Mostre a mensagem “Olá Mundo!” na tela

Atividades 1

Mostre a mensagem “Olá Mundo!” na tela

```
1 algoritmo "quest01"  
2  
3 var  
4  
5 inicio  
6 escreva ("Olá Mundo!")  
7 fimalgoritmo  
8
```

 Console simulando o modo texto do MS-DOS

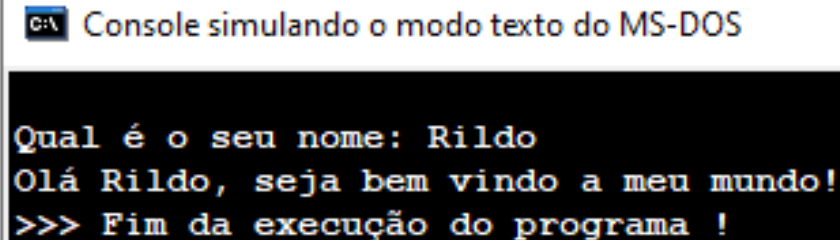
```
Olá Mundo!  
>>> Fim da execução do programa !
```


Atividades 2

Pergunte o nome do usuário e escreva a seguinte mensagem:
Olá **nome_usuario**, seja bem vindo a meu mundo!

Atividades 2

```
1 algoritmo "quest02"  
2 var  
3 // Declaração da variável  
4     nome: caractere  
5 inicio  
6 // Comando de saída. Mensagem perguntando o nome.  
7 escreva ("Qual é o seu nome: ")  
8 // Comando de entrada. Leitura e armazenamento do valor "nome".  
9 leia (nome)  
10 // Comando de saída com inserção do valor "nome" na mensagem de boas vindas!  
11 escreva ("Olá ", nome, ", seja bem vindo a meu mundo!")  
12 fimalgoritmo  
13
```



C:\ Console simulando o modo texto do MS-DOS

Qual é o seu nome: Rildo
Olá Rildo, seja bem vindo a meu mundo!
>>> Fim da execução do programa !

Atividades 3

Agora pergunte o nome do funcionário e seu salário e mostre em tela "O funcionário ", nome, " tem um salário de R\$ ", Sal:6:2, " em Maio."

Dica: após chamar a variável do tipo real se colocarmos :valor:valor então teremos quantas casas decimais antes e/ou depois da virgula.

Exemplo: foi digitado 483

variável:4:3 resultara em 483,000

Atividades 3

```
1 algoritmo "quest03"
2   var
3   // Declaração das variáveis
4     nome: caractere
5     Sal: real
6 inicio
7   // Comandos de saída perguntado os valores "nome" e "Salário"
8   // Comandos de entrada guardando os valores "nome" e "Sal"
9   escreva ("Nome do funcionário: ")
10  leia (nome)
11  escreva ("Salário do funcionário: R$ ")
12  leia (Sal)
13  escreval ("-----")
14  // Comando de saída mensagem, com os valores de entrada.
15  escreva ("O funcionário ", nome, " tem um salário de R$ ", Sal:6:2, " em Maio.")
16
17 fimalgoritmo
18
19
```

CA Console simulando o modo texto do MS-DOS

```
Nome do funcionário: Rildo
Salário do funcionário: R$ 1282
-----
O funcionário Rildo tem um salário de R$ 1282.00 em Maio.
>>> Fim da execução do programa !
```

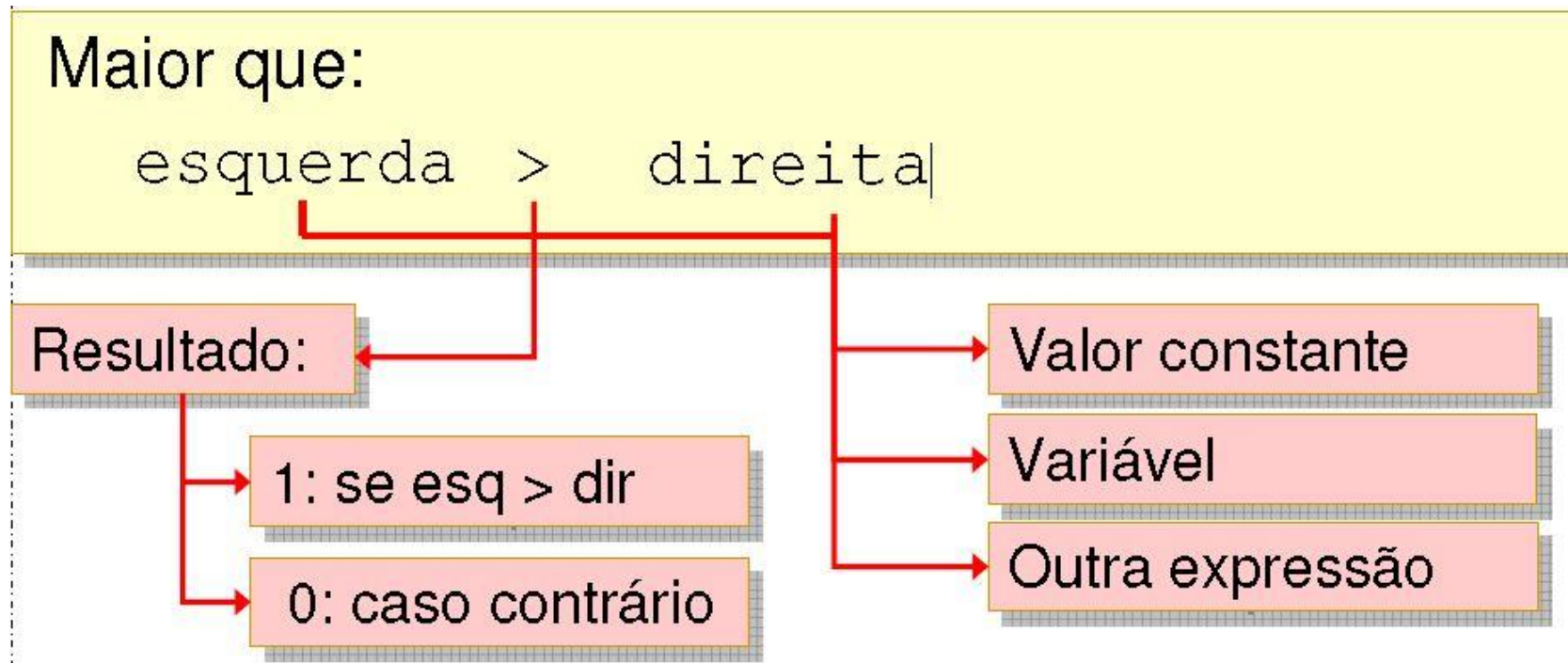
Condições

O que são condições:

- Expressões matemáticas convencionais!
- Testam validade de uma afirmação
- Resultado: número inteiro
- Interpretação:
 - 0: significa *falso*
 - 1: significa *verdadeiro*
- Operadores relacionais:
< <= > >= == !=

Condições

Operador de Comparação:



Condições

Operadores de comparação:

Expressão	Condição	Resultado
$a > b$	se $a > b$	1 (verdadeiro)
	se $a \leq b$	0 (falso)
$a \geq b$	se $a \geq b$	1 (verdadeiro)
	se $a < b$	0 (falso)
$a < b$	se $a < b$	1 (verdadeiro)
	se $a \geq b$	0 (falso)
$a \leq b$	se $a \leq b$	1 (verdadeiro)
	se $a > b$	0 (falso)
$a == b$	se $a = b$	1 (verdadeiro)
	se $a \neq b$	0 (falso)
$a != b$	se $a \neq b$	1 (verdadeiro)
	se $a = b$	0 (falso)

Atenção

Cuidado!

Não confundir igualdade (==) com atribuição (=)!

Condições

Exemplo:

```
int a, b;  
...  
a = (1 < 2);  
b = (3 <= 2);
```

a = 1 (verdade)
b = 0 (falso)

```
int c = 3;  
int d = 10;  
int e, f;  
...  
e = (c == d);  
f = (c != d);
```

e = 0 (falso)
f = 1 (verdade)

ESTRUTURAS CONDICIONAIS

A Estrutura Condicional possibilita a escolha de um grupo de ações e estruturas a serem executadas quando determinadas condições são ou não satisfeitas. A Estrutura Condicional pode ser **Simple**s ou **Composta**.

ESTRUTURAS CONDICIONAIS

Estrutura Condicional Simples

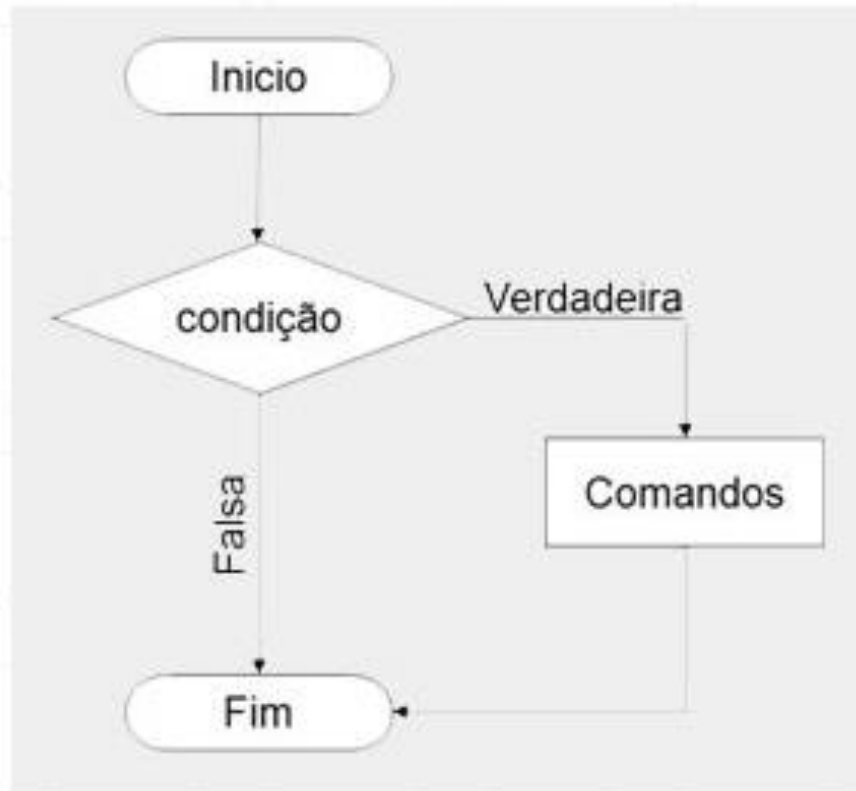
A Estrutura Condicional Simples executa um comando ou vários comandos se a condição for verdadeira. Se a condição for falsa, a estrutura é finalizada sem executar os comandos. O comando que define a estrutura é representado pela palavra SE.

Pseudocódigo:

1. INICIO ALGORITMO
2. SE condição ENTAO
3. comando
4. FIM SE
5. FIM ALGORITMO

ESTRUTURAS CONDICIONAIS

Fluxograma:



Pseudocódigo:

1. INICIO ALGORITMO
2. SE condição ENTAO
3. comando
4. FIM SE
5. FIM ALGORITMO

ESTRUTURAS CONDICIONAIS

Estrutura Condicional Composta

A Estrutura Condicional Composta segue o mesmo princípio da Estrutura Condicional Simples, com a diferença de que quando a condição não é satisfeita, será executado o outro comando. O comando que define a estrutura é representado pelas palavras SE e SE NÃO.

ESTRUTURAS CONDICIONAIS

Pseudocódigo:

1. INICIO ALGORITMO
2. SE condição ENTAO
3. comando A
4. SENAÓ
5. comando B
6. FIM SE
7. FIM ALGORITMO

Fluxograma:



ESTRUTURAS CONDICIONAIS

COMANDO SE...ENTAO...SENAO...FIMSE A estrutura mais usada é do comando SE. Como vimos nos operadores relacionais, as comparações retornam verdadeiro ou falso. É assim que escolhemos os comandos a serem executados. Toda vez que uma condição retorna VERDADEIRO, a cláusula ENTAO será executada. Se a condição retorna FALSO a cláusula SENAO será executada.

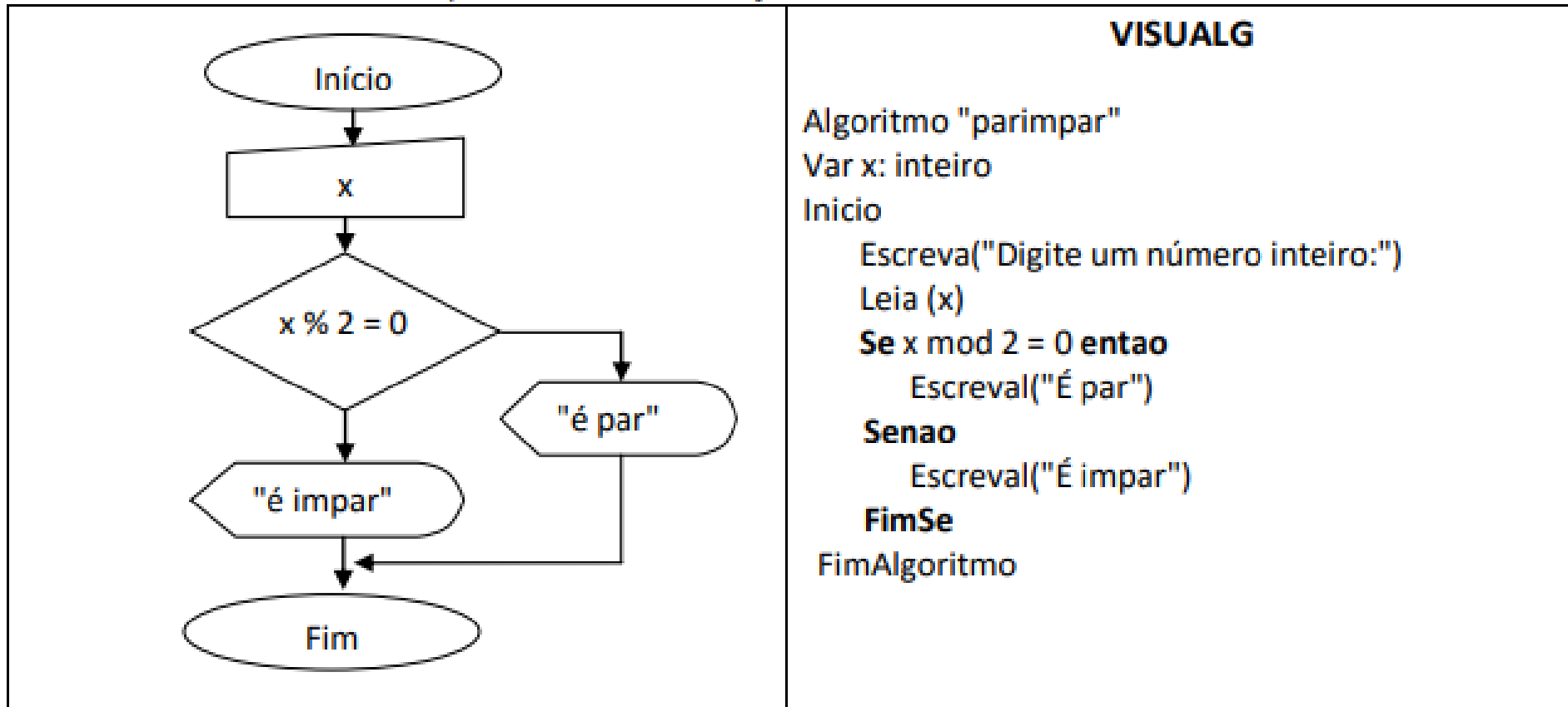
ESTRUTURAS CONDICIONAIS

É importante saber que o comando SE precisa ter um fim, pois quando a condição terminar o programa deve continuar na sua sequência. a cláusula FIMSE determina o fim do comando SE. Depois do FIMSE os comandos são independentes das condições da estrutura condicional.

VISUALG

```
Algoritmo "parimpar"  
Var x: inteiro  
Inicio  
    Escreva("Digite um número inteiro:")  
    Leia (x)  
    Se x mod 2 = 0 entao  
        Escreval("É par")  
    Senao  
        Escreval("É impar")  
    FimSe  
FimAlgoritmo
```

ESTRUTURAS CONDICIONAIS



ESTRUTURAS CONDICIONAIS

A cláusula **SENAO** não é obrigatória. Pode ser que você precise calcular apenas se a condição for verdadeira. Portanto, há várias formas de utilizar o comando **SE**.

```
LEIA (ALTURA)
SE ALTURA > 1.70 ENTAO
    ESCREVAL("A PESSOA É ALTA")
FIMSE
```

ESTRUTURAS CONDICIONAIS

Também podemos ter um comando SE dentro do outro. Se o comando SE serve para executar até duas condições (verdadeira e falsa), se eu necessito executar uma terceira, quarta condição, etc eu posso colocar um comando SE dentro do outro.

Exemplo

Faça um algoritmo que leia a nota final do aluno no Pitágoras. De acordo com a tabela abaixo, exiba a situação final do aluno:

"Aprovado" para a nota maior ou igual a 6 pontos

"Prova Final" para a nota maior ou igual a 4 e menor que 6 pontos

"Reprovado" para a nota menor que 4 pontos.

Exemplo

```
algoritmo "Pitagoras"  
var notafinal: real  
inicio  
    Escreva("Digite a nota final:")  
    Leia(notafinal)  
    Se notafinal >= 6 entao  
        Escreval("Aprovado")  
    Senao  
        Se (notafinal >= 4) e (notafinal < 6) entao  
            Escreval("Prova Final")  
        Senao  
            Escreval("Reprovado")  
        FimSe  
    FimSe  
fimalgoritmo
```

Atividade 4

Faça um Algoritmo para perguntar a velocidade de um carro. Caso ultrapasse 80Km/h, exiba uma mensagem dizendo que o usuário foi multado. Nesse caso, exiba o valor da multa, cobrando R\$5 por cada Km acima da velocidade permitida.

Atividade 4

```
1 algoritmo "quest17"  
2  
3 var  
4 //Declaração das variáveis  
5   Veloc, Multa, MultaKm: Real  
6 inicio  
7 //Comandos de entrada e saída de dados.  
8 escreval ("-----")  
9 escreval ("RADAR")  
10 escreval ("-----")  
11 escreva  
12 escreval ("Via permitida até 80km/h")  
13 escreval  
14 escreva ("Digite a velocidade do carro Km: ")  
15 leia (Veloc)  
16 escreval  
17 //Atribuição de operações as variáveis.  
18 Multa <- Veloc - 80  
19 MultaKm <- Multa * 5  
20 // Condição "SE" para verificar se a velocidade ultrapassou o permitido.  
21 // Comandos de saída com valores das operações.  
22   se (veloc > 80) entao  
23     escreva ("Você foi multado. Velocidade acima do permitido!")  
24     escreva (" Multa no valor de R$", MultaKm:5:2)  
25   Fimse  
26 finalgoritmo  
27
```

```
-----  
RADAR  
-----  
Via permitida até 80km/h  
  
Digite a velocidade do carro Km: 88  
  
Você foi multado. Velocidade acima do permitido! Multa no valor de  
R$40.00  
>>> Fim da execução do programa !
```

Atividade 5

Faça um Algoritmo que lê o ano de nascimento de uma pessoa, calcula a idade dela e depois mostra se ela pode ou não votar.

Atividade 5

```
1 algoritmo "quest18"
2
3 var
4 //Declaração de variáveis.
5     Nome: Caractere
6     AnoNasc, IdVotar: inteiro
7 inicio
8 // Comandos de entrada e saída de dados.
9 escreval ("-----")
10 escreval ("                      ELEIÇÕES 2022")
11 escreval ("-----")
12 escreval
13 escreva ("Olá, qual é o seu nome? ")
14 leia (Nome)
15 escreva ("Digite seu ano de nascimento: ")
16 leia (AnoNasc)
17 //Atribuição de variável com cálculo da idade
18 IdVotar <- 2022 - AnoNasc
19 // Condição "SE" para verificar se é maior de idade ou não. Com saída de dados
20     se (IdVotar >= 16) entao
21         escreval (Nome," você tem ", IdVotar, " anos.")
22     senao
23         escreva (Nome, "não pode votar.")
24     FimSe
25 fimalgoritmo
26
```

ELEIÇÕES 2022

Olá, qual é o seu nome? Rildo
Digite seu ano de nascimento: 1985
Rildo você tem 37 anos.

>>> Fim da execução do programa !

Atividade 6

Faça um Algoritmo que lê o nome e as duas notas de um aluno, calcule a sua média e mostre na tela. No final, analise a média e mostre se o aluno teve ou não um bom aproveitamento (se ficou acima da média 7.0).

Atividade 6

```
1 algoritmo "quest19"
2
3 var
4 //Atribuição das variáveis.
5     NomeAluno: Caractere
6     N1, N2, Media: real
7 inicio
8 //Comandos de entrada e saída de dados.
9 escreval ("-----")
10 escreval ("                Canal Educação                ")
11 escreval ("-----")
12 escreval
13 escreva ("Qual o nome do aluno: ")
14 leia (NomeAluno)
15 escreva ("Digite a primeira nota: ")
16 leia (N1)
17 escreva ("Digite a segunda nota: ")
18 leia (N2)
19 //Atribuição da variável com operação de média entre notas.
20 Media <- (N1 + N2)/2
21 escreval
22 //Condição "SE" para aproveitamento da média com comandos de saída de dados.
23     se (Media >= 7) entao
24         escreval ("A nota do Aluno ", NomeAluno, " foi ", Media:5:2)
25         escreva ("Está acima da MÉDIA com bom aproveitamento.")
26     senao
27         escreval ("A nota do Aluno ", NomeAluno, " foi", Media:5:2)
28         escreva ("Ficou abaixo da MÉDIA!")
29     fimse
30 escreval
31 finalgoritmo
```

C:\ Console simulando o modo texto do MS-DOS

Canal Educação

Qual o nome do aluno: Rildo
Digite a primeira nota: 7
Digite a segunda nota: 3

A nota do Aluno Rildo foi 5.00
Ficou abaixo da MÉDIA!

>>> Fim da execução do programa !

Função ou procedimento

Modularizar nosso programa significa dividir em partes, em módulos, onde cada módulo (função ou procedimento) resolve uma pequena parte do problema. Então, vamos entender melhor como e por que modularizar nossos programas e nas próximas aulas vamos aprender como criar nossas funções e procedimentos.

Outro nome muito comum utilizado para modularização é **sub-rotina**.

MÓDULOS

São blocos de instruções que realizam tarefas específicas.

Carregado uma vez e pode ser executado quantas vezes for necessário.

Podem ser usadas para economizar espaço e tempo de programação já que podem ser usadas várias vezes num mesmo programa.

MÓDULOS

Cada módulo, além de ter acesso às variáveis do programa que o chamou (são as variáveis globais), pode ter suas próprias variáveis (são as variáveis locais), que existem apenas durante sua chamada.

Ao se chamar um módulo, também é possível passar-lhe determinadas informações que recebem o nome de parâmetros.

MÓDULOS

Vantagens

Dividir e estruturar um algoritmo em partes logicamente coerentes;
Facilidade de testar os trechos em separado;
Evitar repetição do código-fonte;
Maior legibilidade de um algoritmo.

Função ou Procedimento

Tipos de subprogramas:

Procedimentos - são estruturas que agrupam um conjunto de comandos, que são executados quando o procedimento é chamado.

Funções - tipo especial de procedimento onde depois de executada a chamada, o valor calculado é retornado no nome da função que passa a ser uma variável da expressão.

Função Recursiva – são aquelas que chamam a si próprias.

Função ou Procedimento

Funções e procedimentos permitem modularizar algoritmos.

As funções são chamados dentro do corpo do programa principal como se fossem comandos. Após seu término, a execução continua a partir do ponto onde foi chamado.

Cada função, tem acesso às variáveis do programa que o chamou (são as variáveis globais), e tem acesso as suas próprias variáveis (são as variáveis locais), que existem apenas durante sua chamada.

Função ou Procedimento

As duas coisas, função e procedimento, possuem basicamente a mesma definição, são pequenos trechos de código que executam uma ação bem específica, ou ainda, resolvem um problema bem específico. Mas então, qual a diferença?

É simples. Imagine que vamos escrever um trecho de código para **imprimir** um vetor na tela. Perceba que a única ação a ser feita é imprimir os elementos do vetor na tela.

Função ou Procedimento

Agora, imagine que vamos escrever um trecho de código para **calcular uma soma**, a soma dos elementos de um vetor. Se vamos calcular uma soma, significa que **precisamos do resultado**, então esse resultado precisa ser retornado para quem chamou esse trecho de código.

Essa é a diferença entre função e procedimento. Uma **função realiza alguma ação e retorna algum valor**, enquanto que um **procedimento apenas realiza alguma ação**, não retornando nenhum valor.

Procedimento

Um procedimento tem o padrão a seguir, a palavra reservada **“função”** seguida do **identificador** (nome) do procedimento.

```
funcao imprimirVetor(){  
    // código para imprimir o vetor  
}
```

Procedimento

A palavra **funcao** é uma palavra reservada da linguagem portugal que indica a criação de uma função / procedimento. É sempre interessante nomear nossos procedimentos / funções com nomes intuitivos, ou seja, que deixem claro o que aquele trecho de código irá fazer.

Para utilizar nosso procedimento, basta chama-lo dentro do procedimento inicio como já estamos habituados a fazer com leia, escreva, dentro outros.

Procedimento

```
1 algoritmo "Seja_bem_vindo"
2
3 var
4
5 procedimento msg()
6 inicio
7 escreval ("=====")
8 escreval ("                Seja Bem Vindo!                ")
9 escreval ("=====")
10 Fimprocedimento
11
12 inicio
13
14 msg()
15
16 fimalgoritmo
17
```

Função

```
1 algoritmo "Soma_usando_funcao"
2
3 var
4 N1, N2, S: inteiro
5
6 Funcao Soma (A, B: inteiro): inteiro
7 inicio
8     Retorne A + B
9 FimFuncao
10
11 inicio
12 escreva ("Digite o primeiro valor: ")
13 leia (N1)
14 escreva ("Digite o segundo valor: ")
15 leia (N2)
16 S <- Soma (N1,N2)
17 escreva ("A soma entre ", N1, " e ", N2, " é igual a ", S)
18 fimalgoritmo
19
```

C:\ Console simulando o modo texto do MS-DOS

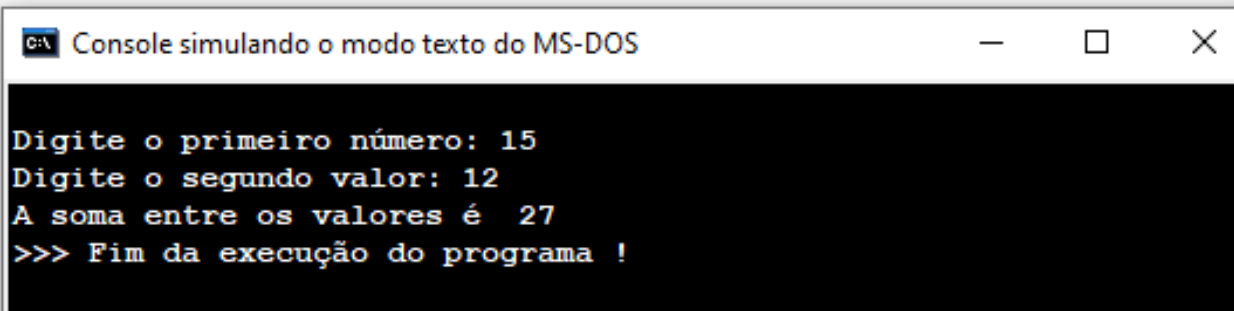
```
Digite o primeiro valor: 8
Digite o segundo valor: 6
A soma entre 8 e 6 é igual a 14
>>> Fim da execução do programa !
```

Atividade 7

Desenvolva um algoritmo que leia dois valores pelo teclado e passe esses valores para um procedimento Somador() que vai calcular e mostrar a soma entre eles.

Atividade 7

```
1 algoritmo "quest90"  
2  
3 var  
4 //Declaração das variáveis globais.  
5 N1, N2:inteiro  
6 procedimento Somador (A,B: inteiro)//Criação do procedimento com parâmetro pass.  
7 inicio //Comando de saída de dados com operação das variáveis locais.  
8 escreva ("A soma entre os valores é ",A+B)  
9 Fimprocedimento  
10 inicio  
11 //Comandos de entrada e saída de dados.  
12 escreva ("Digite o primeiro número: ")  
13 leia (N1)  
14 escreva ("Digite o segundo valor: ")  
15 leia (N2)  
16 Somador (N1,N2)//Chamada do procedimento Somador  
17 fimalgoritmo  
18
```



C:\> Console simulando o modo texto do MS-DOS

```
Digite o primeiro número: 15  
Digite o segundo valor: 12  
A soma entre os valores é 27  
>>> Fim da execução do programa !
```


Atividade 8

Crie um programa que tenha uma função `Media()`, que vai receber as 2 notas de um aluno e **retornar** a sua média para o programa principal.

Atividade 8

```
1 algoritmo "quest96"  
2  
3 var  
4 N1, N2, M: real  
5 Funcao Media (A, B: real): real  
6 inicio  
7     Retorne (A + B) / 2  
8 FimFuncao  
9  
10 inicio  
11 escreva ("Digite a 1ª nota do aluno: ")  
12 leia (N1)  
13 escreva ("Digite a 2ª nota do aluno: ")  
14 leia (N2)  
15 M <- Media (N1,N2)  
16 escreval  
17 escreva ("A média das notas do aluno foi ", M:2:1)  
18 escreval  
19 fimalgoritmo  
20
```

C:\ Console simulando o modo texto do MS-DOS

Digite a 1ª nota do aluno: 10

Digite a 2ª nota do aluno: 7

A média das notas do aluno foi 8.5

>>> Fim da execução do programa !

Dica de Curso

Curso de Algoritmos - Gustavo Guanabara

Diversos cursos na área da programação e neste caso o uso dos conceitos iniciais em lógica com uso do portugol.

Canal do Youtube



REFERÊNCIAS

K19: Lógica de Programação. Abril. 2011. Disponível em:
<http://k19.com.br>.

BERNARDO, Alessandro. **Algoritmos e Linguagens de Programação I.**
FACITEC - FACULDADE DE CIÊNCIAS SOCIAIS E TECNOLÓGICAS -
Tecnologia em Redes de Computadores.

CORMEN, Thomas H.; LEISERSON, Charles E.; RIVEST, Ronald L.;
STEIN, Clifford. **Algoritmos: teoria e prática.** 2.ed. Rio de Janeiro:
Elsevier, 2002.



ATÉ A PRÓXIMA AULA!