

Introdução à Lógica de Programação

Prof. Lucas Amparo Barbosa

Semestre letivo 2020.2

PORTUGUÊS ESTRUTURADO

- Um algoritmo precisa ser claro e sem ambiguidade;
- Com poucas palavras, em português, é possível prototipar um algoritmo.
 - Esse é o Português Estruturado. Popular **Portugol**
- As regras do Portugol são conhecidas como **sintaxe da linguagem**
 - Uma vez que você conheça lógica, basta aprender a sintaxe que você pode programar qualquer linguagem
- HORA DE INSTALAR O **VISUALG**, GALERINHA!!!!

PORTUGUÊS ESTRUTURADO

- Algumas palavras são **reservadas** da linguagem e não podem ser utilizadas como nomes de variáveis
 - leia
 - escreva

PORTUGUÊS ESTRUTURADO

O Código é dividido em três partes

- Cabeçalho
 - Informações de Autor, funcionalidade, etc. podem ser colocadas aqui
- Bloco “VAR”
 - É o local onde **todas** as variáveis utilizadas no sistema serão declaradas
- Corpo do Algoritmo
 - Vai ser onde a mágica acontece

PORTUGUÊS ESTRUTURADO - VARIÁVEIS

- Já sabemos o que são variáveis e constantes
 - Lembrem de seguir as boas práticas
- Como declarar uma variável para receber uma letra? E um número?

PORTUGUÊS ESTRUTURADO - VARIÁVEIS

- Já sabemos o que são variáveis e constantes
 - Lembrem de seguir as boas práticas
- Como declarar uma variável para receber uma letra? E um número?

Área dos algoritmos (Edição do código fonte) -> Nome do arquivo: [EXEMPLO1.ALG]

```
1 Algoritmo "exemplo1"  
2 // Disciplina : [Linguagem e Lógica de Programação]  
3 // Professor : Antonio Carlos Nicolodi  
4 // Descrição : Aqui você descreve o que o programa faz! (função)  
5 // Autor(a) : Nome do(a) aluno(a)  
6 // Data atual : 10/09/2020  
7 Var  
8 // Seção de Declarações das variáveis  
9 letra : caractere  
10 numero : inteiro  
11  
12 Inicio  
13 // Seção de Comandos, procedimento, funções, operadores, etc...  
14 Fimalgoritmo
```

OPERADORES ARITMÉTICOS

- Esses são os mais básicos e “bobos”
 - Já estamos acostumados
- Os básicos são
 - Atribuição :=
 - Atribui um valor a uma variável
 - Adição +
 - Soma dois valores
 - Subtração -
 - Subtrai dois valores
 - Multiplicação *
 - Multiplica dois valores
 - Divisão /
 - Divisão (inteira ou real) entre dois valores
 - Módulo %
 - Módulo retorna o resto de uma divisão inteira

OPERADORES ARITMÉTICOS

- Se eu quiser um algoritmo que receba dois valores e retorne a soma e a subtração dos mesmos?

OPERADORES ARITMÉTICOS

- Se eu quiser um algoritmo que receba dois valores e retorne a soma e a subtração dos mesmos?

```
1 Algoritmo "exemplo2"  
2 // Disciplina : [Linguagem e Lógica de Programação]  
3 // Professor : Antonio Carlos Nicolodi  
4 // Descrição : Aqui você descreve o que o programa faz! (função)  
5 // Autor(a) : Nome do(a) aluno(a)  
6 // Data atual : 10/09/2020  
7 Var  
8 // Seção de Declarações das variáveis  
9 a : inteiro  
10 b : inteiro  
11 soma : inteiro  
12 sub : inteiro  
13  
14 Início  
15 // Seção de Comandos, procedimento, funções, operadores, etc...  
16 leia(a)  
17 leia(b)  
18 soma := a + b  
19 sub := a - b  
20  
21 escreval("Soma = ", soma)  
22 escreval("Subtração = ", sub)  
23  
24 Fimalgoritmo
```

OPERADORES ARITMÉTICOS

- A Ordem de Precedência é similar a matemática
 - Parênteses
 - Funções da linguagem
 - Operações Unárias
 - Multiplicação, Divisão e Módulo
 - Adição e Subtração
- Como evitar confusão nessa ordem de precedência?
 - Quebre suas operações em partes

$$S = [(A + B) / (C + D)] * (E - F)$$

OPERADORES ARITMÉTICOS

- A Ordem de Precedência é similar a matemática
 - Parênteses
 - Funções da linguagem
 - Operações Unárias
 - Multiplicação, Divisão e Módulo
 - Adição e Subtração
- Como evitar confusão nessa ordem de precedência?
 - Quebre suas operações em partes

$$S = [(A + B) / (C + D)] * (E - F)$$

```
Inicio  
// Seção de Comandos, procedimento, funções, operadores, etc...  
bloco1 := A + B  
bloco2 := C + D  
bloco3 := E - F  
bloco4 := bloco1 / bloco2  
final := bloco4 * bloco3
```

OPERADORES ARITMÉTICOS

- Algumas coisas devem ser observadas nesse contexto
 - Algumas operações tem resultados REAIS mesmo que a entrada seja inteira.
 - $5 / 2 = 2.5$
 - $\text{raiz}(10) \approx 3.16$
 - Se houver uma atribuição no cálculo, ela é feita por último
 - Pode ser útil para economizar **memória**

OPERADORES RELACIONAIS

- Esses operadores são **importantes** para programação
 - Eles irão executar uma atividade entre dois valores e retornar uma resposta lógica
- Os operadores são
 - Igual a =
 - Menor que <
 - Maior que >
 - Menor ou igual a <=
 - Maior ou igual a >=
 - Diferente <>

OPERADORES RELACIONAIS

- Algumas linguagens aceitam comparações entre tipos diferentes
 - Isso pode causar erros inesperados, então **cuidado**.
- Algumas linguagens só permitem comparação de grandeza entre números
- As operações relacionais são fundamentais para gerar os fluxos do seu processamento
 - Iremos ver isso em breve

OPERADORES LÓGICOS

- Lembra da Lógica Booleana?
 - Ela pode ser implementada em algoritmos de forma simples
- Ainda lembram das operações?
 - E
 - OU
 - XOU
 - NAO

OPERADORES

- Vejam como agora podemos criar estruturas complexas
- Como eu criaria uma expressão booleana que retorna **verdadeiro** se o aluno tiver uma média maior que 7?

OPERADORES

- Vejam como agora podemos criar estruturas complexas
- Como eu criaria uma expressão booleana que retorna **verdadeiro** se o aluno tiver uma média maior que 7?

```
1 Algoritmo "exemplo3"  
2 Var  
3 // Seção de Declarações das variáveis  
4 media : real  
5 retorno : logico  
6  
7 Inicio  
8 // Seção de Comandos, procedimento, funções, operadores, etc...  
9 leia(media)  
10  
11 retorno := (media > 7)  
12  
13 Fimalgoritmo |
```

OPERADORES

- Vejam como agora podemos criar estruturas complexas
- Como eu criaria um pseudocódigo que que recebe três notas reais, calcula a média e verifica se a nota é maior que 5?

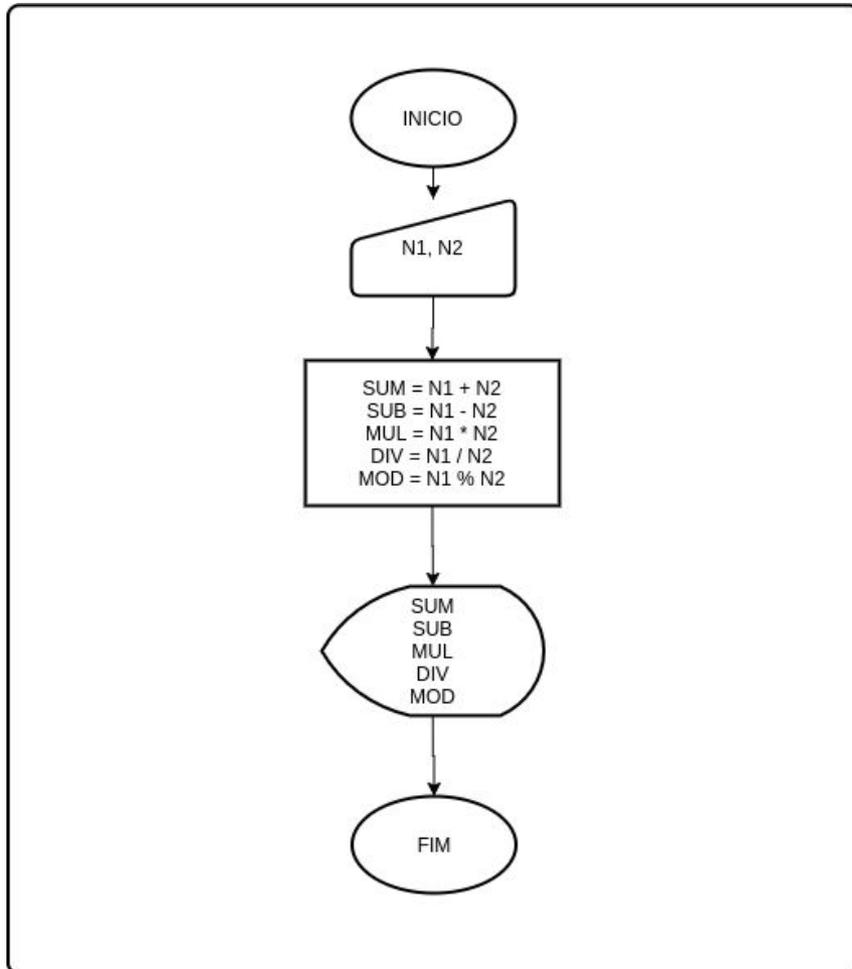
OPERADORES

- Vejam como agora podemos criar estruturas complexas
- Como eu criaria um pseudocódigo que recebe três notas reais, calcula a média e verifica se a nota é maior que 5?

```
1 Algoritmo "exemplo4"  
2 Var  
3 // Seção de Declarações das variáveis  
4 n1 : real  
5 n2 : real  
6 n3 : real  
7 media : real  
8 retorno : logico  
9  
10 Inicio  
11 // Seção de Comandos, procedimento, funções, operadores, etc...  
12 leia(n1)  
13 leia(n2)  
14 leia(n3)  
15  
16 media := (n1 + n2 + n3) / 3  
17  
18 retorno := (media > 5)  
19  
20 Fimalgoritmo
```

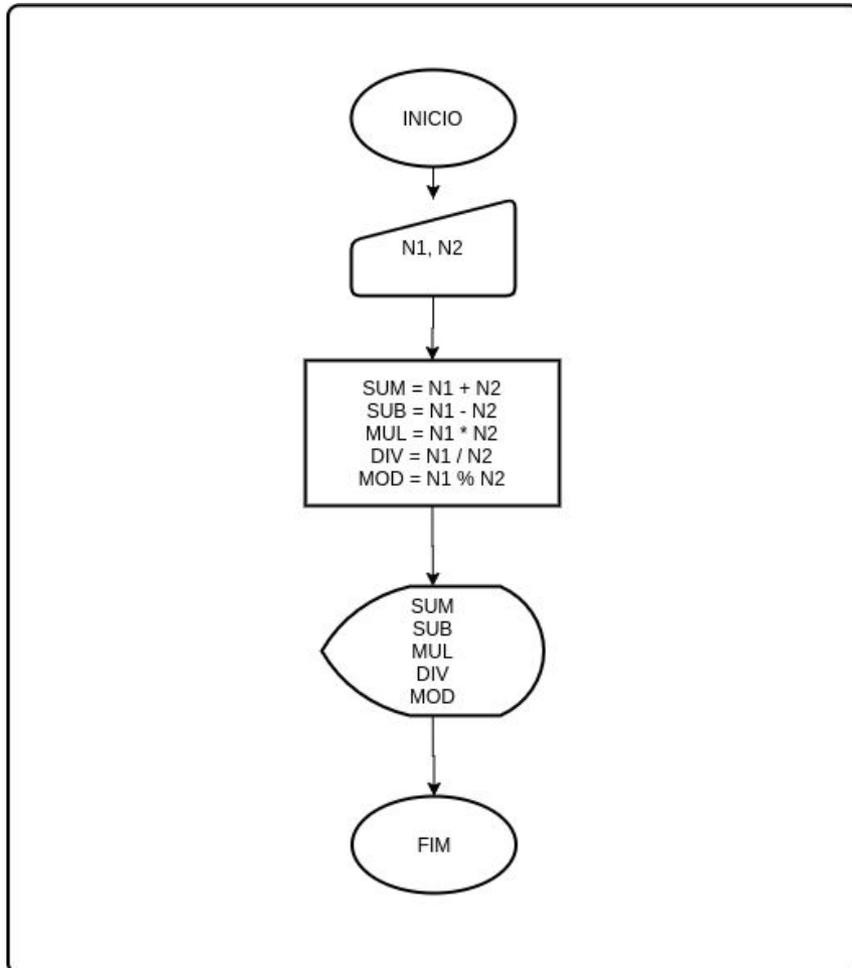
OPERADORES

- Como podemos transformar o Fluxograma abaixo em um pseudocódigo?



OPERADORES

- Como podemos transformar o Fluxograma abaixo em um pseudocódigo?



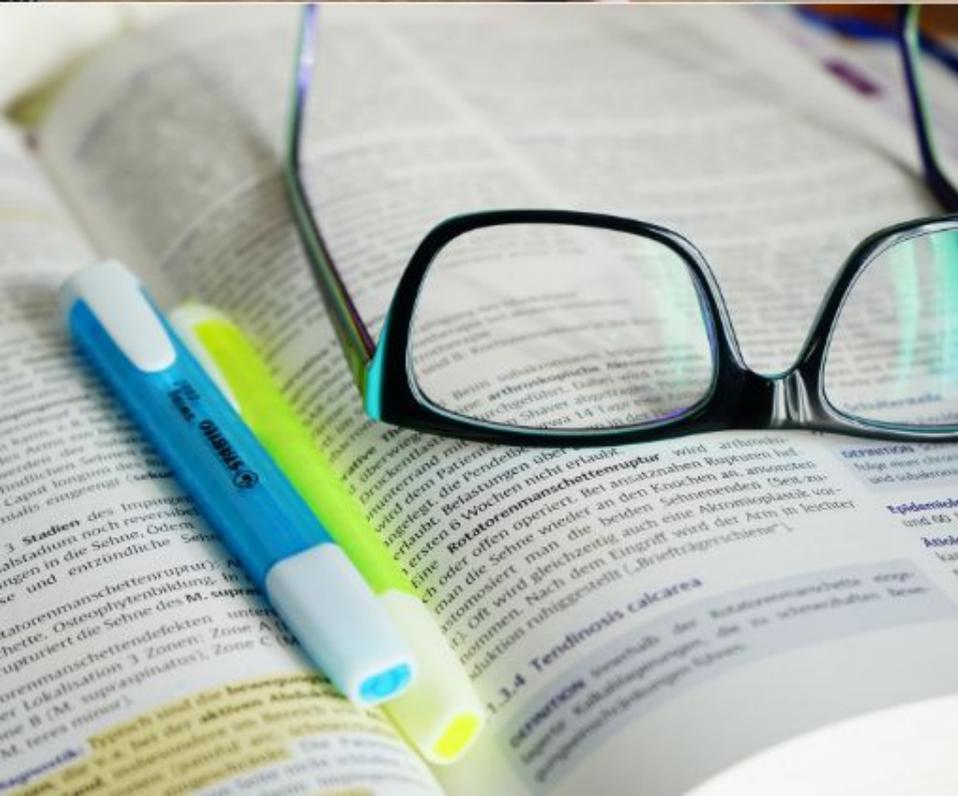
```
1 Algoritmo "exemplo5"  
2 Var  
3 // Seção de Declarações das variáveis  
4 n1 : inteiro  
5 n2 : inteiro  
6  
7 Inicio  
8 leia(n1)  
9 leia(n2)  
10  
11 escreval("Soma = ", n1 + n2)  
12 escreval("Subtração = ", n1 - n2)  
13 escreval("Multiplicação = ", n1 * n2)  
14 escreval("Divisão = ", n1 / n2)  
15 escreval("Módulo = ", n1 % n2)  
16  
17 Fimalgoritmo |
```



PARA SE DISTRAIR

[B.I.R.L. - Uma linguagem da Zoeira](#)
[B.I.R.L. - Como assim, maluco?](#)





Para saber mais...

- [Pseudocódigos](#)
- [Operadores Aritméticos](#)
- [Operadores Relacionais](#)
- [Operadores Lógicos](#)
- [Conhecendo o Visualg](#)
- [Manual do Visualg](#)