



Algoritmos e Programação

Aula 02 – Introdução à Lógica de programação



Objetivos de Aprendizagem

- Identificar na situação problema a existência de aplicar uma estrutura de repetição com teste no final;
- Entender como validar entrada de dados;
- Desenvolver algoritmos com estruturas de repetição.

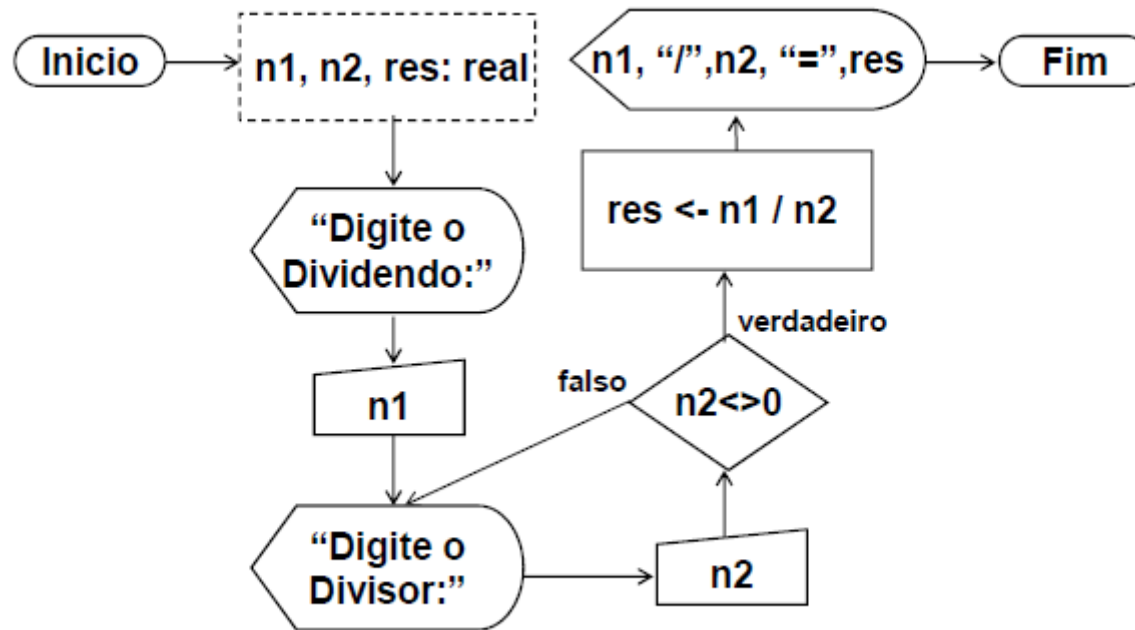


Levantamento de conhecimentos prévios

- Com base no que foi exposto, construa um fluxograma para obter o resultado da divisão entre dois números.
- **Observação:** Caso um dos operandos não seja válido o mesmo deve ser novamente solicitado até um valor válido ser fornecido, ou seja, as entradas devem ser validadas.



Levantamento de conhecimentos prévios





Levantamento de conhecimentos prévios

- Elabore um algoritmo, representando-o através de um pseudocódigo e de um fluxograma, para ler uma sequência de salários, onde o indicador (*flag*) de termino da sequência de salários é um salário igual a 0 (zero).
- O algoritmo deve escrever, em ordem crescente, os três maiores valores dos salários lidos.





Levantamento de conhecimentos prévios

- Faça um algoritmo, representando-o através de um pseudocódigo e de um fluxograma, para escrever a série de *Fibonacci* = $(0, 1, 1, 2, 3, 5, 8, 13, 21, 34, \dots)$, enquanto o valor do termo a ser escrito for menor que 5000.



Levantamento de conhecimentos prévios

- Faça um algoritmo para com base no nome, sexo ("M" = Masculino e "F" = Feminino), três notas e o número de faltas dos alunos de uma turma, onde o Flag será um nome igual a "fim", escrever:
 - A situação final de cada aluno;
 - A média das notas dos homens e a média das notas das mulheres;
 - O percentual de homens e o percentual de mulheres reprovados por média;
 - O percentual de homens e o percentual de mulheres reprovados por falta;
 - O percentual geral de reprovação da turma.
- Observação:
 - As situações possíveis são: Aprovado, Reprovado por Falta ou Reprovado por Média. A média mínima para obter aprovação é 7,00 e o limite de faltas é 15.
 - A reprovação por falta sobrepõe a reprovação por Média. As entradas devem ser validadas.



Estrutura ou laço de repetição

- Ao analisarmos o que ocorre nos laços de repetição estudados (enquanto e repita) perceberemos que, normalmente:
 - Ocorre uma inicialização de uma variável, envolvida na expressão lógica que controla o número de repetições
 - Dentro do laço ocorre uma atualização no valor da variável mencionada, fazendo com que esta venha a tornar o resultado da avaliação da expressão lógica coerente para a finalização da execução do laço de repetição.
- Com base nesta observação foi criado o laço de repetição **para**.

Laço de repetição: para

- Sintaxe:

...

```
para <variável> de <valor-inicial> ate <valor-  
limite> passo <incremento> faca
```

```
    <sequência-de-comandos>
```

```
fimpara
```

...

- <variável> é a variável contadora do número de repetições do laço (deve ser necessariamente uma variável do tipo inteiro).



Laço de repetição: para

- `<valor-inicial>` é uma expressão que especifica o valor de inicialização da variável contadora.
- `<valor-limite>` é uma expressão que especifica o valor máximo que a variável contadora pode alcançar.
- `<incremento>` é opcional. Quando presente, é precedido pela palavra-reservada `passo`, constitui-se de uma expressão que especifica o valor do incremento que será acrescentado à variável contadora em cada repetição do laço.
- O valor padrão, assumido por omissão, de `<incremento>` é 1.
- É possível especificar valores negativos para `<incremento>`.

Laço de repetição: para

- `fimpara` indica o fim da sequência de comandos a serem repetidos.
- Cada vez que o programa chega neste ponto, é acrescentado à variável contadora o valor de `<incremento>`, e o valor resultante é comparado a `<valor-limite>`.
- Se for menor ou igual (ou maior ou igual, quando `<incremento>` for negativo), a sequência de comandos será executada mais uma vez; caso contrário, a execução prosseguirá a partir do primeiro comando que esteja após o `fimpara`.
- `<valor-inicial>`, `<valor-limite>` e `<incremento>` são avaliados uma única vez antes da execução da primeira repetição, e não se alteram durante a execução do laço, mesmo que variáveis eventualmente presentes nessas expressões tenham seus valores alterados.



Exemplo

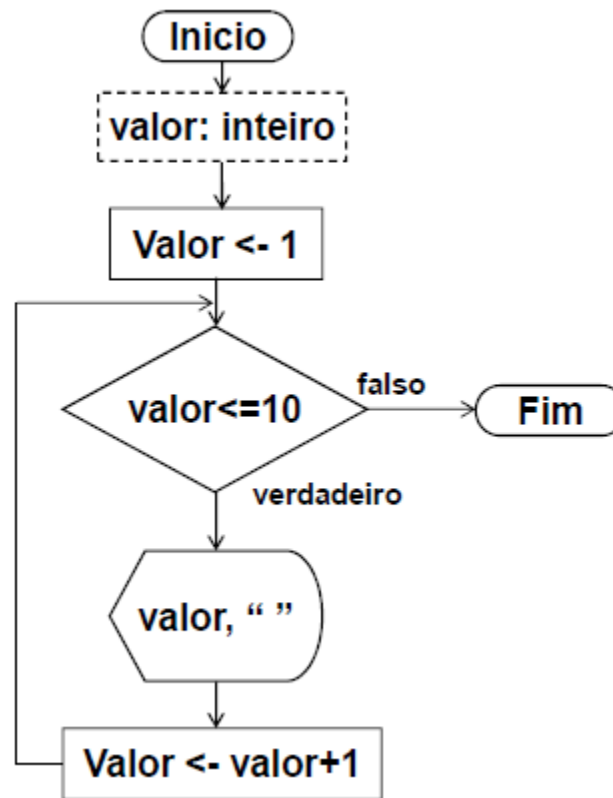
- O pseudocódigo e os fluxogramas a seguir representam algoritmos que escrevem na saída padrão os números inteiros contidos no intervalo $[1, 10]$.



Exemplo

```
algoritmo "exemplo 7"  
var  
    valor: inteiro  
inicio  
    para valor de 1 ate 10 faca  
        escreval (valor)  
    fimpara  
fimalgoritmo
```

Exemplo





Exemplo

- Dada uma reta $ax + by + c = 0$ e cinco pontos, faça um algoritmo para calcular, para cada ponto, o seguinte: se o ponto estiver no primeiro quadrante calcule e informe a distância do ponto a reta caso contrário escreva uma mensagem informando que o ponto não pertence ao primeiro quadrante.

Exemplo

```
algoritmo "exemplo 8"
Var
    a,b,c,x,y: real
    contador: inteiro
inicio
    escreval ("Equação da reta:  $ax+by+c=0$ ")
    escreva ("Coeficiente a da reta = ")
    leia (a)
    escreva ("Coeficiente b da reta = ")
    leia (b)
    escreva ("Coeficiente c da reta = ")
    leia (c)
```


Exemplo

```
para contador de 1 ate 5 faca
    escreval ("Coordenadas do ponto ",contador," :")
    escreva ("Coordenada x do ponto = ")
    leia (x)
    escreva ("Coordenada y do ponto = ")
    leia (y)
    se (x>=0 e y>=0) entao
        escreval ("A distância do ponto ",contador," a reta é: ",
( $(a*x+b*y+c)^2$ )^0.5/(( $a^2+b^2$ )^0.5))
    senao
        escreval ("O ponto ",contador," não está no primeiro
quadrante! ")
    fimse
fimpara
finalgoritmo
```



Exemplo

- Construa um pseudocódigo para representar um algoritmo que exiba na saída padrão uma contagem decrescente do valor 30 até o valor 1.



Exemplo

```
algoritmo "exercício 20"  
var  
    valor: inteiro  
inicio  
    para valor de 30 ate 1 passo -1  
faca  
    escreval (valor)  
fimpara  
finalgoritmo
```



Exemplo

```
algoritmo "exercício 20"  
var  
    valor: inteiro  
inicio  
    para valor de 0 ate 29 faca  
        escreval (30-valor)  
    fimpara  
fimalgoritmo
```



Exemplo

- Construa um algoritmo, representando-o através de um pseudocódigo e de um fluxograma, que:
 - Leia da entrada padrão um número inteiro positivo e retorne na saída padrão uma mensagem indicando se o número lido é ou não um número primo.
 - As entradas devem ser validadas.

Exemplo

```
algoritmo "exercício 21"
Var
    A, i: inteiro
    eh_primo: logico
inicio
    repita
        escreva ("Entre com um valor inteiro
positivo: ")
        leia (A)
    ate (A>0)
    eh_primo <- verdadeiro
    para i de 1+1 ate A-1 faca
```

Exemplo

```
    se (A%i=0) entao
        eh_primo <- falso
    fimse
fimpara
se (eh_primo e A<>1) entao
    escreva ("O número ",A," é primo")
senao
    escreva ("O número ",A," não é
primo")
    fimse
finalgoritmo
```



Atividade de Fixação

- Com base no que foi discutido no slide anterior construa um algoritmo que:
 - Leia da entrada padrão um número inteiro positivo e retorne na saída padrão uma mensagem indicando se o número lido é ou não um número primo.
 - As entradas devem ser validadas.



Atividade Extraclasse

- Fazer a leitura do Capítulos 5 do livro:
 - ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de. **Fundamentos de Programação de Computadores: algoritmos, Pascal, C/C++ (padrão ANSI) e Java. 3ª ed.** São Paulo: Pearson Education do Brasil, 2012. (Biblioteca Universitária Pearson)
- Atividade de Fixação: páginas 117 a 150 do livro Fundamentos da Programação de Computadores (ASCENCIO).



One Minute Paper

- Escreva um parágrafo refletindo sobre o processo de aprendizagem ocorrido na aula.
 1. O que aprendi de mais importante nesta aula?
 2. Qual a maior dúvida que ficou?
 3. Com que colegas da classe eu discuti a última aula?
- Acesso ao Google Forms:
 - bit.ly/omp-algoritmos