

## Aula 5 – Controle do Fluxo de Execução

### 1. Introdução

Muitas vezes precisamos executar ações diferentes em função dos dados de entrada. Até o momento só trabalhamos com um único fluxo de execução, veremos como controlar o fluxo para executar instruções diferentes dependendo do caso.

### 2. Estrutura Sequencial

- Instruções dispostas uma seguida da outra, como vínhamos fazendo até então.

### 3. Estruturas de Decisão

- O fluxo de instruções a ser executado depende de uma ou mais condições, ou seja, expressões lógicas (.V. ou .F.);
- Existem dois tipos: "se" (avaliação de uma única condição) e "Escolha" (avaliação de várias condições).

#### 3.1. Se

- Funciona como uma bifurcação. Uma única condição é avaliada, se for .V., um caminho é seguido. Se for .F., o outro caminho será executado;
- Sintaxe:

<b>Se</b> <Condição> <b>Então</b> <Conjunto de Comandos 1> <b>Senão</b> <Conjunto de Comandos 2> <b>Fim_se</b>	<b>Se</b> <Condição> <b>Então</b> <Conjunto de Comandos 1> <b>Fim_se</b>
--	--

- Semântica: se a "condição" é avaliada como .V., o "conjunto de comandos 1" é executado e, ao final, o programa continua na linha seguinte ao "Fim\_se". Do contrário, o "conjunto de comandos 2" é executado e o programa continua após o "Fim\_se";
- Quando não há conjunto de comandos 2 (vazio), se a condição for .F. o programa simplesmente prossegue para a próxima linha após o Fim\_se;
- Obs.: "Se", "Então", "Senão" e "Fim\_se" são palavras reservadas;
- Exemplo [Saliba, 1992, p. 62]<sup>1</sup>:

```
Algoritmo ExemploSe
VAR idade : inteiro
Início
    Leia idade
    Se idade >= 18 Então
        Escreva "Maior de idade"
    Senão
        Escreva "Menor de idade"
Fim_se
```

---

<sup>1</sup> Saliba, W. L. C. Técnicas de programação: uma abordagem estruturada. Pearson Makron Books, 1992.

**Fim.**

### 3.2. Escolha

- Funciona como o encadeamento de vários "se". Várias condições podem ser avaliadas e, para cada uma, há um caminho a ser seguido;
- Sintaxe:

**Escolha**

**Caso** <Condição 1>  
    <Conjunto de Comandos 1>

**Caso** <Condição 2>  
    <Conjunto de Comandos 2>

...

**Caso** <Condição n>  
    <Conjunto de Comandos n>

**Senão**

    <Conjunto de Comandos Senão>

**Fim\_escolha**

- Semântica: a "condição 1" é testada. Se .V., o "conjunto de comandos 1" é executado, em seguida o programa passa para a próxima linha depois do "Fim\_escolha". Se .F., a "condição 2" é avaliada e o cenário se repete. Se nenhuma das condições for .V., o "conjunto de comandos senão" é executado. Não é obrigatório ter um "senão";
- Obs.: "Escolha", "Caso" e "Fim\_escolha" são palavras reservadas;
- Exemplo [Saliba, 1992, p. 64]:

**Algoritmo** ExemploEscolha

**VAR** salário : **real**  
    cargo : **literal**

**Início**

**Leia** salário, cargo

**Escolha**

**Caso** cargo = "Técnico"  
            salário <- 1.5 \* salário

**Caso** cargo = "Gerente"  
            salário <- 1.3 \* salário

**Senão**

            salário <- 1.1 \* salário

**Fim\_escolha**

**Fim.**

## 4. Estruturas de Repetição

- Estruturas que permitem a repetição de um conjunto de instruções;
- Também chamadas de laços ou *loops*;
- Existem dois tipos: "laços contados" repetem as instruções por um número fixo de vezes previamente conhecido; "laços condicionais" repetem as instruções enquanto uma condição for .v..

### 4.1. Laços contados

- Para quando sabemos previamente quantas repetições queremos fazer;

- Sintaxe:

```
Para <var> de <início> até <final> incr de <inc> Faça  
    <Conjunto de Comandos>  
Fim_para
```

- Semântica:

- O valor "início" é atribuído à variável "var";
- Em seguida, o laço começa: o valor de "var" é comparado com o de "final" e, se "var" for maior o programa continua após o "Fim\_para";
- Se for menor ou igual, o "conjunto de comandos" é executado, é somado o valor de "inc" à variável "var" e o laço volta ao começo (comparação com "final"), se repetindo até "var" ser maior que "final".
- Se "incr de <inc>" for omitido, o incremento será de 1;
- Algumas linguagens permitem incremento negativo (substituindo a comparação de maior que para menor que).

- Obs.: "Para", "de", "até", "incr", "faça" e "Fim\_para" são palavras reservadas;

- Exemplo:

```
Algoritmo ContagemRegressiva  
VAR i : inteiro  
Início  
    Para i de 10 até 1 incr de -1 Faça  
        Escreva i  
    Fim_para  
    Escreva "Fogo!"  
Fim.
```

#### 4.2. Laços condicionais

- *Loops* executados até que uma determinada condição seja satisfeita;
- Sintaxe:

```
Enquanto <Condição> Faça                Repita  
    <Conjunto de Comandos>                <Conjunto de Comandos>  
Fim_enquanto                            até que <Condição>
```

- Semântica "Enquanto": o laço começa com a avaliação da "condição". Se .V. o "conjunto de comandos" é executado e o laço recomeça com a avaliação. Quando a "condição" for .F., o programa continua após "Fim\_enquanto";
- Semântica "Repita": o "conjunto de comandos" é executado e depois a "condição" é avaliada. Se for .F., repete-se o laço. Quando a "condição" for avaliada como .V., o programa continua após a linha "até que <Condição>";
- Obs.: "Enquanto", "faça", "Fim\_enquanto", "Repita", "até" e "que" são palavras reservadas;
- Exemplo: repetir o exemplo da contagem regressiva utilizando "Enquanto" e depois "Repita". Em seguida, ressaltar a diferença entre os dois "zerando" o contador no começo.



```
Algoritmo ContagemRegressiva2
VAR i : inteiro
Início
    i <- 10
    Enquanto i > 0 Faça
        Escreva i
        i <- i - 1
    Fim_enquanto
    Escreva "Fogo!"

    i <- 10
    Repita
        Escreva i
        i <- i - 1
    até que i = 0
Fim.
```

## 5. Aninhamentos

- Construções como *Se*, *Escolha*, *Para*, *Enquanto* e *Repita* abrem o que chamamos de "blocos", onde fica o conjunto de comandos;
- Todas estas construções podem ser incluídas dentro do conjunto de comandos, ou seja, podem estar "aninhadas" umas dentro das outras;
- Os aninhamentos devem ser tal que as construções estejam completamente contidas dentro das outras.



## Exercícios – Controle do Fluxo de Execução

- 1) Escreva um algoritmo para determinar o maior entre dois números reais dados [Saliba, 1992, p. 72].
- 2) Uma empresa decide dar um reajuste a seus funcionários de acordo com o seguinte critério: 50% de aumento para os que ganham menos de R\$ 3.000,00, 20% para os que ganham entre R\$ 3.000,00 e R\$ 10.000,00 (inclusive) e 15% para os demais. Escreva um algoritmo para calcular este reajuste [Saliba, 1992, p. 74].
- 3) Escreva um algoritmo para calcular a soma de 10 números quaisquer fornecidos pelo usuário [Saliba, 1992, p. 77].
- 4) Escreva um algoritmo para calcular os  $n$  primeiros termos de uma progressão aritmética, dados o elemento inicial  $a_0$  e a razão  $r$ .
- 5) Faça um algoritmo para imprimir os  $n$  primeiros termos da sequência de Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21, ...
- 6) Uma empresa decide presentear seus funcionários com um bônus de Natal, cujo valor é definido pelos critérios a seguir. Elabore um algoritmo para calcular o valor do bônus concedido a cada um dos 50 funcionários e o impacto de tal atitude no orçamento da empresa.
  - Os funcionários do sexo masculino com tempo de casa superior a 15 anos terão direito a um bônus de 20% do seu salário;
  - As funcionárias com tempo de casa superior a 10 anos terão direito a um bônus de 25% do seu salário;
  - Os demais funcionários terão direito a um bônus de R\$ 5.000,00.
- 7) Escreva um algoritmo para calcular o fatorial de um número inteiro.



## Resolução dos Exercícios – Controle do Fluxo de Execução

1)

```
Algoritmo Maior
Var a, b : real
Início
    Escreva "Digite dois números reais:"
    Leia a, b
    Escreva "O maior é: "
    Se a > b Então
        Escreva a
    Senão
        Escreva b
    Fim_se
Fim.
```

2)

```
Algoritmo Reajuste
Var salário : real
Início
    Escreva "Salário = "
    Leia salário
    Escolha
        Caso salário < 3000
            salário <- 1.5 * salário
        Caso salário <= 10000
            salário <- 1.2 * salário
        Senão
            salário <- 1.15 * salário
    Fim_escolha
    Escreva "Salário reajustado = ", salário
Fim.
```

3)

```
Algoritmo Soma10
Var número, soma : real
    i : inteiro
Início
    soma <- 0.0
    Para i de 1 até 10 Faça
        Escreva "Informe o ", i, "º número: "
        Leia número
        soma <- soma + número
    Fim_para
    Escreva "Soma = ", soma
Fim.
```



4)

**Algoritmo PA**

**Var** a0, r: **real**

i, n: **inteiro**

**Início**

**Leia** a0, r, n

**Para** i **de** 1 **até** n **Faça**

**Escreva** a0

    a0 <- a0 + r

**Fim\_para**

**Fim.**

5)

**Algoritmo Fibonacci**

**Var** ant1, ant2, i, n, x: **inteiro**

**Início**

**Leia** n

  ant1 <- 1

  ant2 <- 0

**Se** (n > 0) **Então**

**Escreva** 0

**Fim\_se**

**Se** (n > 1) **Então**

**Escreva** 1

**Fim\_se**

**Para** i **de** 3 **até** n **Faça**

    x <- ant1 + ant2

**Escreva** x

    ant2 <- ant1

    ant1 <- x

**Fim\_para**

**Fim.**

6)

```
Algoritmo BônusNatal
Var sexo: literal[1]
      tempoCasa, i: inteiro
      salário, bônus, impacto: real
Início
  impacto <- 0
  Para i de 1 até 50 Faça
    Leia nome, sexo, tempoCasa, salário
    Escolha
      Caso ((sexo = "M") .E. (tempoCasa >= 15))
        bônus = salário * .2
      Caso ((sexo = "F") .E. (tempoCasa >= 10))
        bônus = salário * .25
      Senão
        bônus = 5000.
    Fim_escolha
    impacto <- impacto + bônus
    Escreva nome, " terá bônus de R$ ", bônus
  Fim_para
  Escreva "Impacto no orçamento: R$ ", impacto
Fim.
```

7)

```
Algoritmo Fatorial
Var n, fat: inteiro
Início
  fat <- 1
  Leia n
  Enquanto (n > 0) Faça
    fat <- fat * n
    n <- n - 1
  Fim_enquanto
  Escreva fat
Fim.
```