Aulas – 4 e 5 – Estruturas de Seleção e Repetição

Algoritmos com Seleção

Até agora estávamos trabalhando com algoritmos puramente sequenciais, ou seja, todas as instruções eram executadas seguindo a ordem do algoritmo (normalmente, de cima para baixo). Neste capítulo começaremos a estudar estruturas de seleção. Uma estrutura de seleção, como o próprio nome já diz, permite que determinadas instruções sejam executadas ou não, dependendo do resultado de uma condição (teste), ou seja, o algoritmo vai ter mais de uma saída, uma opção que será executada de acordo com o teste realizado.

**Exemplo 4:**

Escreva um algoritmo para ler um valor. **Se** o valor lido for igual a 6, escrever a mensagem *'Valor lido é o 6'*, caso contrário escrever a mensagem *'Valor lido não é o 6'*.

Quando estivermos utilizando algoritmos com seleção, podemos utilizar dois tipos de estruturas diferentes, dependendo do objetivo do algoritmo, chamadas de "Seleção Múltipla", cujos tipos são:

Estrutura Aninhada e Estrutura Concatenada. Os tópicos a seguir, apresentam estas duas estruturas com suas características.

-Estrutura de Seleção Aninhada

A estrutura de seleção aninhada normalmente é utilizada quando estivermos fazendo várias *comparações (testes) sempre com a mesma variável*. Esta estrutura é chamada de aninhada porque na sua representação (tanto em Chapin quanto em Português Estruturado) fica *uma seleção dentro de outra seleção*.

Vamos utilizar a resposta do exercício como exemplo destes dois tipos de estruturas. Abaixo é apresentada a resposta em Chapin utilizando se a estrutura aninhada:

****

**Exemplo** de Estrutura de Seleção Aninhada

-Estrutura de Seleção Concatenada

A estrutura de seleção concatenada normalmente é utilizada quando estivermos *comparando (testando) variáveis diferentes*, ou seja, independentes entre si. Esta estrutura é chamada de concatenada porque na sua representação (tanto em Chapin quanto em Português Estruturado) as seleções ficam separadas uma da outra (não existe o lado "falso" do Chapin, ou o "Senão" do Português).

Abaixo é apresentada a resposta do exercício, utilizando a estrutura de seleção concatenada em Chapin:

****

**Exemplo** de Estrutura de Seleção Concatenada:

Como pode ser observado nessas duas respostas apresentadas para o exercício (estrutura aninhada e estrutura concatenada), existe uma grande diferença entre as duas estruturas, ou seja, uma característica de execução do algoritmo. Você saberia dizer qual é esta diferença?

-Operadores Relacionais

Operações relacionais são as *comparações* permitidas entres valores, variáveis, expressões e constantes. A Tabela 5 a seguir, apresenta os tipos de operadores relacionais.

**Símbolo Significado**

**>** maior

**<** menor

**=** igual

**> =** maior ou igual

**< =** menor ou igual

**< >** diferente

**Tabela 5:** Operadores Relacionais

A seguir, na Tabela 6, são apresentados alguns exemplos de comparações válidas:

**Comparação Válida Exemplo**

variável e constante X = 3

variável e variável A < > B

variável e expressão Y = W + J

expressão e expressão (X + 1) < (Y + 4)

**Tabela 6:** Exemplos de Comparações Válidas

-Operadores Lógicos

Os operadores lógicos permitem que mais de uma condição seja testada em uma única expressão, ou seja, pode-se fazer mais de uma comparação (teste) ao mesmo tempo. A Tabela 7 a seguir, apresenta os operadores lógicos que utilizaremos nesta disciplina.

**Operação Operador**

Negação **não**

Conjunção **e**

Disjunção (não-exclusiva) **ou**

Disjunção (exclusiva) **xou** (lê-se: *"ou exclusivo"*)

**Tabela 7:** Operadores Lógicos

Note que a Tabela 7 acima, apresenta os operadores lógicos já ordenados de acordo com suas prioridades, ou seja, se na mesma expressão tivermos o operador **ou** e o operador **não**, por exemplo, primeiro devemos executar o **não** e depois o **ou**.

De uma forma geral, os resultados possíveis para os operadores lógicos podem ser vistos na Tabela 8 abaixo, conhecida como **Tabela Verdade**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **A e B** | **A ou B** | **não A** | **A xou B** |
| **F** | **F** | **F** | **F** | **V** | **F** |
| **F** | **V** | **F** | **V** | **V** | **V** |
| **V** | **F** | **F** | **V** | **F** | **V** |
| **V** | **V** | **V** | **V** | **F** | **F** |

**Tabela 8:** Possíveis Resultados dos Testes Lógicos - Tabela Verdade

**Exemplos de Testes utilizando Operadores Lógicos:**

Exemplos usando operadores lógicos:

|  |  |
| --- | --- |
| **Expressão** | **Quando eu não saio?** |
| Se chover *e* relampejar, eu não saio. | Somente quando chover e relampejar ao mesmo  tempo (apenas 1 possibilidade). |
| Se chover *ou* relampejar, eu não saio. | Somente quando chover, somente quando  relampejar ou quando chover e relampejar ao  mesmo tempo (3 possibilidades). |
| Se chover *xou* relampejar, eu não saio. | Somente quando chover, ou somente quando  relampejar (2 possibilidades). |

**Exemplos A, B e C abaixo são apresentados em Português Estruturado:**

1. Se (**salario > 180) e** (**salário < 800)** Então

Escrever **'**Salário válido para financiamento**'**

Senão

Escrever **'**Salário fora da faixa permitida para financiamento**'**

FimSe

1. Se (**idade < 18) ou (idade > 95)** Então

Escrever **'**Você não pode fazer carteira de motorista**'**

Senão

Escrever **'**Você pode possuir carteira de motorista**'**

FimSe

1. Se (**idade > = 18) e (idade < = 95) e (aprovado\_exame = 'sim')** Então

Escrever **'**Sua carteira de motorista estará pronta em uma semana**'**

Senão

Escrever **'**Você não possui idade permitida *ou* não passou nos testes**'**

FimSe.

**Exemplos de expressões utilizando operadores lógicos**:

Para **A = V, B = F** e **C = F** , as expressões abaixo fornecem os seguintes resultados:

|  |  |
| --- | --- |
| **Expressão** | **Resultado** |
| a) não A | não V = **F** |
| b) A e B | V e F = **F** |
| c) A ou B xou B ou C | (V ou F) xou (F ou F) = V xou F = **V** |
| d) não (B ou C) | não (F ou F) = não F = **V** |
| e) não B ou C | (não F) ou F = V ou F = **V** |

**Exemplo utilizando Operadores Lógicos em Chapin:**

**5)** Escreva um algoritmo para ler um valor e escrever se ele está entre os números 1 e 10 ou não está.

Ao verificar as respostas (errada e correta) do exemplo número 5 acima, pode-se constatar que quando precisamos fazer mais de um teste ao mesmo tempo, deve-se utilizar os operadores lógicos apresentados neste capítulo na tabela 7.

-Algoritmos com Repetição

Nos exemplos e exercícios que vimos até agora, sempre foi possível resolver os problemas com uma sequencia de instruções que eram executadas apenas uma vez. Existem três estruturas básicas para a construção de algoritmos, que são: algoritmos sequenciais, algoritmos com seleção e algoritmos com repetição. A combinação dessas três estruturas permite-nos a construção de algoritmos para a resolução de problemas extremamente complexos. Nos capítulos anteriores vimos a estrutura puramente sequencial e algoritmos com seleção. Neste tópico veremos as estruturas de repetição possíveis em algoritmos e existentes na maioria das Linguagens de Programação.

Uma estrutura de repetição permite que uma sequencia de instruções (comandos) seja executada várias vezes, até que uma condição (teste) seja satisfeita, ou seja, repete-se um conjunto de instruções sem que seja necessário escrevê-las várias vezes. As estruturas de repetição também são chamadas de Laços ou Loops.

Para sabermos quando utilizar uma estrutura de repetição, basta analisarmos se uma instrução ou uma sequencia de instruções precisa ser executada várias vezes, se isto se confirmar, então deve-se utilizar uma estrutura de repetição. As estruturas de repetição, assim como a de decisão (seleção), envolvem a avaliação de uma condição (teste). Então as estruturas de repetição permitem que um trecho do algoritmo (conjunto de instruções) seja repetido um número determinado (ou indeterminado) de vezes, sem que o código correspondente, ou seja, as instruções a serem repetidas tenham que ser escritas mais de uma vez.

**Exemplo 6:**

Escreva um algoritmo para comer um cacho de uva.

A resposta apresentada (em Chapin e em Português Estruturado) está correta?

Na solução do exemplo 6*,* não foi utilizada uma ‘estrutura de repetição’, por isto **o algoritmo não está correto.** Nesse exemplo número 6 é necessária uma estrutura de repetição, pois a instrução *"comer 1 uva"* precisa ser repetida várias vezes.

Existem três tipos de estruturas de repetição: **Repita-Até**, **Enquanto-Faça** e **Para-Até-Faça**, cada uma com suas peculiaridades e apropriada para cada problema, normalmente é possível resolver um mesmo problema usando qualquer uma das estruturas de repetição, mas, na maioria das situações, haverá uma mais adequada. Neste capítulo veremos as características de cada uma destas estruturas.

Mas antes, veja a resposta do exemplo número 6 utilizando uma estrutura de repetição (Repita-Até):

Chapin Português Estruturado  ****

cacho vazio

Jogar o cacho fora

Comer 1 uva

Lavar o cacho cacho

pegar o cacho

-Estrutura de Repetição: REPITA-ATÉ

Na estrutura Repita-Até as instruções a serem repetidas são executadas, no mínimo uma vez, já que o *teste (a condição) fica no final da repetição*. Nesta estrutura, a repetição é finalizada quando o teste for Verdadeiro (V), ou seja, o algoritmo fica executando as instruções que estiverem dentro do laço até que o teste seja verdadeiro.

Nas Figuras 5 e 6 abaixo, é apresentada a forma geral da estrutura Repita-Até em Chapin e em Português Estruturado, respectivamente.



Inicio

instruções

repita

instruções

até teste

instruções

Fim

**Figura 5:** Estrutura Repita-Até em Chapin **Figura 6:** Estrutura Repita-Até em Português

Estruturado

**Observações da estrutura de repetição REPITA-ATÉ:**

1) A repetição (o laço) se encerra quando a condição (teste) for verdadeira.

2) As instruções a serem repetidas são executadas pelo menos 1 vez, porque o teste é no final da repetição.

**Pergunta:** No exemplo 6 anterior, utilizando uma estrutura de repetição, que no caso utilizou-se a estrutura Repita, o algoritmo ficou correto?

-Estrutura de Repetição: ENQUANTO-FAÇA

Na estrutura Enquanto-Faça as instruções a serem repetidas podem não ser executadas nenhuma vez, pois o *teste fica no início da repetição*, então a execução das instruções (que estão "dentro" da repetição) depende do teste. Nesta estrutura, a repetição é finalizada quando o teste é Falso (F), ou seja, enquanto o teste for Verdadeiro as instruções serão executadas e, quando for Falso, o laço é finalizado. Veja nas Figuras 7 e 8 abaixo a forma geral da estrutura Enquanto-Faça em Chapin e em Português Estruturado.



Inicio

instruções

enquanto teste faça

instruções

fim enquanto

instruções

Fim

**Figura 7:** Estrutura Enquanto-Faça em Chapin **Figura 8:** Estrutura Enquanto-Faça em

Português Estruturado

**Observações da estrutura de repetição ENQUANTO-FAÇA:**

1) A repetição (o laço) se encerra quando a condição (teste) for falsa.

2) As instruções a serem repetidas podem nunca ser executadas, porque o teste é no início da repetição.

**Atenção:** Resolva o exemplo 6 anterior, utilizando a estrutura de repetição Enquanto-Faça!

-Estrutura de Repetição: PARA-ATÉ-FAÇA

A estrutura de repetição Para-Até-Faça é um pouco diferente das outras duas (Repita-Até e Enquanto-Faça), pois possui uma **variável de controle**, ou seja, com esta estrutura é possível executar um conjunto de instruções um número determinado de vezes. Através da variável de controle, define-se a quantidade de repetições que o laço fará.

**Exemplo 7:**

Escreva um algoritmo para escrever 5 vezes a palavra ETESP na tela.

**Funcionamento da estrutura PARA:**

Na resposta do exemplo 7 acima, o X é a variável de controle, ou seja, uma variável qualquer (com qualquer nome) que vai determinar o número de repetições do laço. O valor 1 é o valor inicial que será atribuído à variável X e o valor 5 é o valor final atribuído à variável X, com isto, tem-se 5 repetições da instrução (ou das instruções) que estiver dentro do laço.

Cada vez que a variável é incrementada (aumenta +1) as instruções de dentro da repetição são executadas, então a variável, no caso o X, inicia com o valor 1 e a cada execução (repetição) ele aumenta +1 (é incrementado) até chegar ao valor final, que também é determinado (no caso é o 5).

**ATENÇÃO:** Você saberia dizer qual é a grande diferença entre a estrutura de repetição **Para** e as estruturas **Repita** e **Enquanto**?

-Dizer SIM para Continuar ou NÃO para Finalizar

**Exemplo 8:**

Escreva um algoritmo para ler dois valores. Após a leitura deve-se calcular a soma dos valores lidos e armazená-la em uma variável. Após o cálculo da soma, escrever o resultado e escrever também a pergunta **'**Novo Cálculo (S/N)?**'**. Deve-se ler a resposta e se a resposta for **'**S**'** (sim), deve-se repetir todos os comandos (instruções) novamente, mas se a resposta for **'**N**'** (não), o algoritmo deve ser finalizado escrevendo a mensagem **'**Fim dos Cálculos**'**.

-Contadores e Acumuladores

Em algoritmos com estruturas de repetição (Repita, Enquanto ou Para) é comum surgir a necessidade de utilizar variáveis do tipo contador e/ou acumulador. Neste capítulo são apresentados esses conceitos detalhadamente.

-Contadores

Um contador é utilizado para contar o número de vezes que um evento (uma instrução) ocorre, ou seja, contar a quantidade de vezes que uma instrução é executada.

Forma Geral: **VARIÁVEL VARIÁVEL + CONSTANTE**

Exemplo: X X + 1

Explicação: um contador é uma variável (qualquer) que recebe ela mesma mais um valor (uma constante), no caso do exemplo acima, a variável X está recebendo o valor dela mesma mais 1.

Normalmente a constante que será somada no contador é o valor 1, para contar de 1 em 1, mas pode ser qualquer valor, como por exemplo 2, se quisermos contar de 2 em 2.

**Observações dos Contadores:**

1) A variável (do contador) deve possuir um valor inicial conhecido, isto é, ela deve ser inicializada.

Normalmente inicializa-se a variável do contador com zero, ou seja, zera-se a variável antes de utilizá-la.

Para zerar uma variável basta atribuir a ela o valor zero: **VARIÁVEL 0**

2) A constante (que é geralmente o valor 1) determina o valor do incremento da variável (do contador), ou seja, o que será somado (acrescido) a ela.

**Exemplo 9:**

Escreva um algoritmo para ler a nota de 10 alunos e contar quantos foram aprovados, sendo que, para ser aprovado, a nota deve ser maior ou igual a 6,0. Escrever o número de aprovados.

**ATENÇÃO:** Se quiséssemos contar o número de reprovados também, no exemplo 9 acima, o que deveria ser feito?

-Acumuladores (ou Somadores)

Um acumulador, também conhecido como Somador, é utilizado para obter somatórios ( ).

Forma Geral: **VARIÁVEL1 VARIÁVEL1 + VARIÁVEL2**

Exemplo: X X + Y

Explicação: um acumulador (somador) é uma variável (qualquer) que recebe ela mesma mais uma outra variável, no caso do exemplo acima, a variável X está recebendo o valor dela mesma mais o valor da variável Y. A variável Y representa o valor a ser somado, acumulado na variável X.

**Observações dos Acumuladores:**

1) A variável1 (do acumulador) deve possuir um valor inicial conhecido, isto é, ela deve ser inicializada. Normalmente inicializa-se a variável do acumulador com zero, ou seja, zera-se a variável antes de utilizá-la. Para zerar uma variável basta atribuir a ela o valor zero: **VARIÁVEL1 0**

2) A variável2 indica o valor a ser acumulado, somado e armazenado na variável1.

**Exemplo 10:**

Altere o exemplo 9 para calcular também a média geral da turma e, depois de calculada, escrever a média.

**ATENÇÃO:** Normalmente inicializa-se as variáveis que serão utilizadas como contador ou como acumulador com o valor zero, mas pode-se inicializá-las com o valor que desejarmos de acordo com a necessidade.

-Determinação do MAIOR e/ou MENOR valor em um Conjunto de Valores

Em muitos algoritmos surge a necessidade de determinarmos qual o maior ou o menor valor dentro de um conjunto de valores e, para isto, não existe uma estrutura especial, apenas utilizamos os conhecimentos que já aprendemos, como mostrado no exemplo a seguir.

**Exemplo 11:**

Escreva um algoritmo para ler a nota de 10 alunos e escrever a nota mais alta, ou seja, a maior nota entre as 10 notas lidas.

**IMPORTANTE:** Quando sabe-se os limites dos valores possíveis, ou seja, por exemplo com as notas sabemos que os valores serão de 0 a 10, então sabe-se quais são os valores limites (o valor mínimo e o valor máximo), não teremos nota menor que 0 e nem nota maior que 10. Nesses casos é mais fácil descobrir o maior ou o menor valor, pois pode-se inicializar a variável Maior, por exemplo, com o valor 0 e a variável Menor com o valor 10 que funcionará perfeitamente. Acontece que se não sabe-se os valores dos limites aí complica um pouco, pois não sabemos com que valor vamos inicializar as variáveis para comparação. Então temos que inicializar tanto a variável Maior quanto a Menor com o “primeiro valor lido” e depois vamos comparando os próximos valores lidos com o primeiro! (os nomes “Maior” e “Menor”, são apenas exemplos, pode-se denominar as variáveis que serão usadas para os testes como quiser).

-Repetição Aninhada

Assim como vimos que é possível ter uma Seleção dentro de outra, também podemos ter uma Repetição dentro de outra, dependendo do problema a ser resolvido. Pode ser necessária uma estrutura de Repita dentro de um Enquanto, por exemplo, ou vice-versa. Ou um Repita dentro de outro Repita, enfim, as combinações são inúmeras. A seguir veremos um exemplo de uma estrutura de repetição Para dentro de outro Para, que é bastante utilizado para leitura e escrita de Matrizes, por exemplo.

**Exemplo 12:**

Dado o algoritmo a seguir (Figura 9), representado em Chapin, diga o que ele faz, ou seja, o que seria escrito na tela ao ser executado.



**Figura 9:** Exemplo de Repetição Aninhada

**Funcionamento de "um PARA dentro de outro PARA":**

**1)** A execução inicia pelo Para de fora (mais externo), depois desvia para o Para de dentro e só volta para o Para de fora quando terminar toda execução do Para de dentro (quando a variável de controle chegar no valor final).

**2)** Um Para fica "parado" enquanto o outro Para é executado, ou seja, enquanto sua variável de controle varia até chegar no valor final determinado para ela.

-Vetores

Podemos definir um Vetor como uma variável dividida em vários "pedaços", em várias "casinhas", onde cada pedaço desses é identificado através de um número, referente à posição de uma determinada informação no vetor em questão. O número de cada posição do vetor é chamado de **índice**.

Conceito: Vetor é um conjunto de variáveis, onde cada uma pode armazenar uma informação diferente, mas todas compartilham o mesmo nome. São associados índices a esse nome, que representam as posições do vetor, permitindo assim, individualizar os elementos do conjunto.

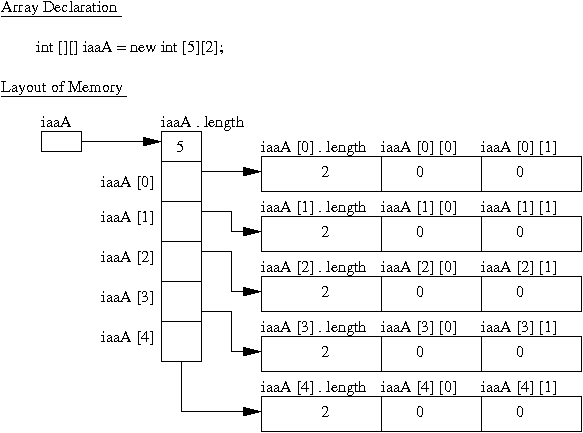
Podemos imaginar que na memória do computador o vetor seja mais ou menos da seguinte forma:



**Figura 10:** Exemplo de um Vetor na Memória do Computador



**Figura 11:** Exemplo de Vetor e Matriz



**Figura 12:** Exemplo de uma Matriz na Memória do Computador

**Exemplos de Manipulação de Vetores:**

**1) Escrever Nomes [3]**

Esta instrução escreve na tela, o conteúdo da Posição 3 do Vetor chamado Nomes, ou seja, escreve a palavra **‘**Maria**’**.

**ATENÇÃO:** Sempre que estivermos nos referindo a um Vetor, devemos colocar o **Nome do Vetor** e a **Posição** (o Índice) correspondente entre *colchetes*.

**2) Nomes [5] 'André'**

Esta instrução armazena (atribui) a palavra 'André' na posição 5 do Vetor chamado Nomes.

**3) Escrever Nomes [X]**

Esta instrução escreve o conteúdo da posição X do Vetor Nomes, ou seja, é possível utilizarmos variáveis para indicar a posição (o índice) do Vetor. Neste exemplo 3, o que será escrito depende do valor da variável X. Por exemplo, se antes da instrução Escrever, tivéssemos a instrução: **X 4**.

Então seria escrito o conteúdo da posição 4 do Vetor Nomes, neste caso, seria escrita a palavra **‘**Ana**’**.

**4) Escrever Nomes [X-2]**

Esta instrução é para mostrar que também pode-se utilizar cálculos (expressões) para indicar a posição (o índice). Digamos que o X tenha recebido o valor 4, como no exemplo 3 acima, então, neste caso, seria escrita a palavra **‘**Alex**’**.

-Como LER um Vetor (Preencher)

Para Ler um vetor, ou seja, para **preencher um vetor** com informações (dados) (armazenar informações em um vetor) é necessária uma estrutura de repetição, pois um vetor possui várias posições e temos que preencher uma a uma. A estrutura de repetição normalmente utilizada para vetores é o **Para-Até-Faça**, então veja no exemplo abaixo como preencher (ler) um vetor de 10 posições:



**Figura 11:** Exemplo para Ler um Vetor

Na Figura 11, acima, está demonstrado como preencher um vetor chamado A de 10 posições, ou seja, serão lidas 10 informações (valores, nomes, letras, etc.) e cada uma será armazenada em uma posição do vetor A. Sendo que se utiliza a própria variável da repetição “**Para”,** para representar a posição (índice) do vetor.

-Como ESCREVER um Vetor

Para Escrever um vetor, ou seja, para escrever o conteúdo de cada posição de um vetor, também precisamos utilizar uma estrutura de repetição, já que os vetores possuem mais de um conteúdo (mais de uma posição). Como explicado no tópico anterior (Como Ler um Vetor), normalmente utiliza-se a estrutura **Para-Até-Faça** também para escrever o vetor. Veja no exemplo abaixo, como escrever um vetor de 10 posições, isto é, como escrever o conteúdo de cada uma das 10 posições do vetor:



**Figura 12:** Exemplo para Escrever um Vetor

Na Figura 12, acima, está demonstrado como escrever um vetor chamado A de 10 posições, ou seja, ao executar essa instrução seria escrito o conteúdo de cada uma das 10 posições do vetor A na tela do computador.

**Exemplo 13:**

Escreva um algoritmo para ler a nota de 30 alunos, calcular a média geral da turma e escrever quantos alunos tiveram a nota acima da média calculada.

Resposta em Chapin Resposta em Português Estruturado

Início

soma 0

 media 0

cont 0

Para x de 1 Até 30 Faça

Ler nota [x]

soma soma + nota [x]

FimPara

Media soma / 30

Para n de 1 Até 30 Faça

Se nota [n] > media Então

cont cont + 1

FimSe

FimPara

Escrever cont

Fim