descomplica impulsotec

Algoritmos e Lógica de Programação: Algoritmos de repetição, matrizes e vetor

Resumo

Referências Bibliográficas

- ARAÚJO, Sandro de. **Lógica de Programação e Algoritmos [recurso eletrônico].** Curitiba: Contentus., 2020.
- FORBELLONE, André Luiz Villar; EBERSPACHER, Henri Frederico. **Lógica de Programação: a construção de algoritmos e estruturas de dados.** 3a Edição. São Paulo: Prentice Hall, 2005.

Introdução

Nesta aula, vamos ver como podemos desenvolver algoritmos que se repetem. Mais do que desenvolver algoritmos sequências e com desvios de decisão de quais blocos de comandos serão executados, para simplificar os algoritmos, utilizaremos estruturas de repetição.

Em todas as estruturas de repetição, há três informações importantes que precisam ser consideradas: a inicialização de uma variável de controle, para garantir que a entrada na estrutura de repetição aconteça adequadamente; a condição para garantir a saída da estrutura de repetição; a atualização da variável de controle, para garantir que a estrutura de repetição não fique no que a gente chama de loop infinito.

Repetição Para, Enquanto e Faça/Enquanto em Algoritmos

Temos a **estrutura de repetição do enquanto**, que é utilizada quando precisamos verificar se uma condição é verdadeira para entrar e continuar na estrutura de repetição do enquanto:

```
<inicialização da variável de controle>;
enquanto (<condição>) faça
  <comandos>;
  <atualização da variável de controle>;
fimenquanto;
```

Por exemplo: x<-1 é a inicialização da variável de controle, x<5 é a condição e x<-x+1 é a atualização da variável de controle:

```
x <- 1;
enquanto (x<5) faça
  escreva ("O valor de x é: " , x);
  x <- x + 1;
fimenquanto;
```

Temos **a estrutura de repetição do faça/enquanto**, que é utilizada quando precisamos executar os comandos da estrutura de repetição, pelo menos uma vez, independente se a condição for verdadeira ou falsa:

```
<inicialização da variável de controle>;
faça
  <comandos>;
  <atualização da variável de controle>;
enquanto(<condição>);
```

Por exemplo: x<-1 é a condição de inicialização da variável de controle, x<-x+1 é a atualização da variável de controle e x<5 é a condição para continuar na estrutura de repetição:

```
x <- 1;
faça
  escreva ("O valor de x é: " , x);
  x <- x + 1;
enquanto (x<5);
```

Temos a **estrutura de repetição do para**, que é utilizada quando sabemos a quantidade de repetições serão executadas:

```
para <variável de controle> de <valor inicial> até <valor final> passo <atualização> faça
  <comandos>;
fimpara;
```

Por exemplo: x de 1 é a condição de inicialização da variável de controle; até 4 é a condição de parada da estrutura de repetição; e +1 é a atualização da variável de controle:

```
para x de 1 até 4 passo +1 faça
  escreva ("O valor de x é: " , x);
fimpara;
```

Estrutura de Repetição em Algoritmos

Para entender a aplicação das estruturas de repetição, vamos desenvolver um algoritmo que calcula a soma de todos os números inteiros de 10 a 20, usando a estrutura de repetição do enquanto. Para isso, é importante entender que precisamos de uma variável de controle i e uma variável que vai armazenar o acúmulo da soma dos números s inicializado com 0, que é o elemento neutro da adição:

```
Algoritmo Soma
  Declarar
    i, s <- 0 numérico_inteiro;
  i <- 10; // variável de controle inicializando com 10
  // processamento de dados
  enquanto (i <= 20) faça // condição para entrar e permanecer no enquanto
    s <- s + i; // acúmulo da soma parcial a cada passo na estrutura do enquanto
    i <- i + 1; // atualizando a variável de controle para impedir o loop infinito
  fimenquanto;
  // saída de resultados
  escreva ("A somatória de 10 até 20 é " , s);
fim_algoritmo.
```

Vamos entender a aplicação da estrutura de repetição do faça/enquanto: vamos desenvolver um algoritmo que calcula a soma de todos os números inteiros de 10 a 20, usando a estrutura de repetição do faça/enquanto. Para isso, é importante entender que precisamos de uma variável de controle i e uma variável que vai armazenar o acúmulo da soma dos números s inicializado com 0, que é o elemento neutro da adição. Percebe neste caso que, independentemente do valor inicial de i; pelo menos uma vez, a estrutura de repetição do faça/enquanto é executada:

```
Algoritmo Soma
  Declarar
    i, s <- 0 numérico_inteiro;
  i <- 10; // inicialização da variável de controle
  // processamento de dados
  faça
    s <- s + i; // acúmulo da soma parcial a cada passo na estrutura do faça/enquanto
    i <- i + 1; // atualização da variável de controle
  enquanto (i <= 20); // condição para permanecer na estrutura do faça/enquanto
  // saída de resultados
  escreva ("A somatória de 10 até 20 é " , s);
fim_algoritmo.
```

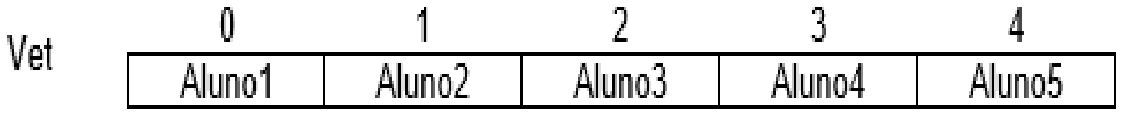
Vamos entender a aplicação da estrutura de repetição do para: vamos desenvolver um algoritmo que calcula a soma de todos os números inteiros de 10 a 20, usando a estrutura de repetição do para. Para isso, é importante entender que precisamos de uma variável de controle i e uma variável que vai armazenar o acúmulo da soma dos números s inicializado com 0, que é o elemento neutro da adição. Percebe neste caso que sabemos exatamente que queremos repetir a estrutura para de 10, de um a um até o 20:

```
Algoritmo Soma
  Declarar
    i, s <- 0 numérico_inteiro;
  // inicialização da variável de controle i de 10
  // condição para permanecer na estrutura de para até 20
  // atualização da variável de controle +1
  para i de 10 até 20 passo +1 faça
    s <- s + i; // acúmulo da soma parcial a cada passo da estrutura do para
  fimpara;
  // saída de resultados
  escreva ("A somatória de 10 até 20 é " , s);
fim_algoritmo.
```

Vetores e Matrizes em Algoritmos

Um vetor é uma condição especial de matriz. Uma matriz é um conjunto de dados dispostos em linhas e colunas. Um vetor é uma matriz que possui uma única linha com várias colunas.

Observe na figura, a seguir, um vetor chamado Vet com uma linha e cinco colunas. Temos cinco elementos: Aluno1, Aluno2, Aluno3, Aluno4 e Aluno5:



Aluno1 está na posição 0 do vetor Vet, Vet[0] = Aluno1.

Aluno2 está na posição 1 do vetor Vet, Vet[1] = Aluno2.

Aluno3 está na posição 2 do vetor Vet, Vet[2] = Aluno3.

Aluno4 está na posição 3 do vetor Vet, Vet[3] = Aluno4.

Aluno5 está na posição 4 do vetor Vet, Vet[4] = Aluno5.

Para declarar e atribuir elementos num vetor:

```
Declarar
  <nome do vetor>  [<número de elementos>]  <tipo dos elementos do vetor>;
<nome do vetor>  [<posição>]  <-  <valor>; //atribuindo valor
escrever (<nome do vetor>[<posição>]); //saída de dados
```

Por exemplo:

Declarar
// declarando um vetor chamado notas de tamanho 10, do tipo numérico_inteiro
notas [10] numérico_inteiro;
notas [0] <- 8; // atribuindo o inteiro8 na posição 0 do vetor notas
escrever ("A nota é: " , notas[0]); // saída de resultados

Vamos entender mais sobre uma matriz que é um conjunto de dados dispostos em linhas e colunas.

Observe na figura abaixo: uma matriz chamada Mat com 10 linhas e cinco colunas. Temos 10 x 5 = 50 elementos: Aluno1, Aluno2, Aluno3, ..., Aluno49 e Aluno50:

Mat	0	1	2	3	4
0	Aluno1	Aluno2	Aluno3	Aluno4	Aluno5
1	Aluno6	Aluno7	Aluno8	Aluno9	Aluno10
2	Aluno11	Aluno12	Aluno13	Aluno14	Aluno15
3	Aluno16	Aluno17	Aluno18	Aluno19	Aluno20
4	Aluno21	Aluno22	Aluno23	Aluno24	Aluno25
5	Aluno26	Aluno27	Aluno28	Aluno29	Aluno30
6	Aluno31	Aluno32	Aluno33	Aluno34	Aluno35
7	Aluno36	Aluno37	Aluno38	Aluno39	Aluno40
8	Aluno41	Aluno42	Aluno43	Aluno44	Aluno45
9	Aluno46	Aluno47	Aluno48	Aluno49	Aluno50

Aluno1 está na linha 0 e na coluna 0 da Matriz Mat, Mat[0][0] = Aluno1.
Aluno2 está na linha 0 e na coluna 1 da Matriz Mat, Mat[0][1] = Aluno2.
Aluno3 está na linha 0 e na coluna 2 da Matriz Mat, Mat[0][2] = Aluno3.
...
Aluno49 está na linha 9 e na coluna 3 da Matriz Mat, Mat[9][3] = Aluno49.
Aluno50 está na linha 9 e na coluna 4 da Matriz Mat, Mat[9][4] = Aluno50.

Para declarar e atribuir elementos numa matriz:

Declarar
<nome da matriz> [<nro linhas>] [<nro colunas>] <tipo dos elem da matriz>;
<nome da matriz> [<nro. da linha>] [<nro. da coluna>] <- <valor>;
escrever (<nome da matriz>[<nro. da linha>][<nro. da coluna>]);

Por exemplo:

```
Declarar
  // declarando uma matriz notas de 3 linhas e 2 colunas do tipo numérico_real
  notas [3][2] numérico_real;
//atribuindo o real 8.5 na linha 0 e na coluna 0 da matriz notas
notas [0][0] <- 8.5;
// saída de resultados
escrever (notas [0][0]);
```

Estrutura de Vetores e Matrizes em Algoritmos

Vamos praticar vetores desenvolvendo um algoritmo que recebe 10 valores numéricos inteiros e mostra a soma destes 10 números.

Entendendo a situação-problema: os 10 valores que vamos receber do usuário, vamos armazenar num vetor de números inteiros, com 10 posições. O processamento será calcular a soma dos 10 inteiros e a saída de resultados, mostrar o resultado desse cálculo.

```
Algoritmo Somar
  início_algoritmo
    Declarar // declaração de variáveis e/ou constantes
      // declaração do vetor VetSoma com 10 posições
      // declaração da variável soma começando com zero
      VetSoma [10] , i , soma <- 0 numérico_inteiro;
    // processamento de dados utilizando uma estrutura de repetição do para
    // passando pelas posições 0 a 9 do vetor VetSoma
    // i de 0, inicializando a variável de controle; até 9 parada da estrutura do para
    // +1, atualização da variável de controle
    para i de 0 até 9 passo + 1 faça
      escrever ("Digite um valor inteiro"); // mensagem ao usuário
      ler (VetSoma[i]); // entrada de dados, passando pelas posições de 0 a 9
      // acumulando a soma parcial a cada posição do vetor VetSoma
      soma <- soma + VetSoma[i];
    fimpara;
    escrever ("A soma dos 10 valores digitados é: " , soma); // saída de resultados
  fim_algoritmo.
```

Vamos praticar matrizes desenvolvendo um algoritmo que recebe 16 valores numéricos reais numa matriz 4x4 e mostra esses números.

Entendendo a situação-problema: os 16 valores que vamos receber do usuário, vamos armazenar numa matriz de 4 linhas e 4 colunas, de números reais, com 4x4=16 posições. O processamento será receber esses números e depois mostrar essas informações:

```
Algoritmo Mostrar
  início_algoritmo
    Declarar // declaração de variáveis e/ou constantes
      // declaração da matriz Mat com 4 linhas e 4 colunas = 16 posições
      // declaração da variável i e j de controle
      Mat [4][4] numérico_real;
      i, j numérico_inteiro;
    // processamento de dados utilizando duas estruturas de repetição encadeadas
    // i para passar pelas linhas da matriz e j para passar pelas colunas da matriz
    // variável de controle i, passando pelas linhas 0 a 3 da matriz Mat
    para i de 0 até 3 passo + 1 faça
      // variável de controle j, passando pelas colunas 0 a 3 da matriz Mat
      para j de 0 até 3 passo + 1 faça
        escrever ("Digite um valor real"); // mensagem ao usuário
        ler (Mat[i][j]); // entrada de dados para cada uma das 16 posições da matriz
        escrever (Mat[i][j]);// saída de resultados
      fimpara;
    fimpara;
  fim_algoritmo.
```

Exercícios

1. Sobre as estruturas de repetição, é INCORRETO afirmar que:
   a) Há três informações importantes que precisam ser consideradas; umas delas é a inicialização de uma variável de controle, para garantir que a entrada na estrutura de repetição acontece adequadamente.
   b) Há três informações importantes que precisam ser consideradas; uma delas é a condição para garantir a saída da estrutura de repetição.
   c) Há três informações importantes que precisam ser consideradas; uma delas é a atualização da variável de controle, para garantir que a estrutura de repetição não fique no que a gente chama de loop infinito.
   d) Há três informações importantes que precisam ser consideradas; uma delas é ter apenas uma linha de comando dentro da estrutura de repetição.
   e) Há três informações importantes que precisam ser consideradas; uma delas é para garantir que a estrutura de repetição não fique no que a gente chama de loop infinito.

2. Sobre as estruturas de repetição, é CORRETO afirmar que:
   a) Existe apenas uma forma de se escrever a estrutura de repetição, que é a estrutura de repetição do para.
   b) Existem apenas duas formas de se escrever a estrutura de repetição, que são a estrutura de repetição do enquanto e a estrutura de repetição do faça/enquanto.
   c) Existem três formas de se escrever a estrutura de repetição, que são a estrutura de repetição do para, do enquanto e do faça/enquanto.
   d) Não existem formas de se escrever estruturas de repetição.
   e) Existem muitas formas de se escrever a estrutura de repetição, você decide como desenvolvê-las.

3. Sobre a estrutura de repetição do enquanto, é CORRETO afirmar que:
   a) ela é utilizada quanto precisamos verificar se uma condição é verdadeira para entrar e continuar na estrutura de repetição.
   b) ela é utilizada quando precisamos executar os comandos da estrutura de repetição, pelo menos uma vez, independente se a condição for verdadeira ou falsa.
   c) ela é utilizada quando sabemos a quantidade de repetições serão executadas.
   d) ela é utilizada quando não precisamos realizar repetições.
   e) ela é utilizada quando precisamos repetir comandos em loop infinito.

4. Sobre a estrutura de repetição do faça/enquanto, é CORRETO afirmar que:
   a) ela é utilizada quanto precisamos verificar se uma condição é verdadeira para entrar e continuar na estrutura de repetição.
   b) ela é utilizada quando precisamos executar os comandos da estrutura de repetição, pelo menos uma vez, independente se a condição for verdadeira ou falsa.
   c) ela é utilizada quando sabemos a quantidade de repetições serão executadas.
   d) ela é utilizada quando não precisamos realizar repetições.
   e) ela é utilizada quando precisamos repetir comandos em loop infinito.

5. Sobre a estrutura de repetição do para, é CORRETO afirmar que:
   a) ela é utilizada quando precisamos verificar se uma condição é verdadeira, para entrar e continuar na estrutura de repetição.
   b) ela é utilizada quando precisamos executar os comandos da estrutura de repetição, pelo menos uma vez, independente se a condição for verdadeira ou falsa.
   c) ela é utilizada quando sabemos a quantidade de repetições serão executadas.
   d) ela é utilizada quando não precisamos realizar repetições.
   e) ela é utilizada quando precisamos repetir comandos em loop infinito.

6. Considere as seguintes afirmações sobre vetores e matrizes:
   I. Uma matriz é um conjunto de dados dispostos em linhas e colunas.
   II. Um vetor é uma matriz que possui uma única linha com várias colunas.
   III. Um vetor é uma condição especial de matriz.

   É correto afirmar que:
   a) Apenas a afirmação I é verdadeira.
   b) Apenas a afirmação II é verdadeira.
   c) Apenas a afirmação III é verdadeira.
   d) Apenas as afirmações I e II são verdadeiras.
   e) As afirmações I, II e III são verdadeiras.

Gabaritos

1. **D**
   Há três informações importantes que precisam ser consideradas: a inicialização de uma variável de controle, para garantir que a entrada na estrutura de repetição acontece adequadamente; a condição para garantir a saída da estrutura de repetição; e a atualização da variável de controle para garantir que a estrutura de repetição não fique no que a gente chama de loop infinito.

2. **C**
   Existem três formas de se escrever a estrutura de repetição. Uma delas é a estrutura do para, outra é a estrutura do enquanto e outra é a estrutura do faça/enquanto. Em todas, há três informações importantes que devem ser consideradas para entrar na estrutura de repetição, para sair dela e para impedir que ela entre em loop infinito.

3. **A**
   A estrutura de repetição do enquanto, que é utilizada quando precisamos verificar se uma condição é verdadeira para entrar e continuar na estrutura de repetição do enquanto.

4. **B**
   A estrutura de repetição do faça/enquanto, que é utilizada quando precisamos executar os comandos da estrutura de repetição, pelo menos uma vez, independente se a condição for verdadeira ou falsa.

5. **C**
   A estrutura de repetição do para, que é utilizada quando sabemos a quantidade de repetições serão executadas.

6. **E**
   Um vetor é uma condição especial de matriz. Uma matriz é um conjunto de dados dispostos em linhas e colunas. Um vetor é uma matriz que possui uma única linha com várias colunas.