LÓGICA DE **PROGRAMAÇÃO**

Esp. Marcelo Uemura

INICIAR

Introdução

Nesta unidade, serão apresentadas as estruturas de controle, como a seleção, que são utilizadas nos algoritmos para decisões que implicam em diferentes caminhos no fluxo de execução das instruções de um programa; e as estruturas de repetição em que blocos de instruções são executados de forma repetitiva.

Existem diversos cenários que necessitam de tais estruturas para a solução de problemas e otimizações de código nas estruturas sequenciais, baseadas em condições lógicas pré-definidas. No caso das estruturas de seleção, decisões sobre os caminhos a serem seguidos serão tomadas em função dos resultados das condições. Diferentes tipos de seleção podem ser utilizados, como as simples, as compostas, as encadeadas e as de múltipla escolha.

As estruturas de repetição são utilizadas no caso de repetir a execução de instruções por um número determinado de vezes, evitando a cópia de tais instruções dentro de uma estrutura sequencial.

Bons estudos!

Estrutura de Controle

Nas linguagens de programação, devem haver possibilidades para decisões sobre o fluxo do programa para que caminhos diferentes possam ser seguidos baseados em condições. Essas decisões são conhecidas como estruturas de controle.

As estruturas de controle fazem parte da estrutura de programação, compostas por instruções (comandos) que representam a lógica de um algoritmo para resolver um problema. Essas estruturas podem ser sequenciais, de seleção e de repetições (Souza et al., 2011, p.126).

As estruturas sequenciais representam ações que serão executadas em uma sequência linear, de cima para baixo (FORBELLONE; EBERSPÄCHER, 2005, p.30), de forma imperativa, sem que haja desvio algum no caminho (SOUZA et al., 2011, p.126).

As estruturas de repetição permitem que uma sequência de comandos seja executada de forma repetitiva, até que uma condição interrompa (FARRER et al., 2013, p.53).

As estruturas de seleção (desvio ou condicional) permitem a tomada de decisão sobre um caminho a ser seguido, de acordo com o resultado de uma expressão lógica (condição), segundo Souza et al. (2011, p.127). Podem ser classificadas em seleção simples, composta, encadeada e múltipla escolha.

As estruturas de seleção encadeadas representam o agrupamento de várias seleções (Souza et al., 2011, p.36). Podemos classificar a seleção do tipo encadeada em homogêneas e heterogêneas. A seguir, veremos a definição de cada uma delas.

Seleção Encadeada Homogênea

A seleção encadeada homogênea consiste em várias estruturas de seleção encadeadas que seguem um determinado padrão lógico (SOUZA et al., 2011, p.40). Podemos utilizar as estruturas SE-ENTÃO-SE e SE-SENÃO-SE, como estruturas de seleção encadeada homogêneas.

Estrutura SE-ENTÃO-SE

Essa estrutura pode ser utilizada quando uma ação (ou bloco de instruções) deve ser executada somente quando um conjunto de condições seja satisfeito.

Estrutura SE-ENTÃO-SE:

1. ***se*** *<condição 1>*
    a. ***então se*** *<condição 2>*
        i. ***então se*** *<condição 3>*
            1. ***então*** *ação 1;*
        ii. fimse;
    b. fimse;
2. fimse;

Nessa estrutura, a ação 1 só será executada se as condições encadeadas 1, 2 e 3 forem satisfeitas. Após cada se, existe um *então*, sendo que nesse tipo de

estrutura não existem *senões*, caracterizando uma homogeneidade. O fluxograma que representa essa estrutura segue a seguir:

Nas linguagens de programação C, C++ e Java, a instrução para essa estrutura é através de *if(condição) then*. A seguir, veremos uma outra estrutura de seleção encadeada, a heterogênea.

Estrutura SE-SENÃO-SE

Nessa estrutura de seleção encadeada homogênea, após cada senão existe um se, até que a ação final seja executada quando as condições anteriores não forem satisfeitas.

Estrutura SE-SENÃO-SE:

*1. **se** <condição 1>*
*a. **então** ação 1;*
*b. **senão se** <condição 2>*
i. então ação 2;
*ii. **senão se** <condição 3>*
1. então ação 3;
iii. fimse;
c. fimse;
2. fimse;

Na estrutura apresentada, a ação 3 será executada se a condição 3 for satisfeita, porém, com a dependência de que as condições 1 e 2 não foram satisfeitas. A seguir, segue o fluxograma desta estrutura.

A instrução para essa estrutura de seleção encadeada em C, C++ e Java é *if(condição) then-else*. Agora, veremos no tópico seguinte, a estrutura de seleção encadeada heterogênea.

Seleção Encadeada Heterogênea

Na seleção encadeada heterogênea, não é seguido um padrão lógico de encadeamento, como apresentado na estrutura de seleção encadeada homogênea. A seguir, um exemplo para a estrutura heterogênea:

*1. **se** <condição 1>*
*a. **então***
*i. **se** <condição 2>*
*1. **então** ação 1;*
2. senão
*a. **se** <condição 3>*
*i. **então** ação 2;*
b. fimse;
ii. fimse;
2. fimse;

No exemplo apresentado, não existe um padrão lógico no encadeamento, caracterizando uma seleção encadeada heterogênea.

Atividade

Nas estruturas de seleção encadeada, as decisões são baseadas em uma sequência de condições testadas de forma dependente, hierarquizadas. Considere o algoritmo de estrutura de seleção encadeada a seguir e faça uma análise das condições testadas.

1. início
2. inteiro: A, B;
3. leia(A);
4. leia(B);
5. se < B = A>
    a. então
        i. se < A > 5>
            1. então
                a. escreva (“B é maior que 5!”);
            2. senão
                a. escreva (“B não é maior que 5!”);
        ii. fimse;
6. fimse;
7. fim;

Para que o resultado seja “B não é maior que 5!”, a alternativa de valor para A e B que atenderia é:

- ○ **a)** A=4, B=4.
- ○ **b)** A=5, B=6.
- ○ **c)** A=6, B=6.
- ○ **d)** A=10, B=5.
- ○ **e)** A=4, B=3.

Estrutura de Seleção

Veremos neste segundo tópico um pouco sobre a estrutura de seleção múltipla escolha.

Seleção Múltipla Escolha

A estrutura de seleção múltipla escolha é aplicada quando um conjunto de valores discretos precisa ser testado e, nesse caso, diferentes ações serão executadas em função desses valores, sendo similar a uma estrutura homogênea SE-SENÃO-SE (FORBELLONE; EBESPÄCHER, 2005, p.42).

O modelo a seguir apresenta a estrutura de seleção de múltipla escolha:

*1. **escolha** X*
    *a. **caso** V1: ação 1;*
    *b. **caso** V2: ação 2;*
    *c. **caso** V3: ação 3;*
    *d. **caso contrário**: ação padrão;*
2. fimescolha

Nessa estrutura, temos que a partir do valor obtido em X, diferentes ações podem ser executadas, como a ação 1, quando o valor de X é V1. Quando o valor de X não é nenhum dos valores previstos, então, uma ação padrão pode ser executada. O fluxograma a seguir representa a estrutura de seleção de múltipla escolha.

Nas linguagens de programação C, C++ e Java, a instrução que representa a múltipla escolha é *switch(valor) - case(V1): [bloco de instruções] - default:[bloco de instruções].*

Atividade

As estruturas de seleção de múltipla escolha permitem que diferentes ações possam ser executadas em função de um valor escolhido. Nesse contexto, analise o algoritmo a seguir.

1. início
2. cadeia de caracteres: vermelho, amarelo, verde;
3. leia (sinal);
4. escolha (sinal);
    a. caso vermelho: escreva (“Pare”);
    b. caso amarelo: escreva (“Atenção”);
    c. caso verde: escreva (“Prossiga”);
    d. caso azul: escreva (“Em manutenção”);
    e. caso contrário: escreva (“Erro no sinal”);
5. fimescolha;
6. fim;

Baseado neste algoritmo, qual seria o resultado no caso do sinal lido ser laranja?

- ○ **a)** Pare.
- ○ **b)** Atenção.
- ○ **c)** Prossiga.
- ○ **d)** Em manutenção.
- ○ **e)** Erro no sinal.

Implementação das Estruturas de Decisão

As estruturas de seleção, também conhecidas como estruturas de decisão, permitem que mais de um caminho possa ser definido para o fluxo de execução, baseado em condicionais lógicas. Nesse tópico, serão vistos alguns exemplos de implementação de estruturas de decisão baseados em problemas apresentados.

Estrutura de Decisão Simples

Nessa estrutura, uma ação é executada se uma condição é testada e satisfeita.

Problema: implemente uma lógica de programação para um radar em que, no caso da velocidade de um veículo ter sido excedida acima de 60km/h, registre uma multa.

Implementação:

1. início
2. real: velocidade

3. cadeia de caracteres: placa
4. leia (placa);
5. leia (velocidade);
6. ***se*** *(velocidade >= 60)*
    a. então
        i. escreva (“multa de excesso de velocidade para:”, placa);
7. fimse;
8. fim;

Essa é uma implementação de estrutura de decisão simples, pois quando a condição é atingida, uma ação deve ser executada, não havendo ações para a condição não atingida.

Estrutura de Decisão Composta

Nessa estrutura, duas alternativas dependem de uma mesma condição (FORBELLONE; EBERSPÄCHER, 2005, p.35).

Problema: implemente uma solução de lógica de programação em que seja verificada a permissão de entrada em um estabelecimento baseado na idade do indivíduo. A entrada é permitida para maiores de idade (18 anos), porém, deve ser proibida para menores.

Implementação:

1. início
2. inteiro: idade;
3. leia (idade);
4. ***se*** *(idade >= 18)*
    a. então
        i. escreva (“entrada permitida”);
    b. senão
        i. escreva (“entrada proibida”);
5. fimse;
6. fim;

Para essa implementação, duas alternativas são definidas para uma mesma condição (idade >=18), sendo, então, uma implementação de estrutura de decisão composta.

Estrutura de Decisão Homogênea

A estrutura de decisão homogênea está relacionada a diversas condicionais encadeadas de forma homogênea.

Problema: implemente uma lógica de programação em que um morador de Curitiba, que tenha casa própria e renda superior a R$ 50.000,00, deva pagar um imposto.

Implementação:

1. início
2. lógica: casa_própria, mora_Curitiba;
3. real: renda
4. leia (mora_Curitiba);
5. leia (casa_própria);
6. leia (renda);
7. ***se*** *(mora_Curitiba)*
    a. então
    b. ***se*** *(casa_própria)*
        i. então
        ii. ***se*** *(renda >= 50.000)*
            1. então
                a. escreva ("paga imposto");
        iii. fimse;
    c. fimse;
8. fimse;
9. fim;

A lógica apresentada segue uma seleção encadeada homogênea, tendo uma sequência de encadeamento de SE-ENTÃO para decidir se o morador deve ou não pagar o imposto.

Estrutura de Decisão Heterogênea

Uma estrutura de decisão heterogênea possui condições aninhadas que permite diferentes possibilidades de caminho sem um padrão lógico (FORBELLONE; EBERSPÄCHER, 2005, p.37).

Problema: dado 3 lados de um triângulo, identifique se o mesmo é equilátero, isósceles ou escaleno (FORBELLONE; EBERSPÄCHER, 2005, p.39).

Implementação:

1. início
2. inteiro: A, B, C;
3. leia (A, B, C);
4. ***se*** *((A<B+C) e (B<A+C) e (C<A+B))*
a. então
b. ***se*** *((A=B) e (B=C))*
i. então
1. escreva (“triângulo equilátero”);
ii. senão
1. ***se*** *((A=B) ou (A=C) ou (B=C))*
a. então
i. escreva (“triângulo isósceles”);
b. senão
i. escreva (“triângulo escaleno”);
2. fimse;
c. fimse;
5. fim;

Nessa implementação, primeiro é verificada a condição de triângulo ((A<B+C) e (B<A+C) e (C<A+B)). Caso válido, é checado se todos os lados são iguais ((A=B) e (B=C)), condição para o triângulo equilátero. Senão, é checado se é um triângulo isósceles, em que dois lados são iguais e um é diferente, através da condição ((A=B) ou (A=C) ou (B=C)). Caso essa condição não seja satisfeita, conclui-se que o triângulo é escaleno, com os três lados diferentes.

Estrutura de Decisão de Múltipla Escolha

Na estrutura de decisão de múltipla escolha, baseada em um valor, vários caminhos podem ser selecionados.

Problema: construa um algoritmo que, baseado no número inteiro selecionado de 1 a 5, uma ação de escrita no sistema seja executada, conforme tabela a seguir:

Tabela 2.1: Ações do Sistema
Fonte: Elaborada pelo autor.

Implementação:

1. início
2. inteiro: opção
*3. **escolha** opção;*
    *a. **caso 1**: escreva (“ligar luzes”);*
    *b. **caso 2**: escreva (“ligar máquinas”);*
    *c. **caso 3**: escreva (“ligar aquecimento”);*

*d. **caso 4**: escreva (“ligar refrigeração”);*
*e. **caso 5**: escreva (“ligar alarme”);*
*f. **caso contrário**: escreva (“opção inválida”);*
4. fimescolha;
5. fim;

Na implementação exibida, a partir da opção selecionada, diferentes ações podem ser executadas.

Atividade

As estruturas de controle condicionais permitem que diferentes abordagens possam ser realizadas para definir caminhos nos fluxos de execução de um programa, baseado nas condições estabelecidas. No caso de haverem múltiplos caminhos em função do valor de uma variável, o tipo de estrutura de seleção conhecido que pode ser utilizado de forma mais otimizada é:

- ○ **a)** Simples.
- ○ **b)** Composta.
- ○ **c)** Encadeada homogênea.
- ○ **d)** Encadeada heterogênea.
- ○ **e)** Múltipla escolha.

Estruturas de Repetição

Segundo Souza et al. (2011), estruturas de repetição são estruturas que permitem a repetição de comandos de forma controlada, através de condições, podendo ser do tipo ENQUANTO-FAÇA, REPITA-ATÉ e PARA-ATÉ-FAÇA.

Os trechos do algoritmo nos quais ocorrem as repetições são também conhecidos como laços de repetições (loops), que apesar de se possível ter um número indeterminado de repetições, deve ser finito (FORBELLONE; EBESPÄCHER, 2005).

Estrutura ENQUANTO-FAÇA

A estrutura de repetição ENQUANTO-FAÇA é uma estrutura cujo teste da condição é feita no início, antes de cada execução do laço.

Estrutura ENQUANTO-FAÇA:

1. ***enquanto*** *<condição>* ***faça***
    a. *ação 1*

b. ação 2
c. ação N
2. fimenquanto;

Na estrutura apresentada anteriormente, enquanto a condição não for satisfeita, as ações 1, 2, N deverão ser executadas repetidas vezes. A seguir, segue a estrutura ENQUANTO-FAÇA no formato de fluxograma:

Em muitas linguagens de programação, como C, C++, Java, a instrução utilizada para esse laço de repetição é *while(condição) [bloco de instruções]*. A seguir, temos outra estrutura de repetição, conhecida como REPITA-ATÉ.

Estrutura REPITA-ATÉ

A estrutura REPITA-ATÉ é uma estrutura cuja condição é testada no final, sendo verificada após a execução do loop de repetição.

Estrutura REPITA-ATÉ:

1. repita
a. ação 1;
b. ação 2;
c. ação N;
2. ***até*** *<condição>;*

Nessa estrutura, as ações 1, 2 e N são executadas e repetidas, até que a condição de teste seja atingida. O fluxograma que representa a estrutura REPITA-ATÉ é a que segue:

Nas linguagens de programação C, C++, Java, dentre outras, a instrução utilizada para esse laço de repetição é do *[bloco de instruções] while(condição)*. A seguir, teremos mais uma estrutura de repetição, PARA-ATÉ-FAÇA.

Estrutura PARA-ATÉ-FAÇA

Conforme Souza et al. (2011), a estrutura PARA-ATÉ-FAÇA é um caso da estrutura ENQUANTO-FAÇA, na qual os comandos são repetidos com o uso de um contador que possui um valor inicial e, através de incrementos unitários, chega-se a um valor final predefinido. Essa estrutura é utilizada para casos em que é difícil determinar o número de vezes em que o bloco de comandos será executado (FORBELLONE; EBERPÄCHER, 2005).

Estrutura PARA-ATÉ-FAÇA:

1. ***para*** *V de vi* ***até*** *vf passo p* ***faça***
    a. *ação 1;*
    b. *ação 2;*

c. ação N;
2. fimpara;

Na estrutura apresentada, é utilizada uma variável de controle “V”, que recebe um valor inicial “vi” e executa loop de repetições com o incremento de “V” até que o valor final “vf” seja atingido.

A seguir, segue a estrutura PARA-ATÉ-FAÇA no formato de fluxograma:

A instrução utilizada por muitas linguagens de programação, como C, C++ e Java para esse laço de repetição é for (variável de controle, valor inicial, valor

final) [bloco de instruções].

Atividade

As estruturas de repetição permitem que ações sejam executadas de forma repetida até que determinadas condições sejam atingidas. Isso facilita a programação de blocos de instruções, que precisa ser executada de forma repetitiva, porém, finita. Considere o algoritmo a seguir e analise-o:

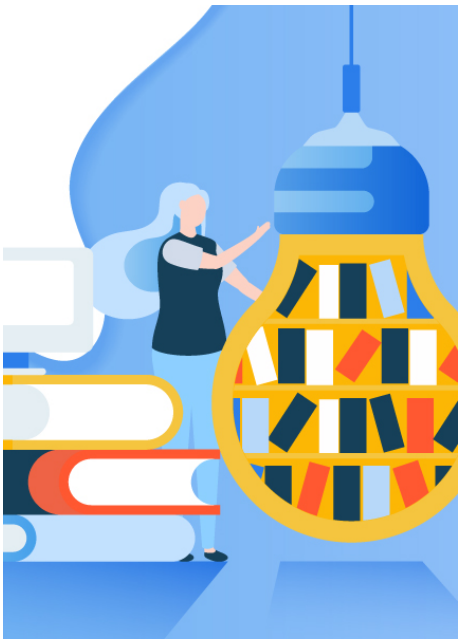
1. início
    a. inteiro: A, C;
    b. A=1;
    c. C=0;
    d. repita
        i. A = A+2;
        ii. C = C+1;
    e. até (C > 10);
    f. escreva (A);
2. fim.

Assinale a alternativa que representa o resultado final da variável A.

- ○ **a)** 23.
- ○ **b)** 21.
- ○ **c)** 25.
- ○ **d)** 22.
- ○ **e)** 24.

indicações

Material Complementar



LIVRO

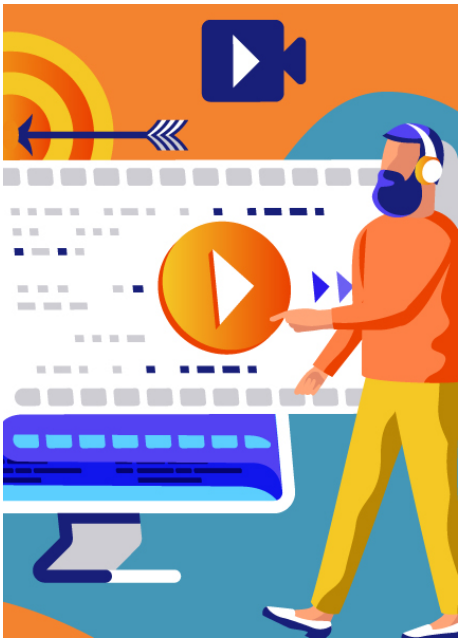
“Algoritmos e Programação”.

Editora: Novatec.

Autor: Marcos Medina e Cristina Fertig.

ISBN: 85-7522-073-X.

Comentário: esse livro expõe conceitos práticos de aplicação de algoritmos, com programas em C e Pascal, de linguagem estruturada, que facilitam o aprendizado das lógicas de programação envolvendo estruturas condicionais e de repetição.



FILME

“Laços de Repetição”.

Ano: 2017.

Comentário: esse vídeo apresenta, em Portugol, o uso de laços de repetição, comentados nesta unidade. As estruturas de repetição são colocadas com exemplos práticos de algoritmos, com sintaxe em português, facilitando o aprendizado.

TRAILER

conclusão

Conclusão

Nesta unidade, vimos as estruturas de controle de seleção e de repetição, que apoiam os algoritmos com condicionais para a execução de blocos de instrução, facilitando a criação de fluxos para diferentes caminhos que uma lógica de solução pode exigir de um programa.

Nas estruturas de seleção encadeadas, podemos implementar a dependência entre condições, sendo que uma ação pode ser tomada no caso de diversas condicionais serem atendidas ou não. Essas estruturas, conforme vistas neste conteúdo, podem ser homogêneas ou heterogêneas.

Nas estruturas de seleção de múltipla escolha, é possível definir diferentes ações em função da escolha de um valor, permitindo um conjunto variado de opções de caminhos no fluxo de um programa.

Essas estruturas ajudam o programador na flexibilização dos algoritmos para a determinação de diferentes fluxos de execução, apoiando a lógica de programação para a solução de problemas.

referências

Referências Bibliográficas

FARRER, H. BECKER, C. G.; FARIA, E. C.; MATOS, H. F.; SANTOS, M A.; MAIA, M. L. **Algoritmos Estruturados**. Rio de Janeiro: LTC, 2013.

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. **Lógica de Programação**: A construção de algoritmos e estruturas de dados. São Paulo: Prentice Hall, 2005.

MEDINA, M.; FERTIG, C. **Algoritmos e Programação**: Teoria e Prática. São Paulo: Novatec Editora, 2006.

SOUZA, M. A. F.; GOMES, M. M.; SOARES, M. V.; CONCILIO, R. **Algoritmos e Lógica de Programação**. São Paulo: Cengage Learning, 2011.

IMPRIMIR