



Algoritmos e Lógica de Programação

Estruturas de decisão ou seleção

Juliana Schiavetto Dauricio

© 2015 por Editora e Distribuidora Educacional S.A

Todos os direitos reservados. Nenhuma parte desta publicação poderá ser reproduzida ou transmitida de qualquer modo ou por qualquer outro meio, eletrônico ou mecânico, incluindo fotocópia, gravação ou qualquer outro tipo de sistema de armazenamento e transmissão de informação, sem prévia autorização, por escrito, da Editora e Distribuidora Educacional S.A.

2015

Editora e Distribuidora Educacional S. A.
Avenida Paris, 675 – Parque Residencial João Piza
CEP: 86041 -100 – Londrina – PR
e-mail: editora.educacional@kroton.com.br
Homepage: <http://www.kroton.com.br/>

Sumário

Unidade 2 Estruturas de decisão ou seleção	5
Seção 2.1 - Instruções primitivas: entrada de dados, atribuição e saída	9
Seção 2.2 - Estrutura condicional simples	25
Seção 2.3 - Estrutura condicional composta	41
Seção 2.4 - Estrutura condicional sequencial e encadeada	53

ESTRUTURAS DE DECISÃO OU SELEÇÃO

Convite ao estudo

Contextualização

Ao iniciar esta unidade de ensino, você, aluno, pode até se questionar: qual será mesmo o objetivo de aprender como fazer um algoritmo e, além disso, como escolher uma estrutura de programação que atenda às necessidades do problema que terei de solucionar através do desenvolvimento de um algoritmo, um programa ou até mesmo um aplicativo? Pois é, agora você entenderá que para cada tipo de problema há uma estrutura específica que poderá ser aplicada.

Neste contexto, entenda quais são e o que representam estas estruturas de programação. São classificadas em sequenciais, de decisão ou de repetição.

Instruções sequenciais:

São aquelas estudadas com maior ênfase na Unidade de Ensino 1. Relembre que os problemas resolvidos apresentavam como solução ações que dependiam fundamentalmente do passo anterior e, desde que a lógica estivesse correta, não havia a necessidade de tomada de decisão, e, sim, apenas a execução de ações sequenciais.

Instruções de decisão:

As instruções de decisão indicam que o programa deverá verificar se há alguma condição que valide ou invalide a operação que será realizada. Nesta unidade, você verá estas estruturas com maior ênfase. Estão inclusas as funções "se", "senão" e "se" encadeados.

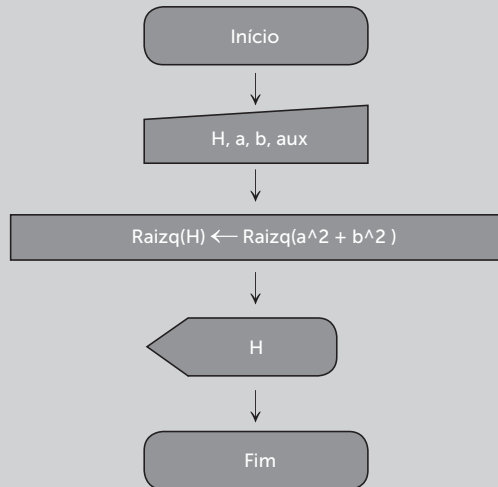
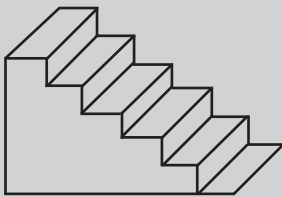
Instruções de repetição:

Estas instruções de repetição, como o próprio nome diz, indicam que o programa deverá executar uma determinada operação por mais de uma vez e, por vezes, em escala.

De acordo com Böhm e Jacopini (1966, apud SOUZA, 2013, p. 126), as estruturas sequenciais, de decisão e de repetição são denominadas estruturas primitivas de programação: *“permitem a descrição de qualquer algoritmo que seja compatível, sendo implementável em um computador. Em resumo, qualquer programa de computador pode ser escrito combinando-se esses três tipos de estruturas”*.

Observe as diferenças entre as estruturas sequenciais e as de decisão, apresentadas logo abaixo pelos seus respectivos fluxogramas. Siga em frente!

Suponha que você foi contratado para auxiliar em um escritório de projetos de construção civil e precisa determinar o padrão e o tamanho de uma escada residencial. Considere que a parede tem 8 metros de altura e a base poderá ocupar no máximo 6 metros; ao considerar o espaço e a disposição dos demais cômodos, desenvolva o fluxograma que representa esse cálculo. A recomendação é que você utilize o Teorema de Pitágoras para realizar esse cálculo, pois neste caso, a regra é: “a soma dos quadrados dos **catetos** é igual ao quadrado da **hipotenusa**”. Sendo assim, temos a fórmula: $a^2 + b^2 = c^2$. Mas observe como pode ser estruturado sequencialmente:



Já quando se tratam de estruturas de decisão, a esta sequência é agregado um teste de verificação de uma condição, para somente após este teste seguir com o processamento das operações.

Apresenta basicamente uma operação para cada resultado, seja ele verdadeiro (*true*) ou falso (*false*).

Observe o exemplo: **Agora, vamos verificar primeiro se podemos utilizar o Teorema de Pitágoras. Para isso, a soma dos quadrados dos catetos "a" e "b" tem que resultar em um ângulo de 90°.**

Mas, afinal, como verificar se este é o caso de um triângulo reto e que, portanto, pode ser resolvido através do Teorema de Pitágoras?

É importante que você saiba inserir este teste no algoritmo, utilizando uma estrutura de decisão, sendo que, neste caso, a função "SE" ou "if" atende a este requisito.

Para que possa resolver esta situação, você precisa desenvolver algumas competências e habilidades, sendo assim, destaca-se a competência de fundamentos de área.

Competência de fundamentos de área:

- Conhecer os princípios e conceitos que envolvem o aprendizado em construção de algoritmos e programação e a sua importância para o universo do desenvolvimento de sistemas.

O que é preciso saber a partir de agora para que você esteja apto a desenvolver a solução de problemas que envolvam testes e verificação de condições? Conheça os objetivos que pretendemos atingir com o estudo desta unidade de ensino.

Objetivos Específicos:

- Conhecer os conceitos e aplicações das estruturas de decisão "SE".
- Conhecer como fazer uma estrutura condicional simples.
- Conhecer e aplicar as estruturas condicionais compostas.
- Conhecer e aplicar estruturas condicionais encadeadas.

Diante do conteúdo que você conhecerá, será possível retomar a implementação que iniciou incrementando elementos condicionais

ao protótipo que será apresentado para os comerciantes do Litoral Sul. Relembrando, deverá ser elaborado um algoritmo que contemple as seguintes entregas:

- Após realizar a consulta, o usuário é direcionado a um painel com os ícones que representam o seu índice de satisfação: insatisfeito, satisfeito ou plenamente satisfeito. O prazo para desenvolvimento deste é de seis meses.
- Um algoritmo que apresenta o índice de satisfação do usuário quanto à facilidade de navegação e uso do aplicativo a partir da seguinte regra: se ele estiver satisfeito, o sistema simplesmente exibe uma mensagem de agradecimento, senão, pede ao usuário que faça uma sugestão.

Para realizar esse procedimento, primeiramente o desenvolvedor identificará:

- a. As variáveis e constantes necessárias ao desenvolvimento do algoritmo; os tipos de dados e estrutura de decisão que precisará contemplar para esta solução.
- b. Implementar esta solução através do VisuAlg.
- c. Implementar esta solução através do Dev C++.
- d. Incrementar o algoritmo com as estruturas estudadas.

Desde já, bons estudos!

Seção 2.1

Instruções primitivas: entrada de dados, atribuição e saída

Diálogo aberto

No cotidiano, muitas situações requerem uma tomada de decisão. Quando se traduz uma regra de negócio, ou seja, uma condição que precisa ser atendida, é preciso estabelecer um processo computacional que viabilize a realização deste teste. Então, como fazer com que essa regra seja mantida e, principalmente, cumprida, com a sua automatização?

Por exemplo, para o protótipo do aplicativo de divulgação de hotelaria e gastronomia solicitado pelos comerciantes do Litoral Sul, é preciso cumprir a regra de negócio que contabiliza o nível de satisfação do usuário. Para tal, você poderá resolver essa situação da seguinte forma:

- Escrever o algoritmo em linguagem natural com as respectivas necessidades de testes e verificações.
- Transcrever esse algoritmo em uma linguagem computacional. A princípio, pode ser elaborado no VisuAlg.

No entanto, alguns conceitos importantes terão de ser aplicados. De acordo com Souza (2013, p. 127), estruturas de decisão *“são estruturas que permitem a tomada de decisão sobre qual o caminho a ser escolhido, de acordo com o resultado de uma expressão lógica”*. Sendo assim, classifica-as em três formas fundamentais: SE-ENTÃO, SE-ENTÃO-SENÃO e CASO.

Você pode notar que as palavras acima (Se, Então, Senão) indicam que este é um comando condicional. Isso significa que as estruturas de decisão “Se - Então” e “Se-Então - Senão” podem representar rotinas que o sistema deverá executar para cada uma das expressões lógicas determinadas.

Já a palavra “Caso” indica que, a partir da escolha de uma opção pelo usuário, o programa executará comandos específicos para aquela determinada opção. Segundo Piva Junior (2012, p. 152), elas podem evitar erros de programação: “[...] o uso de

condições para permitir a escolha de executar ou não um trecho de programa é muito utilizado, principalmente quando precisamos incluir no programa condições de controle, para evitar situações não permitidas que podem resultar em erros”.

A seguir, as três formas fundamentais serão conceituadas e contextualizadas para que você exercite e pratique a resolução de exercícios com essas estruturas.

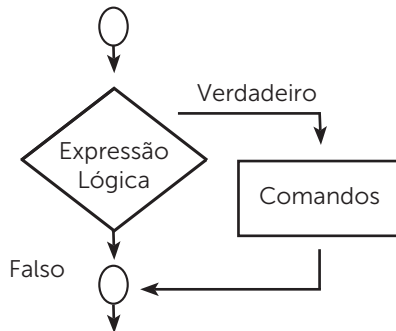
Siga em frente!

Não pode faltar

Uma expressão lógica sempre resulta em um valor verdadeiro ou falso, como mencionado anteriormente. O que fica evidenciado nesse tipo de estrutura é a necessidade de se pensar nas respostas tanto para o caso de a operação resultar em verdadeira, quanto para o caso de esta ser falsa. Isso implica o desenvolvimento de ações que contemplem as operações que o sistema deverá desenvolver para ambos casos.

Observe, abaixo, a figura que representa a lógica computacional envolvida em estruturas de decisão:

Figura 2.1 | Lógica de uma estrutura de decisão **Se-Então**



Fonte: Adaptado de Souza (2013, p. 127).

A Figura 2.1 mostra que a tomada de decisão em um sistema computacional depende de uma expressão lógica que indique o teste a realizar e, a partir do resultado, a decisão que será tomada caso seja verdadeiro ou falso o resultado.



Assimile

A lógica corresponde basicamente ao seguinte (PIVA JUNIOR et al., 2012):

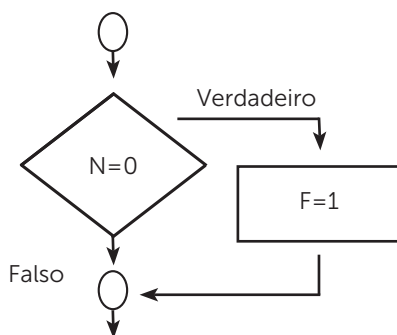
se condição **então**

comando 1

fimse

Então, por exemplo, o cálculo do fatorial de um número seria:

Figura 2.2 | Fatorial de zero



Fonte: Adaptado de Piva Junior et al. (2012, p. 155).

Você sabia que o fatorial de zero é 1?



Reflita

Não há uma fórmula para o cálculo do fatorial. [...] A condição de teste para saber se atribui o valor 1 ao resultado e então passar para o cálculo do fatorial pela fórmula é dada pela questão: n é igual a 1 (um)? [...] se n for igual a 1, então, o comando fará a atribuição à variável fatorial. (PIVA JUNIOR et al., 2012, p. 155)



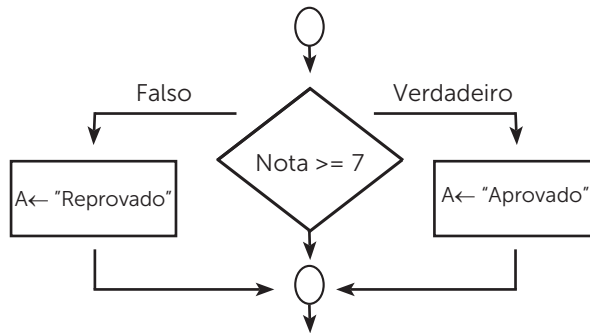
Pesquise mais

Fatorial: quando se pretende calcular o fatorial de um número, quer dizer que este apresentará o produto de todos os seus antecessores. Ex.: $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$. Disponível em: <<http://www.matematicadidatica.com.br/Fatorial.aspx>>. Acesso em: 11 maio 2015.

Vamos a mais uma forma de cálculo de fatorial para exemplificar a utilização da estrutura de decisão **Se- Então- Senão!**

A figura abaixo indica que há a inserção de um valor para verificação. Neste caso, o valor da nota que é solicitada. Em seguida, se a nota for maior ou igual a 7.0 (sete), significa que a operação a realizar é a que contém a operação para o valor verdadeiro, pois a nota inserida obedece à restrição imposta. Senão, significa que a nota inserida é menor do que 7.0 (sete) e, portanto, o programa deve executar os comandos para a operação quando esta é falsa, ou seja, apresentará o resultado para o caso de o valor inserido não atender à condição descrita no processo de verificação, através da expressão lógica "Nota \geq 7".

Figura 2.3 | Verifica nota para aprovação ou reprovação



Fonte: Adaptado de Piva Junior et al., (2012, p. 155).

Observe que, como no exemplo, a condição será sempre representada por uma expressão lógica. Veja também como deve ser representado o pseudocódigo da estrutura **se- então- senão**:

se condição **então**

comando 1

Este primeiro comando sempre será referente à operação para resultado do teste como "verdadeiro".

senão

comando 2

O segundo comando sempre será referente à operação para resultado do teste como "falso".

fimse

Note também a indentação. Isto significa que o comando 1 é executado para atender à condição expressamente representada na estrutura “se”, assim como o comando 2 pertence à condição contrária (falso). É muito importante esta representação, em função da organização do código e da identificação de referência do comando.



Refleta

As opções: então e senão são excludentes. Isto é, o algoritmo não executará as duas sequências de comandos para um mesmo valor da condição (PIVA JUNIOR et al., 2012, p. 154).

Conceituando também a estrutura de decisão CASO, obtém-se para esta a seguinte representação sintática:

escolha <apresentação das opções>

caso 1 <constante 1>

<sequência de comandos a executar para esta opção>

caso 2 <constante 2>

<sequência de comandos a executar para esta opção>

.

caso n <constante n>

<sequência de comandos a executar para esta opção>

outrocaso: < demais sequências de comandos>

Vamos compreender melhor esta estrutura?

Trata-se basicamente de uma estrutura de decisão que permite a escolha de uma delas pelo usuário. Com isso, é necessário apresentá-las primeiro (escolha ou switch em C). A seguir, a partir da seleção da opção (caso 1, caso 2), serão executados os comandos inerentes àquela opção, e o encerramento da estrutura se dá através da inserção do comando “outrocaso”, para que sejam apresentadas as instruções de comandos para o caso do usuário poder indicar, ou mesmo não aceitar, as opções que lhe foram apresentadas. Veja o exemplo em VisuAlg:

algoritmo “calculadora simples” //seção de declarações

```
var
```

```
x, y: real
```

```
opcao: caractere
```

```
inicio //seção de comandos
```

```
Escreval ("Digite dois números e informe se deseja (S)omar, (M)ultiplicar, (Sub)trair  
ou (D)ividir":)
```

```
Leia (x, y, opcao)
```

```
escolha opcao
```

```
caso "S"
```

```
    Escreval (x, opcao, y, "=", x+y)
```

```
caso "M"
```

```
    Escreval (x, opcao, y, "=", x * y)
```

```
caso "Sub"
```

```
    Escreval (x, opção, y, "=", x - y)
```

```
caso "D"
```

```
    se y <= 0 entao
```

```
        Escreval ("Informe número diferente de o (zero)!")
```

```
    senao
```

```
        Escreval (x, opcao, y, "=", x / y)
```

```
        fimse
```

```
outrocaso
```

```
    Escreval ("Opção inválida!")
```

```
fimescolha
```

```
fimalgoritmo
```

Agora que você já conhece as três principais formas das estruturas de decisão, vamos praticar um pouco!



Exemplificando

A turma "B" do curso de exatas "EPP" está fazendo aulas de revisão de matemática, no entanto, eles precisam implementar os conceitos aprendidos em um ambiente computacional. Foi lançado o seguinte desafio: elabore um algoritmo que identifique o tipo de triângulo a partir da inserção das medidas dos lados em VisuAlg:

algoritmo "classificação triângulo"

var

a, b, c: inteiro

inicio

escreval ("Digite os lados do triângulo que deseja classificar")

leia (a, b, c)

se $(a < b+c)$ e $(b < a+c)$ e $(c < a+b)$ entao

escreval ("Para as medidas:", a, " ", b, " ", c, " indicadas, é possível inferir que estas representam um triângulo:")

se $(a = b)$ e $(a = c)$ entao

escreval ("Triângulo informado é: EQUILÁTERO.")

senao

se $(a = b)$ ou $(a = c)$ ou $(b = c)$ entao

escreval ("Triângulo informado é: ISÓSCELES.")

senao

("Triângulo informado é: ESCALENO.")

fimse

fimse

senao

escreval ("As medidas informadas não representam um triângulo!")

fimse

fimalgoritmo

Observe que este exemplo apresenta inclusive o uso dos operadores lógicos **"e"** e **"ou"**.

Com isso você conheceu o uso do **"se - entao"**, **"se - entao - senao"**! Aproveite para treinar um pouco mais a linguagem de programação C e bons estudos!!!



Refleta

Quando a condição **"se"** indica o uso do operador lógico **"e"**, significa que todas as condições impostas precisam ser atendidas para que seja verdadeiro o resultado.

Quando a condição **"se"** indica o uso do operador lógico **"ou"**, significa que se apenas uma das condições impostas for atendida, é possível considerar o valor como verdadeiro.

Transcreva o algoritmo em uma linguagem de programação (preferencialmente C).

Orienta-se pelo exemplo a seguir para praticar um pouco a linguagem:



Faça você mesmo

```
#include <stdio.h>
```

```
void main ()
```

```
int a, b, c;
```

```
printf("Classificação do triângulo: informe a medida dos lados:");
```

```
scanf("%d %d %d", &a, &b, &c);
```

```

if (a < b + c && b < a + c && c < a + b)
{
    printf("\n\n Dadas as medidas: %d, %d, %d, temos um
    triângulo", a, b, c)
    if( a == b && a == c)
    {
        printf("Este é um triângulo EQUILÁTERO! \n");
    }
    else
    {
        if ( a==b | a == c | b ==c)
            printf("Este é um triângulo ISÓSCELES!
            \n");
        else
            printf("Este é um triângulo ESCALENO! \n");
        }
    }
    else
        printf("\n\n As medidas fornecidas, %d,%d,%d não formam
        um triângulo", a, b, c);
    }
    fflush (stdin);
}

```



Pesquise mais

Material de aulas e tutoriais. Disponível em: <<http://www.ic.unicamp.br/~mc102/algoritmos.html>>. Acesso em: 12 maio 2015.



Lembre-se

Estamos trabalhando na solução para a proposta de algoritmo que deverá ser entregue aos comerciantes do Litoral Sul. Estes esperam que o aplicativo a partir de agora ofereça ao usuário a possibilidade de indicar efetivamente a sua avaliação de uso do aplicativo.



Atenção!

Para tal, ele deverá escolher o tipo de serviço utilizado e o sistema mostrará as opções e o índice de satisfação, conforme recomendação abaixo.

1. Inicia.
2. Opção para escolher tipo de serviço: hotelaria ou gastronomia.
3. Solicita para cada uma delas que o usuário responda ao índice de satisfação.
4. Se o usuário indicar satisfeito, exibe mensagem de agradecimento.
5. Senão, exibe mensagem de inserção de sugestão.
6. Caso o usuário informe uma opção inexistente, solicita que informe opção existente.
7. Encerra procedimento.

Agora tente você mesmo resolver em VisuAlg. Está fácil, vamos lá!

algoritmo "Litoral Sul Fase 2.1"

// Função : apresentar opções e contabilizar acessos. Gerar média de acessos.

// Autor : JJJ

// Data : 12/05/2015

// Seção de Declarações

var

acessos, opcao2, G, H, satisfeito, insatisfeito: inteiro

opcao1: caractere

inicio

// Seção de Comandos

escreval("Olá, vamos avaliar este serviço de busca!")

escreval(" Escolha o tipo de serviço pesquisado: G - gastronomia ou H- hotelaria")

leia (opcao1)

escolha opcao1

caso "H"

 escreval ("**Índice de satisfação de usabilidade do aplicativo**")

 escreval (" Digite: 1 - satisfeito ou 2 - insatisfeito")

 leia (opcao2)

 se (opcao2 = 1) entao

 escreval ("Obrigado pela preferência!")

 senao

 escreval ("Faça a sua sugestão!")

 fimse

caso "G"

 escreval ("**Índice de satisfação de usabilidade do aplicativo**")

 escreval (" Digite: 1 - satisfeito ou 2 - insatisfeito")

 leia (opcao2)

 se (opcao2 = 1) entao

 escreval ("Obrigado pela preferência!")

 senao

 escreval ("Faça a sua sugestão!")

 fimse

outrocaso

escreval ("Informe opção existente!")

fimescolha

fimalgoritmo

Avançando na prática

Pratique mais!	
Instrução Desafiamos você a praticar o que aprendeu transferindo seus conhecimentos para novas situações que pode encontrar no ambiente de trabalho. Realize as atividades e depois compare-as com as de seus colegas e com o gabarito disponibilizado no apêndice do livro.	
Instruções primitivas: entrada de dados, atribuição e saída	
1. Competência de fundamentos de área	Conhecer os princípios e conceitos que envolvem o aprendizado em construção de algoritmos e programação e a sua importância para o universo do desenvolvimento de sistemas.
2. Objetivos de aprendizagem	<ul style="list-style-type: none">• Conhecer os conceitos e aplicações das estruturas de decisão "SE".• Conhecer como fazer uma estrutura condicional simples.• Conhecer e aplicar as estruturas condicionais compostas.• Conhecer e aplicar estruturas condicionais encadeadas.
3. Conteúdos relacionados	<ul style="list-style-type: none">• Instruções primitivas: entrada de dados, atribuição e saída• Estrutura condicional simples
4. Descrição da SP	Uma indústria de luminárias precisa mensurar, dentro de três horas, em qual dos setores houve maior nível de produção. Para tal, na troca de turno, cada líder deve lançar respectivamente os valores das suas últimas três horas de trabalho. Sendo assim, desenvolva um algoritmo que identifique e mostre o maior valor dentre os lançados. Este valor servirá de parâmetro para identificar a maior produtividade dos três turnos.
5. Resolução da SP	Em VisuAlg testar a seguinte solução: var a, b, c, maior: inteiro inicio // Seção de Comandos escreval ("Informe o primeiro número") leia (a) escreval ("Informe o segundo número") leia (b) escreval ("Informe o terceiro número") leia (c) se (a >= b) entao maior <- a

(continua)

	<pre> senao maior <- b fimse se (maior <= c) entao maior <- c fimse escreva ("O maior número é: ", maior) finalgoritmo.</pre>
--	----------------------------------------------------------------------------------------------------------------------------------------



Lembre-se

O uso dos operadores lógicos e relacionais o auxiliará na descrição da expressão que melhor representará a condição que o sistema deverá executar. Quanto mais praticar o desenvolvimento de algoritmos, melhor!

Faça valer a pena!

1. Considerado relativamente de fácil implementação, desenvolva um algoritmo que apresente uma saudação ao usuário de acordo com o seu sexo: feminino ou masculino.

A partir deste breve enunciado, assinale a alternativa que representa a declaração das variáveis:

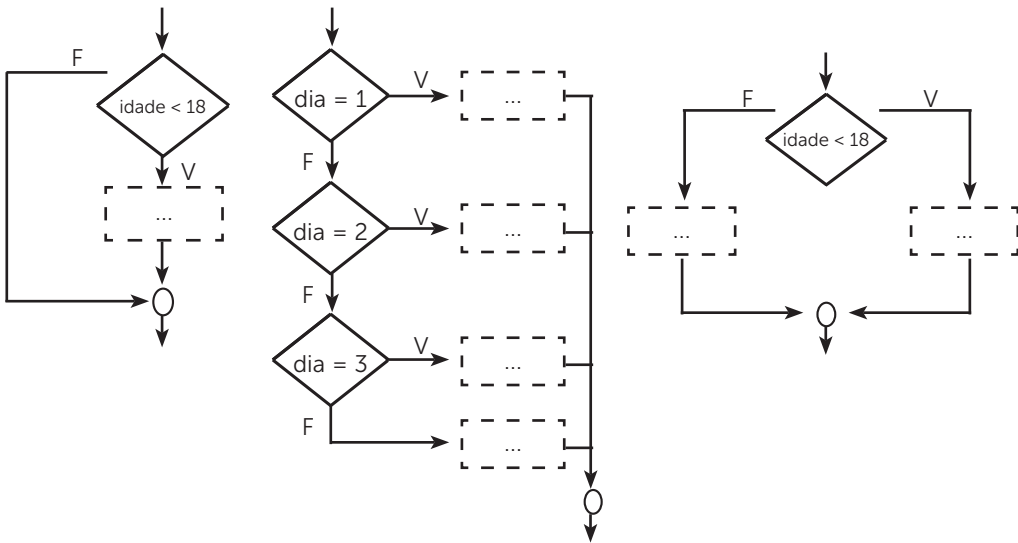
- a. N, S: caracter
- b. escreval ("Informe o seu nome")
- c. escreva ("Seja Bem-Vinda ", N) fimse
- d. inicio // Seção de Comandos
- e. se (S = "masculino") entao

2. São consideradas estruturas de decisão ou seleção o que está indicado em qual das alternativas?

- a. a, se, então
- b. caso, for, para
- c. se- então, se-então-senão, se- senão, caso
- d. while, do/while, if
- e. if/else , for, se

3. Desenvolva o algoritmo para resolver a situação apresentada no exercício 1.

4. Assinale a alternativa que apresenta a sequência descrita nas figuras:



Disponível em: <<http://www.ramon.pro.br/comandos-portugol-vs-fluxograma/>>. Acesso em: 11 maio 2015.

- se- senão, caso, se- então
- se- então, caso, se- então- senão
- caso, se, se-então
- se- senão, se- então-senão
- se, caso, então

5. Leia as afirmações e assinale a alternativa correspondente:

- Estruturas de decisão são estruturas que permitem a tomada de decisão sobre qual o caminho a ser escolhido, de acordo com o resultado de uma expressão lógica.
- Classificam-se em duas formas fundamentais: SE-ENTÃO- SENÃO e CASO.
- Quando a condição "se" indica o uso do operador lógico "ou",

significa que se apenas uma das condições impostas for atendida, é possível considerar o valor como verdadeiro.

- a. V, F e V.
- b. F, F e F.
- c. V, V e F.
- d. V, F e F.
- e. V, V e V.

6. Explique a diferença de uso dos operadores lógicos “e” e “ou”.

7. Conceitue as estruturas de decisão se- senão e se- então- senão.

Seção 2.2

Estrutura condicional simples

Diálogo aberto

Conhecer e saber aplicar as estruturas de decisão infere em ir além de simplesmente ter contato com elas. Implica identificar qual delas utilizar em face da situação que precisará resolver. Lembre-se, estamos falando das estruturas de decisão ou seleção: se- então, se-então-senão ou caso. A partir disso, você já pode aprofundar os seus conhecimentos na estrutura de decisão se- então e, no decorrer desta unidade, você terá a oportunidade de ter mais contato com as demais.

Neste sentido, retome o contexto da estrutura se- então. Observe o exemplo abaixo que pretende inserir uma verificação da idade de uma pessoa para que apresente uma resposta padrão, de acordo com a condição inserida. O uso do se- então já atende à necessidade de resolução da situação proposta. Vamos então para as instruções deste teste:

se (idade >= 65) **entao**

escreval ("Idade permitida para solicitação de recursos previdenciários!")

fimse

Observe que a **semântica** é relativamente simples do ponto de vista lógico. Mas, e para inserir mais de uma verificação com o uso de um operador lógico, como fazer? Para este mesmo exemplo, suponha que, além de ter idade igual ou superior a 65 (sessenta e cinco) anos, a pessoa também precisa ser do sexo feminino. Veja a sintaxe como fica:

se (idade >= 65 **e** sexo == feminino) **entao**

escreval ("Idade permitida para solicitação de recursos previdenciários!")

fimse

Agora, veja outro exemplo com o operador lógico "ou":

se (idade >= 65 **ou** tempoTrabalho>=30) **entao**

escreval ("Idade permitida para solicitação de recursos previdenciários!")

fimse

Com isso, você pode notar que é importante saber trabalhar com os operadores lógicos e relacionais. Estes são elementos-chave para as estruturas de decisão. Agora que você já retomou alguns elementos básicos, pode aplicá-los para desenvolver o protótipo do aplicativo da Think Now. Bons estudos e práticas a você!

Não pode faltar

Também conhecida como estrutura condicional simples, se- então pode ser utilizada para desenvolver soluções para programas que necessitem, além de realizar cálculos, atribuições, receber valores e exibir resultados ou operações, indicar condições e testá-las para a tomada de decisão.



Assimile

Condição é uma expressão lógica que, sendo verdadeira, então o comando 1 é executado e, sendo falsa, o comando 2 é executado. Somente após executado um dos comandos o controle passará para o próximo comando, após o fim do comando condicional. (PIVA JUNIOR et al., 2012, p. 163)

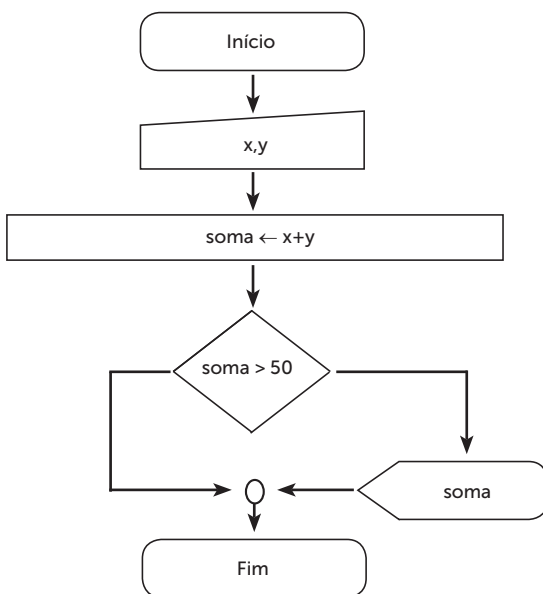
Como evidenciado acima, tenha sempre em mente que é preciso verificar se apenas uma condição será suficiente para atender à necessidade de solução do problema que foi apresentado. Também neste contexto é possível afirmar que o programa, após verificar a condição, se esta for falsa, não retornará valor algum referente ao teste, e sim, simplesmente, dará continuidade no processamento das informações.

Uma vez que este desvio para a realização do teste não resultou em verdadeiro, não há a necessidade de testar os comandos inerentes a esta operação. Contudo, não se esqueça também de que a melhor forma de gravar uma informação é realizando exercícios. Para tal, busque sempre identificar:

1. O cenário e o processo que deverá ser customizado.
2. Qual estrutura de decisão se adequa melhor à situação proposta.
3. Desenvolva a condição de forma a atender às regras de negócio que precisam ser testadas para se realizar a tomada de decisão.

Tendo isso em mente, você já pode iniciar estas práticas. Siga em frente! Vamos revisar o exemplo da soma entre dois números e, agora, incrementando uma condição: o programa deve exibir o valor da soma caso esta seja superior a 50.

Primeiramente observe o exemplo de fluxograma que Souza (2013, p. 61) utiliza para explicar esta operação:



O desvio na operação indica, justamente, que após a verificação do teste, se o seu resultado for verdadeiro diante da condição imposta, o programa exibirá o resultado na tela.

E então, caso o resultado seja falso, o procedimento adotado é continuar a execução dos comandos descritos na sequência.



Exemplificando

Observe o exemplo que implementa esta situação com a inserção de uma condição na estrutura “se- então” em pseudocódigo:

algoritmo Exibe Soma

var

x, y, soma: inteiro

inicio

escreval ("Informe o valor de dois números inteiros:")

leia (x, y)

soma \leftarrow x + y

se (soma > 50) **então**

escreval ("O valor da soma é:", soma)

fimse

fimalgoritmo



Faça você mesmo

Agora, tente implementar esta estrutura inserindo a condição simples, utilizando a linguagem de programação C. Isso facilitará a sua compreensão da dinâmica do algoritmo descrito em pseudocódigo e deste já implementado em uma linguagem de programação. Vamos lá!

Observe e teste: seguindo praticamente o mesmo raciocínio utilizado no algoritmo anterior em pseudocódigo, desenvolva um programa que recebe dois números inteiros, os exibe na tela e, ainda, verifica qual deles é maior, exibindo, por fim, o que for maior entre eles. Veja a seguir uma sugestão para esta solução:

```
/*Programa que recebe dois valores, verifica qual deles é maior e exibe
na tela*/

#include <stdio.h>

main(){
float x, y, maior;

printf("Informe dois números:");

scanf("%f%f", &x, &y);

maior = x;

if (y > x) {

        maior = y;

}
```

```
printf("Os valores informados foram %f%f", &x, &y);  
printf("O maior valor dentre os números digitados é: %f", &maior);  
}
```

Vamos entender o programa anteriormente descrito?

Primeiramente você pode notar a presença de um comentário, pela indicação da frase que vem dentro dos símbolos `/* */`. Toda vez que tiver de inserir um comentário, seja para iniciar o programa, seja para explicar um comando ou outra estrutura, você poderá usar esta referência. Em seguida, há a descrição da biblioteca que deve ser chamada para buscar os comandos de entrada e saída de dados, atribuição, entre outros. Esta vem logo no início com `#include`, que justamente inclui neste programa, quando solicitado um comando que esteja nesta biblioteca, as funções que serão executadas.

Além desta, o exemplo anterior retoma comandos de entrada e saída como `printf` e `scanf`. Observe ainda que a estrutura condicional simples é utilizada, e de acordo com a sintaxe da linguagem de programação C. Por exemplo, a palavra `se` é escrita em inglês `if`, e indica o início dessa estrutura de verificação da condição. Esta, inclusive, será comum a outras plataformas e linguagens de programação.

A condição é inserida entre parênteses `()` e, na sequência, vem o comando que deverá ser executado caso esta condição seja verdadeira. Se a condição resultar em falso, o programa executará a expressão diretamente descrita após o encerramento do comando `if`. Para este algoritmo o que se tem é a exibição dos números digitados e também do maior deles.

Na estrutura condicional simples, como é chamada, é comum encontrar situações em que há necessidade de se implementar uma estrutura condicional composta, agregando à estrutura simples anteriormente exemplificada um outro comando a realizar dentro desta mesma etapa, caso a resposta ao teste estabelecido seja falsa. Neste caso, é preciso utilizar o `se-então-senão`.



Refleta

- *As palavras se, então e senão representam o comando condicional;*
- *A condição deve ser uma expressão lógica;*
- *O comando avalia a condição. Se o resultado da expressão for verdadeiro, então será executado o comando 1. Mas, se o resultado for falso, será executado o comando 2 (PIVA JUNIOR et al., 2012, p. 153).*

No mesmo raciocínio do exercício anterior, podemos praticar mais um pouco! Vamos agora seguir o exemplo que Evaristo (2001, p. 154) traz para este exercício de desenvolvimento do raciocínio lógico e compreensão da sintaxe em C. O programa consiste em solicitar a inserção de um número decimal qualquer, seja ele uma **dízima periódica** ou não. Observe no exemplo que os dados podem ser determinados pelo usuário. Se pensar em uma outra aplicação, os dados poderão ser atribuídos também na declaração das variáveis sendo, assim, um valor constante.

Neste contexto, vamos verificar como fazer o algoritmo que arredonda um número. Siga em frente utilizando o Dev C++, de preferência, como plataforma de desenvolvimento.

```
/*Programa para arredondar valores*/

#include <stdio.h>

main()
{
    float num, parteFracionada;
    int Arredondamento;
    printf("Digite um número:");
    scanf("%f", &num);
    Arredondamento = num;
    parteFracionada = num - Arredondamento;
    if (parteFracionada >= 0.5)
```

```

        Arredondamento = Arredondamento + 1;

    printf("O valor digitado %f quando arredondado obtém-se %d", num,
    Arredondamento);

}

```

Esta operação é considerada de fácil compreensão e manipulação. Por esse motivo, além do VisuAlg, é importante que você refaça todos os exercícios sugeridos também no Dev C++, a fim de desenvolver maior autonomia, seja quanto ao uso e conhecimento da linguagem, seja quanto à construção do raciocínio lógico que advém das práticas.

Agora vamos praticar mais um pouco, com o desenvolvimento de um algoritmo que verifica um dado textual, ou seja, um tipo de dado caractere ou char. Suponha que você precise desenvolver um algoritmo que solicite ao usuário a inserção do nome, da idade e do sexo.

Este algoritmo deverá verificar se o conteúdo informado pelo usuário referente à variável sexo é feminino ou masculino, e caso seja diferente de uma destas opções, o programa exibe uma mensagem ao usuário para que este escolha apenas uma das opções apresentadas. Vamos lá!

```

#include <stdio.h>

#include <stdlib.h>

main()
{
    char sexo, nome;
    int  idade;
    system("cls");
        sexo = ' ';

    printf("\nInforme o seu nome: \n");
    printf("\nInforme a sua idade: \n");
    printf("\nDigite F para o sexo feminino ou M para masculino:\n");
    scanf("%c%d%c", &nome, &idade, &sexo);
    if (sexo != 'F' && sexo != 'M')

```

Este comando limpa a tela de exibição dos dados que serão imediatamente apresentados.


```

{
    printf ("Dado inválido.");
}

printf ("\n\n");

system("pause");

return (0);

}

```

Este comando faz com que a tela de exibição dos resultados (DOS) permaneça aberta logo após o processamento das operações. Não feche automaticamente.

Muito bem, até aqui foram apresentados vários algoritmos que utilizam-se da estrutura de decisão condicional simples “se-então”. Além de conhecer e poder aplicar em VisuAlg, você também pode empregar esta prática diretamente no ambiente de desenvolvimento Dev C++, conforme recomendado para os exercícios anteriores.

A partir de agora, recomenda-se também que você investigue um pouco mais sobre outros recursos computacionais que permitem o desenvolvimento de algoritmos. Um dos exemplos que passaremos a indicar é o Scilab. Abaixo, algumas recomendações de leituras para a sua familiarização com o ambiente.



Pesquise mais

Abaixo, seguem dois *links* que apresentam uma linguagem de programação também bastante utilizada para o ensino e desenvolvimento de algoritmos: Scilab.

- Apostila de Scilab da Universidade Federal do Rio Grande do Norte. Acaba de ser disponibilizada para consultas em: <<http://www.dca.ufrn.br/~estefane/academica/progsci.pdf>>. Acesso em: 19 maio 2015.
- Apostila de Scilab disponibilizada pela Universidade Estadual de Campinas: <http://www.ime.unicamp.br/~encpos/VIII_EnCPos/Apostila_Scilab.pdf>. Acesso em: 19 maio 2015.
- Vídeo: <<https://www.youtube.com/watch?v=Tdx3xErGakw>>. Acesso em: 19 maio 2015.



Vocabulário

1. **Dízima periódica:** atribui-se esse nome ao número decimal ou fracionário que contém uma repetição periódica e infinita de algarismos.
2. **Semântica:** ação que resulta da execução do programa, porém, que apresenta um erro lógico. Diferentemente da sintaxe, que se não estiver correta o compilador identificará. Já um erro semântico, e por isso lógico, não é indicado na definição do comando por si, e sim, no resultado da ação proveniente desse comando.
3. **Compilador:** interpretador da linguagem de programação para a linguagem de máquina. Com isso é possível realizar o processamento das operações realizadas. É composto por processador de texto para a codificação; um depurador que executará comando a comando (e este ainda facilita a busca por erros e os indica quando houver) um manual ou help para possíveis apresentações sintáticas e semânticas; e um linker, que faz com que um programa utilize recursos ou ainda outros programas (EVARISTO, 2001).

Sem medo de errar

Considere a solicitação de verificação de *login* que foi realizada para a Think Now inserir no protótipo do aplicativo, sendo que esta deverá solicitar ao usuário a indicação do tipo de usuário: se é comerciante ou se é cliente, uma única vez ao realizar o cadastro. Observe a seguir uma proposta para este algoritmo:

algoritmo "Teste cadastro usuário"

// Função : algoritmo verifica se o usuário pretende logar com uma rede social

//em seguida, verifica se o usuário é comerciante

// Autor :

// Data : 25/05/2015

// Seção de Declarações

var

nome,email: caracter

logarRedeSocial, comerciante: inteiro

inicio

// Seção de Comandos

escreval ("**Cadastro Usuário Gastronomia e Hotéis do Litoral Sul**")

escreval ("Informe o seu nome:")

leia(nome)

escreval("Logar com uma rede social?")

escreval("Responda 1 para Sim ou 2 para Nao.")

leia(logarRedeSocial)

se (logarRedeSocial=1) entao

 escreval("Aguarde enquanto o aplicativo configura o seu cadastro!")

senao

 escreval("Informe o seu email!")

 leia(email)

fimse

escreval("Você é comerciante?")

 escreval("Responda 1 para Sim ou 2 para Nao.")

 leia(comerciante)

 se (comerciante=1)entao

 escreval("Prezado Parceiro de Negócios, finalize o seu cadastro!")

 fimse

escreval("Cadastro realizado com sucesso!")

fimalgoritmo

Avançando na prática

Pratique mais!	
Instrução Desafiamos você a praticar o que aprendeu transferindo seus conhecimentos para novas situações que pode encontrar no ambiente de trabalho. Realize as atividades e depois compare-as com as de seus colegas e com o gabarito disponibilizado no apêndice do livro.	
Estrutura condicional simples	
1. Competência de fundamentos de área.	Conhecer os princípios e conceitos que envolvem o aprendizado em construção de algoritmos e programação e a sua importância para o universo do desenvolvimento de sistemas.
2. Objetivos de aprendizagem	<ul style="list-style-type: none"> • Conhecer os conceitos e aplicações das estruturas de decisão "SE". • Conhecer como fazer uma estrutura condicional simples. • Conhecer e aplicar as estruturas condicionais compostas. • Conhecer e aplicar estruturas condicionais encadeadas.
3. Conteúdos relacionados	<ul style="list-style-type: none"> • Instruções primitivas: entrada de dados, atribuição e saída • Estrutura condicional simples
4. Descrição da SP	Para treinar os seus conhecimentos, propomos que você faça um algoritmo para resolver uma equação de segundo grau.
5. Resolução da SP	<pre> /*Algoritmo Cálculo de raízes da equação de 2º grau. var A, B, C: real //indica a declaração das variáveis e o seu tipo de dado específico delta, x1, x2: real //indica a declaração das variáveis e o seu tipo de dado específico INICIO //indica o início do algoritmo Leia(A,B,C) //significa que o programa precisa ler estas variáveis, ou seja, verificar os seus valores delta ← B*B - 4.*A*C //expressão linear que indica o cálculo do "delta" componente da fórmula Se (delta >= 0) então // estrutura de seleção que atribui uma condição para a realização das operações x1 ← (-B + sqrt(delta))/(2.*A) //expressão linear com o uso da função sqrt que calcula a raiz quadrada para a raiz 1 x2 ← (-B - sqrt(delta))/(2.*A) //expressão linear com o uso da função sqrt que calcula a raiz quadrada para a raiz 2 Escreva('As raízes reais são:', X1, X2) //comando que exibe o resultado do cálculo das raízes. senao Escreva('Nao existem raízes reais') //comando que será executado caso a condição predeterminada não seja atendida FIM //indica o encerramento das execuções do programa </pre>



Lembre-se

Bibliotecas do C:

stdio.h: é responsável pelos comandos de entrada e saída de dados.

math.h: é responsável pelo acesso aos comandos para a declaração de funções matemáticas.

Stdlib.h: é responsável por converter os números, alocar ou realocar memória, entre outras funções.



Faça você mesmo

Implemente o algoritmo anterior usando a linguagem C. Siga o exemplo:

```
#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <math.h>

int main(void){

    // declaracao de variáveis

    float A, B, C, delta, x1,x2;

    printf("Digite com os coeficientes da equação\n"); // leitura dos dados

    scanf("%f %f %f",&A,&B,&C); // %d representa o tipo de dado inteiro; %f
    representa o tipo de dado real; %c representa o tipo de dado char; %s
    representa o tipo de dado string

    delta = B*B - 4.*A*C; // cálculo do delta

    // Teste do discriminante

    if (delta >= 0) {

        // calculo das raizes

        x1 = (-B + sqrt(delta))/(2.*A);

        x2 = (-B - sqrt(delta))/(2.*A);
```

```
printf("A equacao dada e: %f X^2 + %f X + %f\n", A,B,C);

printf("As raizes reais sao: raiz 1= %f 2a. raiz2= %f\n", x1, x2); }

else { printf("Nao existem raizes reais"); } getch(); } // fim do programa
```

Faça valer a pena!

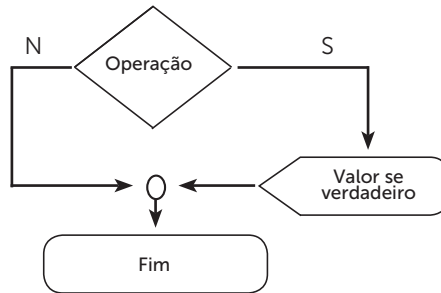
1. Dada a estrutura de seleção abaixo, assinale a alternativa que melhor descreve o processo:

```
se nivel = 1 entao
    escreval("O professor ganha"horas*12)
senao
    se nivel = 2 entao
        escreval("O professor ganha"horas*17)
    senao
        escreva("O professor ganha"horas*25)
fimse
fimse
finalgoritmo
```

(Disponível em: <http://partilho.com.br/visualg/exercicios-visualg/visualg-lista-de-exercicios/#Exercicio_6>. Acesso em: 27 maio 2015).

- Os comandos condicionais acima representam uma sucessão de testes com o uso de estruturas de seleção simples.
- Os comandos indicam que há primeiro uma estrutura de seleção que verifica se o nível indicado é o 1, em seguida, através de uma composição com o senão, há o teste em uma condição encadeada que verifica se o dado inserido equivale ao nível 2 e, caso não seja, implica em aplicar uma terceira regra, finalizando o processo.
- Os comandos de repetição acima representam uma sucessão de testes com o uso de estruturas de seleção simples.
- Os comandos de decisão e repetição inseridos verificam e atribuem ao salário a somatória dos indicadores apresentados em cada uma das condições.
- Os comandos se, então e senão, neste caso, representam uma agregação que poderá ser representada em fluxograma, e não na implementação.

2. O diagrama de blocos a seguir indica, respectivamente:



- condição, resultado e encerramento do processo.
- repetição, condição e operação.
- condição, valor se verdadeiro e valor se falso.
- repetição, valor se verdadeiro e encerramento.
- repetição, valor se verdadeiro e operação.

3. Dada a sintaxe abaixo, é possível afirmar o que está descrito na alternativa:

```

if (sexo != 'F' && sexo != 'M')
{
    printf ("Dado inválido.");
}
  
```

- o uso das chaves é de extrema importância nesse caso, pois indica que o comando é referente à condição imposta logo acima pelo comando condicional e não pode ser suprimida.
- as chaves podem ser suprimidas independente da quantidade de argumentos que serão executados na estrutura de decisão.
- os operadores relacionais e lógicos são aplicados de forma errônea e interferem no resultado da operação.
- o uso das chaves pode ser suprimido apenas quando há um argumento que deverá ser executado após a verificação do comando.
- a sintaxe do comando descrito acima está errada.

4. Para o comando abaixo, descrito em linguagem C, a palavra que completa o raciocínio lógico aplicado com o comando condicional é: Assinale a alternativa equivalente: `if (delta >= 0)`.

- a. `senão`
- b. `faça`
- c. `else`
- d. `then`
- e. `if`

5. O que representam as palavras `se`, `então` e `senão` para as estruturas de decisão ou seleção?

6. Explique as principais funções das bibliotecas:

- a. `stdio.h`:
- b. `math.h`:
- c. `stdlib.h`:

7. Analise a condição abaixo quanto à sintaxe e assinale a alternativa correspondente:

```
if (y > x) {  
    maior = y;  
}
```

- a. a condição apresentada verifica, através do uso do operador relacional maior "`>`", se o valor de "`y`" é maior que o valor da variável "`x`".
- b. o maior valor é armazenado na própria variável "`y`".
- c. a condição está semanticamente incorreta.
- d. a sintaxe está incorreta.
- e. o valor de `x` será sempre o maior valor de acordo com a condição inserida.

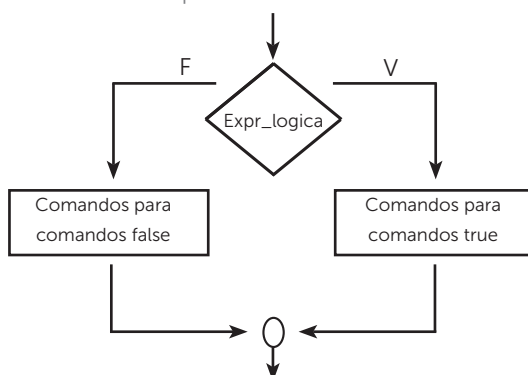
Seção 2.3

Estrutura condicional composta

Diálogo aberto

Olá, aluno. Primeiramente vamos recordar a estrutura de decisão composta: se-então-senão. Observe abaixo a estrutura que estudaremos: será com esse modelo lógico que os exercícios e soluções serão desenvolvidos.

Figura 2.4 | Estrutura de decisão composta



Fonte: Adaptado de Souza (2013, p. 128).

Vamos recapitular as entregas realizadas até o momento para o protótipo do aplicativo de divulgação dos locais de gastronomia e hotelaria do Litoral Sul: primeiro entregamos identificação das ações do aplicativo; depois um algoritmo para contabilizar os acessos; em seguida, a entrega foi de um fluxograma das ações do sistema, integrando o procedimento de *login* ao processo de contabilização de acessos; identificamos outros processos que poderão ser acoplados ao protótipo e a sua respectiva necessidade de declaração de variáveis. Estamos agora na segunda etapa e as entregas realizadas foram: um algoritmo de medição de índice de satisfação do cliente e, também, um algoritmo que verifica no momento do cadastro se o usuário é um comerciante ou não, para que ele seja direcionado à sua respectiva atividade no sistema. Agora, você precisará implementar a seguinte verificação quando o usuário for identificado como comerciante:

- a. O usuário deverá informar alguns dados de cadastro para se manter como cooperado do sistema. Então, ele deverá informar, além dos dados de cadastro advindos da rede social, caso tenha optado por esta forma de *login*, o seu endereço comercial: Rua, número, bairro, cidade, Estado e CEP, para envio de boleto de contribuição mensal para o desenvolvimento e manutenção do *site* e aplicativo.
- b. Se o usuário escolher a opção gastronomia, ele pagará o valor de R\$ 30,00 mensais para manter o vínculo com o *site*. Senão, ele será do setor de hotelaria, e deverá pagar o valor de R\$ 20,00 para manter o vínculo com o *site*. Bons estudos!

Não pode faltar

As estruturas de decisão compostas, como o próprio nome diz, propõem as verificações de condições e a execução dos comandos pertinentes a ela. Isso significa que para uma verificação de condição que resulte em verdadeira, serão executados comandos pertinentes a essa resposta, senão, os comandos que serão executados pertencem ao processo referente ao resultado que seja negativo ou falso, do ponto de vista do teste realizado.



Refleta

*[...] representa alguma expressão lógica, que se resultar **true**, vai permitir a execução de um conjunto de um ou mais comandos quaisquer existentes no caminho true, os quais podem ser sequenciais, de decisão ou de repetição. [...] se o resultado for **false**, será executado um conjunto contendo um ou mais comandos quaisquer, existentes no caminho false, podendo, novamente, ter estruturas sequenciais, de decisão ou de repetição. Ambos os fluxos convergem para o final da estrutura* (SOUZA, 2013, p. 128).

É sabido, portanto, que em uma estrutura de decisão composta o sistema deverá apresentar uma resposta ou outra apenas. Recorda-se do exemplo de algoritmo para cálculo da média de alunos? A condição imediatamente expressa no comando “se” ou “if”. Vejamos, por exemplo, uma verificação da nota. Se esta for maior que 7 (sete), o aluno deverá visualizar a mensagem de que está “Aprovado” na disciplina, senão, caso sua nota seja inferior ao valor estabelecido na condição, a mensagem exibida deverá ser “Reprovado”. Então, seguindo este raciocínio de apresentação de uma resposta caso a condição seja verdadeira ou falsa, é possível aplicar tal estrutura para diversas situações em que há a necessidade de verificação de uma regra. Observe no exemplo como resolver esta simples verificação:



Exemplificando

Algoritmo "Cálculo da média"

// Função: Algoritmo calcula a média ponderada a partir da apresentação das notas e exibe mensagem de aprovação ou reprovação"

// média (ponderada), informar o nome e sua menção aprovado (media ≥ 7), senão, aluno reprovado.

// Autor : JJJ

// Data : 28/05/2015

// Seção de Declarações

var

nomeAluno: caractere

n1, n2, n3, media: real

inicio

// Seção de Comandos

escreval("***Cálculo médias***")

escreva("Digite a primeira nota: ")

leia(n1)

escreva("Digite a segunda nota: ")

leia(n2)

escreva("Digite a terceira nota: ")

leia(n3)

media $\leftarrow (n1 \cdot 4 + n2 \cdot 2 + n3 \cdot 4) / 10$

se media ≥ 7 entao

 escreval("Aluno aprovado!")

senao

 escreval("Aluno reprovado!")

fimse

fimalgoritmo



Faça você mesmo

Agora você mesmo já pode implementar este exemplo de cálculo da média no Dev C++. Tente com a média ponderada e também a aritmética. Bom trabalho!

```
/*Programa para verificar a aprovação de um aluno*/

#include <stdio.h>

main() {

float Bim1, Bim2, Bim3, Bim4, MedBim, PrFinal, MedFinal;

printf("Digite as quatro notas bimestrais:");

scanf("%f%f%f%f", &bim1, &Bim2, &Bim3, &Bim4);

MedBim= (Bim1+Bim2+Bim3+Bim4)/4;

MedFinal=MedBim;

if((MedBim<7) && (MedBim>=5))

{

    printf("Digite a nota da prova final");

    scanf("%f", &PrFinal);

    MedFinal=(MedBim*6 + PrFinal*4)/10;

}

if (MedFinal>5.5)

    printf("Aluno aprovado com media final %.2f \n", MedFinal);

else

    printf("Aluno reprovado com media final %.2f \n", MedFinal);

}
```

Fonte: Evaristo (2001, p. 48)



Pesquise mais

Verifique o procedimento indicado para desenvolver algoritmos no Scilab no *link* disponível em: <http://www.academia.edu/4760779/SCILAB_na_Engenharia>. Acesso em: 27 maio 2015. Aprenda ainda mais com essa ferramenta. Prepare-se para novos desafios!



Vocabulário

True: verdadeiro

False: falso

Vamos agora falar de algumas particularidades dos algoritmos anteriormente apresentados. Primeiramente no pseudocódigo, quando o autor especificou que o aluno estará aprovado apenas se obtiver nota " \geq " (maior ou igual) a 7 (sete), limitou e definiu através de operadores relacionais que todas as notas inferiores a sete fariam com que o resultado fosse negativo para a condição imposta. Lembre-se quando você estudou limites, funções e inequações? Pois bem, estas podem fazer cada vez mais parte de sua rotina de desenvolvimento de soluções, então, a sugestão é que você saiba cada vez mais aplicar os conteúdos matemáticos às regras de negócios existentes nos mais variados segmentos de mercado.

Outro fator interessante de se observar está no comando para calcular a média: $\text{media} \leftarrow (n1*4 + n2*2 + n3*4) / 10$. Observe que o cálculo obedecerá à representação da expressão e efetuará a operação de multiplicação, adição e divisão, respectivamente, em função da precedência matemática e prioridade de execução. Se a expressão não tivesse os parênteses, o compilador entenderia que apenas " $n3*4$ " seria dividido por 10, por exemplo. Então, não se esqueça de alguns fatores simples, porém, elementares no ato de desenvolvimento de softwares.

Agora, já no algoritmo em C, o que vamos destacar são os seguintes comandos:

- **`scanf("%f%f%f%f", &Bim1, &Bim2, &Bim3, &Bim4);`**

a. neste comando, os sinais de percentual (%) seguidos da letra f indicam ao compilador que este deverá atribuir valores do tipo de dado float, ou seja, que permitem valores decimais, às variáveis em destaque "**&Bim**". O símbolo "**&**" indica a variável em que será alocado o valor informado.

- `if((MedBim<7) && (MedBim>=5))`

a. Já o destaque deste comando condicional está para o uso do operador lógico “e”, ou como na sintaxe acima, “&&”, que indica que as duas condições precisam ser atendidas para que a ação de aprovar o aluno seja realizada, no caso, o valor verdadeiro.

- `printf("Aluno reprovado com media final %0.2f \n", MedFinal);`

a. A expressão %0,2f indica que os valores decimais serão apresentados com até no máximo duas casas após a vírgula.

Muito bem, agora que já compreendeu outros fatores que podem influenciar no processamento e resultado final do seu algoritmo, conheça o operador ternário. A lógica proposta por este operador ternário poderá ser utilizada não apenas em aplicações desenvolvidas em C.

Costuma-se utilizá-lo quando a verificação tiver que retornar apenas o valor se falso ou o valor se verdadeiro. Uma das vantagens é que precisa de apenas uma única linha de comando, ao contrário de ter de usar a estrutura de decisão *if/else* (*se-então-senão*), o que pode otimizar o processamento dessa operação.

Observe como a sintaxe do operador ternário pode auxiliar no caso de condições compostas como *se-então-senão*. Então temos:

NomeDaVariável = (a expressão lógica que deverá ser testada) ? O comando para valores verdadeiros : O comando para valores se a condição for falsa.

Veja o exemplo:

`Maximo = (a >= b) ? a : b;`

Compreenda que a variável de nome “Máximo” receberá o valor de a ou b após a realização do teste, logo, é possível inferir que a operação representa a seguinte ação:

- **Se** o valor da variável a for maior ou igual que o valor contido na variável b, então, o resultado será apresentar o valor de a, senão, o resultado será apresentar o valor de b.

Outro exemplo de uso do operador ternário pode ser aplicado para o cálculo da média, por exemplo:

`Media = (n1 >= 7) ? "aprovado" : "reprovado";`

Todas as situações que envolvem regras de negócios ou, ainda, várias verificações antes de efetuar a execução dos comandos subsequentes, podem

ser mais ou menos complexas. Portanto, suponha que você tenha de executar o cálculo da média, oferecer opções para escolha e ainda apresentar outro resultado para esta operação, por exemplo, se o objetivo do programa fosse a criação de vários cenários dentro da mesma situação-problema.



Assimile

O comando condicional pode variar no seu formato, pode ser composto com outros comandos e pode conter um comando condicional dentro de outro comando condicional (PIVA JUNIOR et. al, 2013, p. 154).

Você pode, com as estruturas de decisão compostas, implementar tais verificações ou opções.

Sem medo de errar

algoritmo "Verifica dados cadastro"

// Função : permite a inserção de dados cadastrais e verifica a existência de
//campos vazios.

// Autor: JJJ

// Data : 29/05/2015

// Seção de Declarações

var

nome, end:caractere

op: inteiro

inicio

// Seção de Comandos

escreval("Prezado Cooperado, informe os seus dados:")

escreval("Nome:")

leia(nome)

escreval("Endereço:")

leia(end)

escreval("Informe 1 para Gastronomia e 2 para Hotelaria:")

leia (op)

se (op = 1) entao

 escreval("Imprimir o boleto no valor de R\$30,00")

senao

 escreval("Imprimir o boleto no valor de R\$20,00")

fimse

fimalgoritmo



Atenção!

Você ainda poderá implementar uma verificação de erros para essa rotina! Insira outros testes, ou seja, outras estruturas de decisão!



Lembre-se

Outras funções podem ser usadas no Visualg:

Abs(expressão) - Retorna o valor absoluto de uma expressão do tipo inteiro ou real.

ArcCos(expressão) - Retorna o ângulo (em radianos).

ArcSen(expressão) - Retorna o ângulo (em radianos).

ArcTan(expressão) - Retorna o ângulo (em radianos).

Cos(expressão) - Retorna o cosseno do ângulo (em radianos).

CoTan(expressão) - Retorna a cotangente do ângulo (em radianos).

Exp(base, expoente) - Retorna o valor de base elevado a expoente, sendo

ambos expressões do tipo real.

GraupRad(expressão) - Retorna o valor em radianos correspondente ao valor em graus representado por expressão.

Int(expressão) - Retorna a parte inteira do valor representado por expressão.

Log(expressão) - Retorna o logaritmo na base 10 do valor representado por expressão.

LogN(expressão) - Retorna o logaritmo neperiano (base e) do valor representado por expressão.

Pi - Retorna o valor 3.141592.

Quad(expressão) - Retorna o quadrado do valor representado por expressão.

RadpGrau(expressão) - Retorna o valor em graus correspondente ao valor em radianos representado por expressão.

RaizQ(expressão) - Retorna a raiz quadrada do valor representado por expressão.

Rand - Retorna um número real gerado aleatoriamente, maior ou igual a zero e menor que um.

RandI(limite) - Retorna um número inteiro gerado aleatoriamente, maior ou igual a zero e menor que limite.

Sen(expressão) - Retorna o seno do ângulo (em radianos) representado por expressão.

Tan(expressão) - Retorna a tangente do ângulo (em radianos) representado por expressão.

Disponível em: <<http://www.apoioinformatica.inf.br/produtos/visualg/linguagem/item/30-as-funcoes-do-visualg-versao-2-0>>. Acesso em: 29 maio 2015.

Avançando na prática

Pratique mais!	
Instrução Desafiamos você a praticar o que aprendeu transferindo seus conhecimentos para novas situações que pode encontrar no ambiente de trabalho. Realize as atividades e depois compare-as com as de seus colegas e com o gabarito disponibilizado no apêndice do livro.	
Estrutura condicional composta	
1. Competência de fundamentos de área	Conhecer os princípios e conceitos que envolvem o aprendizado em construção de algoritmos e programação e a sua importância para o universo do desenvolvimento de sistemas.
2. Objetivos de aprendizagem	<ul style="list-style-type: none">• Conhecer os conceitos e aplicações das estruturas de decisão "SE".• Conhecer como fazer uma estrutura condicional simples.• Conhecer e aplicar as estruturas condicionais compostas.• Conhecer e aplicar estruturas condicionais encadeadas.
3. Conteúdos relacionados	<ul style="list-style-type: none">• Instruções primitivas: entrada de dados, atribuição e saída• Estrutura condicional simples
4. Descrição da SP	Desenvolva um programa em linguagem de programação C que identifique se estamos em um ano bissexto.
5. Resolução da SP:	<pre>/*Programa que verifica se um dado ano é bissexto */ #include <stdio.h> main() { int Ano; printf("Digite o ano"); scanf("%d", &Ano); if (Ano % 4 == 0) printf("%d e' bissexto %d \n", Ano); else printf("%d nao e' bissexto %d \n", Ano); } Fonte: Adaptado de Evaristo (2001, p. 41).</pre>



Lembre-se

Operações mais complexas poderão ser inseridas com o uso da estrutura de decisão encadeada. Continue os estudos e pratique!

Faça valer a pena!

1. Desenvolva um algoritmo na plataforma Dev C++ que ordene três números inseridos pelo usuário.

2. Assinale a alternativa em que há explicação correta para o comando `"scanf("%f%f%f%f", &Bim1, &Bim2, &Bim3, &Bim4);"`:

a. Os sinais de percentual (%) seguidos da letra f, indicam ao compilador que este deverá atribuir valores do tipo de dado float. O símbolo "&" indica a variável em que será alocado o valor informado.

b. Os sinais de percentual (%) seguidos da letra f indicam a variável em que será alocado o valor informado.

c. O símbolo "&" indica ao compilador que este deverá atribuir valores do tipo de dado float.

d. Indica que se o valor da variável Bim1 for maior ou igual que o valor contido na variável Bim2, então, o resultado será apresentar o valor de Bim1, senão, o resultado será apresentar o valor de Bim2.

e. Todas as situações que envolvem várias verificações antes de efetuar a execução dos comandos subsequentes utilizam o "%f" e o "&".

3. Leia as afirmações e assinale a alternativa que apresenta a sequência correta (Verdadeiro ou Falso):

I. O comando condicional pode variar no seu formato, pode ser composto com outros comandos e pode conter um comando condicional dentro de outro comando condicional.

II. A expressão `%0,2f` indica que os valores decimais serão apresentados em seu formato sem vírgulas.

III. O comando condicional não pode conter mais de um comando para execução.

a. V, V, V

b. F, F, F

c. F, V, F

d. V, F, F

e. V, V, F

4. Analise o comando "Maximo = (a >= b) ? a : b;" e assinale a alternativa que representa a sua estrutura:

- a. se-então
- b. operador ternário
- c. se-então-senão
- d. se-senão
- e. se-então-se

5. Descreva a lógica envolvida na estrutura do operador ternário:

6. O que representa "if ((Mes == 4) || (Mes == 6) || (Mes == 9) || (Mes == 11))"?

- a. a expressão usa o operador lógico "e".
- b. a expressão usa o operador lógico "ou".
- c. a expressão usa o operador lógico "nao".
- d. a expressão usa o operador lógico "xou".
- e. a estrutura de decisão simples.

7. O comando printf("O mes %d tem %d dias", Mes, NumDias); está:

- a. correto
- b. incorreto
- c. parcialmente correto
- d. incompleto
- e. semanticamente incorreto

Seção 2.4

Estrutura condicional sequencial e encadeada

Diálogo aberto

Olá, aluno!

Vamos avançar um pouco mais e conhecer as estruturas condicionais compostas sequenciais e encadeadas?

A princípio, você identificará as suas formas de aplicação e em que situações estas são mais apropriadas. Então, quando se pensa em um aplicativo ou em um software que precisará verificar algumas condições, estas opções são bastante comuns.

No entanto, você estudará também outras estruturas que oferecem maneiras de otimizar tanto a codificação quanto o processamento.

Mas antes de conhecer tais estruturas, é importante saber diferenciar e aplicar as estruturas de decisão em suas diversas formas. Nesta seção, vamos conceituá-las e aplicá-las sequencialmente e encadeadas.

Neste ponto do desenvolvimento você incrementará o cadastro de usuário, porém, considerando um processo na visão do administrador desse aplicativo. Você deverá implementar uma opção que permita incluir, alterar e excluir informações de cadastro. Para tal, você ainda pode utilizar o Visualg.

Além de outras ferramentas que podem ser utilizadas, no entanto, se quiser transcrever em alguma linguagem computacional, a preferência para este momento de aprendizagem ainda repousa sobre a linguagem C, no Dev C++.

Por esse motivo, as estruturas condicionais ou de decisão, sequenciais e encadeadas, serão importantes para que você consiga organizar tais informações e implementar as opções.

Nesse sentido, você precisa pensar na aplicação de tais estruturas e em como podem auxiliar nesse processo.

Refleta para a situação proposta, qual será a melhor maneira de estruturar esse

algoritmo de forma que o seu custo, ou seja, o tempo de processamento e a necessidade de consumo de recursos de máquina não sejam muito elevados.

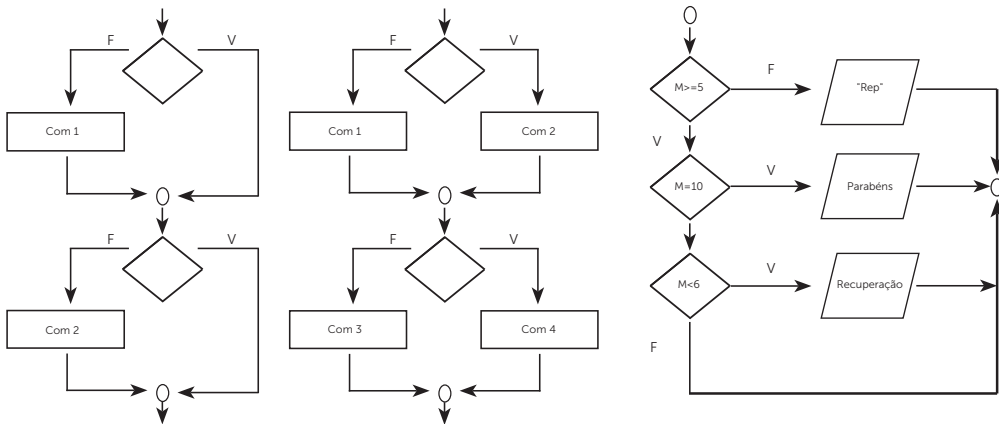
De acordo com estas novas implementações é que o seu estudo será orientado!

Desde já, bons estudos e práticas a você!

Não pode faltar

Você já aprendeu e aplicou o conceito da estrutura condicional simples e composta, porém, terá maior contato com a estrutura sequencialmente aplicada a partir deste estudo. O mesmo se aplica à estrutura encadeada. Então, quando ocorre o desvio sequencial ou encadeado em um algoritmo? É correto dizer que a estrutura sequencialmente aplicada é utilizada quando há uma estrutura condicional simples ou composta de forma que, após uma verificação, ou seja, uma estrutura condicional aplicada, vem logo em seguida outra aplicação de estrutura condicional, sendo, por essa razão, sequencial. Observe o exemplo no diagrama de bloco abaixo:

Figura 2.5 | Diagrama de bloco (fluxograma) estruturas condicionais simples e composta sequenciais



Fonte: Adaptado de Souza (2013, p. 64).

Fonte: Adaptado de Souza (2013, p. 64).

Fonte: Adaptado de Piva Junior et al. (2012, p. 175).

A Figura 2.5 mostra três tipos de operações que podem ser realizadas com as estruturas de decisão sequencial simples e compostas. A primeira imagem representa a verificação da condição e a resposta caso esta seja verdadeira (Com 1); se for falsa, na sequência, pode ser inserida outra verificação que indicará outros comandos, do grupo de comandos 2 (Com 2) a executar, caso a condição seja verdadeira, e assim, sucessivamente.

A segunda imagem traz também a estrutura de decisão sequencial composta. Observe que a resposta para a verificação, tanto se verdadeira quanto se falsa, pertencem à mesma condição de verificação na estrutura. Em seguida, há outra verificação sendo executada e as respectivas respostas à condição, tanto se verdadeira quanto se falsa.

Na terceira imagem, a estrutura condicional composta apresenta a sequência de execução de atividades exibindo a resposta caso a condição verificada seja falsa, e então, se a condição for verdadeira, dá-se sequência às demais verificações de condições impostas nas estruturas de decisão. Também nesse caso, todas convergem para o encerramento da execução conforme a indicação.



Assimile

- Estrutura de decisão sequencial simples:

se (condição 1) **entao**

comandos 1 para condição verdadeira

fimse

se (condição 2) **entao**

comandos 2 para condição verdadeira

fimse

- Estrutura de decisão sequencial composta:

se (condição 1) **entao**

comandos 1 para condição verdadeira

senao

comandos 2 para condição falsa

fimse

se (condição 2) entao

comandos 2 para condição verdadeira

senao

comandos 2 para condição falsa

fimse

Anteriormente foram apresentadas as estruturas de decisão sequencial simples e composta em pseudocódigo, evidenciando a forma como estas devem ser tratadas na codificação.



Refleta

As formas [...] apresentadas podem ser combinadas entre si, gerando outras possibilidades. Assim sendo, podem existir tomadas de decisão sequenciais com tomadas de decisão simples em conjunto com tomadas de decisão compostas. A codificação dessas estruturas segue as formas dos respectivos diagramas de blocos (MANZANO; OLIVEIRA, 2013, p. 66).

A fim de compreender melhor tais estruturas, veja no exemplo a seguir, mencionado pelos autores, a sua representação tanto em diagrama de blocos, quanto em pseudocódigo.



Exemplificando

Suponha que em uma cidade, a empresa que faz o controle e tratamento de água e esgoto precisa controlar o nível de água das caixas d'água dos bairros, para que os controladores possam identificar se a caixa d'água está cheia, vazia, transbordando ou sem vazão. Para tal, foram propostas as seguintes regras para verificação:

1. Verificar se o valor exibido na tela é igual a 1. Se sim, exibe a mensagem "Caixa d'água cheia." Senão, executar a condição que vem expressamente na sequência.

2. Verificar se o valor informado é igual a 2. Se sim, exibe a mensagem "Caixa d'água vazia." Senão, executar a condição que vem expressamente na sequência.
3. Verificar se o valor informado é igual a 3. Se sim, exibe a mensagem "Caixa d'água transbordando." Senão, executar a condição que vem expressamente na sequência.
4. Verificar se o valor informado é menor ou igual a 0. Se sim, exibe a mensagem "Caixa d'água sem vazão." Senão, executar a condição que vem expressamente na sequência.

Agora, no Visualg, faça o seguinte pseudocódigo, que auxilia na resolução do problema proposto no exemplo:



Faça você mesmo

```
//algoritmo decisão_sequencial
var
N: inteiro
inicio
    leia (N)
    se (N =1) entao
        escreval ("Caixa d'agua cheia.")
    fimse
    se (N =2) entao
        escreval ("Caixa d'agua vazia.")
    fimse
    se (N = 3) entao
        escreval ("Caixa d'agua transbordando.")
    fimse
    se (N <= 0) entao
```

escreval ("Caixa d'água sem vazão.")

fimse

fimalgoritmo

Fonte: Adaptado de Manzano (2013, p. 67).



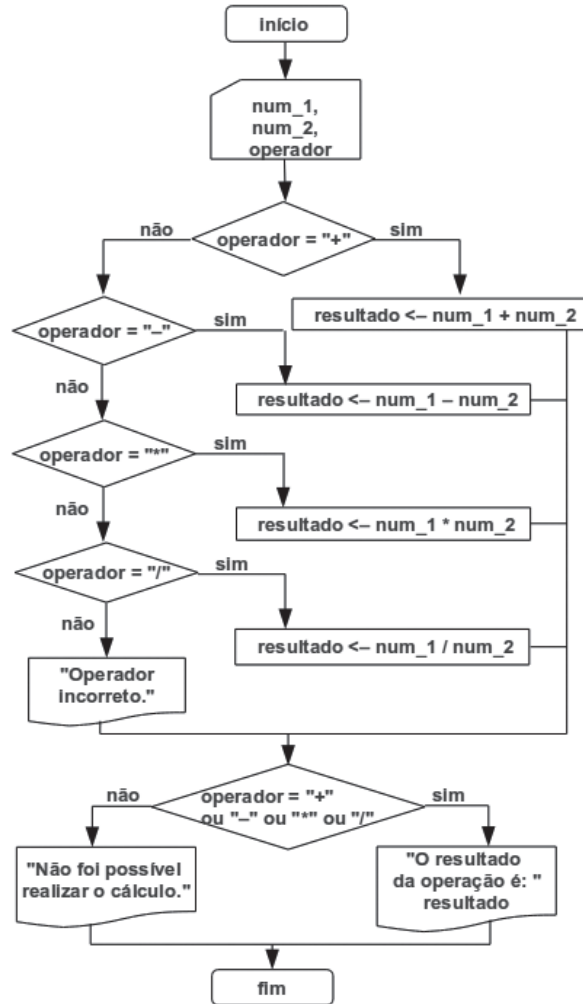
Pesquise mais

Material didático da Universidade Estadual de Maringá

Disponível em: <<http://www.din.uem.br/~teclopes/FCaula5.pdf>>.
Acesso em: 02 jun. 2015.

A partir de agora, o foco está nas estruturas de decisão encadeadas. Estas são bastante utilizadas quando se deseja inserir, dentre várias opções, respostas para cada uma das verificações, tanto para o caso de esta condição ser verdadeira, quanto para o caso de a condição resultar em falsa. Neste caso, a resposta para a condição quando esta é falsa é a inserção de um outro teste através da estrutura de decisão encadeada, sem a necessidade de finalizar a estrutura com "fimse" todas as vezes que se deseja inserir outra verificação. Observe a estrutura de decisão encadeada apresentada na figura sugerida como exemplo:

Figura 2.6 | Estrutura de decisão encadeada



Fonte: <http://www.activeinfo.com.br/curso_programacao/estruturas_de_decisao.html>. Acesso em: 03 jun. 2015.

Veja que a sequência de execução das instruções tem como consequência a outra condição ser verificada como resposta da estrutura anterior, tanto para o caso desta ser verdadeira quanto para o caso desta ser falsa.



Pesquise mais

Confira mais dicas de programação. Disponível em:

<<http://www.dicasdeprogramacao.com.br/estrutura-de-decisao-senao-senao/>>. Acesso em: 03 jun. 2015.

Os comandos condicionais encadeados podem ser substituídos em alguns casos por estruturas de seleção (CASO) que oferecem uma lista de opções ao usuário.

Sem medo de errar

```
algoritmo "Cadastro"
// Função :
// Autor : JJJ
// Data :
// Seção de Declarações
var
opcao,tipo:inteiro
nome, endereco:caracter
cpf,telefone:real
inicio
// Seção de Comandos
escreval("***Cadastro de cliente**")
escreval("1-Incluir")
escreval("2-Alterar")
escreval("3-Excluir")
leia(opcao)
se (opcao = 1) entao
```

```
escreval("Inclua o nome:")
leia(nome)
escreval("Inclua o endereço:")
leia(endereco)
escreval("Inclua o CPF:")
leia(cpf)
escreval("Inclua o Telefone:")
leia(telefone)
escreval("Nome:", nome)
escreval("Endereço:", endereco)
escreval("CPF:", cpf)
escreval("Telefone:", telefone)
senao
se (opcao = 2) entao
    escreval("**Selecione o tipo de alteração**")
    escreval("1- Alterar nome:")
    escreval("2- Alterar o endereco:")
    escreval("3- Alterar o CPF:")
    escreval("4- Alterar o telefone:")
    leia(tipo)
    se (tipo =1) entao
        escreval("Nome:")
        leia(nome)
    fimse
    se (tipo = 2) entao
        escreval("Endereço:")
```

```

    leia(endereco)

senao

    se (tipo = 3) entao
        escreval("CPF:")
        leia(cpf)
    senao
        se (tipo = 4)entao
            escreval("Telefone:")
            leia (telefone)
        fimse
    fimse

fimse

fimse

fimse

    escreval("Nome:", nome)
    escreval("Endereço:", endereco)
    escreval("CPF:", cpf)
    escreval("Telefone:", telefone)

fimse

se (opcao = 3) entao
    nome <- "-"
    endereco <- "-"
    cpf <- 000 000 000 00
    telefone <- 00 00000 0000
    escreval("Dados excluídos com sucesso!")
    escreval("Nome:", nome)
    escreval("Endereço:", endereco)

```

```
    escreval("CPF:", cpf)
    escreval("Telefone:", telefone)
fimse
finalgoritmo  escreval("Excluir Telefone - 4")
```

```
leia(tipo)
    se tipo=1 entao
        escreva("Exclua o Nome:")
    senao
        se tipo=2 entao
            escreva("Exclua o Endereço:")
        senao
            se tipo=3 entao
                escreva("Exclua o CPF:")
            senao
                se tipo=4 entao
                    escreva("Exclua o Telefone:")
                senao
                    se tipo=5 entao
                        escreva("Exclua a Data de Nascimento:")
                    fimse
                fimse
            fimse
        fimse
    fimse
```


fimse

fimse

fimse

fimse

fimse

fimse

fimse

fimse

fimse

fimse

fimse

fimse

fimse

fimalgoritmo



Atenção!

Para se criar registros e poder consultar os dados, logo adiante serão apresentadas as estruturas de vetores e matrizes.



Lembre-se

Você pode desenvolver um algoritmo ótimo! Realize você mesmo uma possibilidade de implementação com estruturas de decisão encadeadas e pratique!

Avançando na prática

Pratique mais!

Instrução

Desafiamos você a praticar o que aprendeu transferindo seus conhecimentos para novas situações que pode encontrar no ambiente de trabalho. Realize as atividades e depois compare-as com as de seus colegas e com o gabarito disponibilizado no apêndice do livro.

Estrutura condicional composta e encadeada

1. Competência de fundamentos de área	Conhecer os princípios e conceitos que envolvem o aprendizado em construção de algoritmos e programação e a sua importância para o universo do desenvolvimento de sistemas.
2. Objetivos de aprendizagem	<ul style="list-style-type: none"> • Conhecer os conceitos e aplicações das estruturas de decisão "SE". • Conhecer como fazer uma estrutura condicional simples. • Conhecer e aplicar as estruturas condicionais compostas. • Conhecer e aplicar estruturas condicionais encadeadas.
3. Conteúdos relacionados	<ul style="list-style-type: none"> • Estrutura condicional composta e encadeada
4. Descrição da SP	<p>Considerando o problema exemplificado anteriormente, que apresenta um algoritmo de controle de abastecimento de caixas d'água, desenvolva o seu diagrama de blocos (fluxograma):</p> <p>Retome na seção 2.4 de autoestudo, em Não Pode Faltar, o exemplo:</p> <p>"Suponha que em uma cidade a empresa que faz o controle e tratamento de água e esgoto precisa controlar o nível de água das caixas d'água dos bairros, para que os controladores possam identificar se a caixa d'água está cheia, vazia, transbordando ou sem vazão. [...]".</p>
5. Resolução da SP	<pre> graph TD Inicio([Início]) --> N1[N] N1 --> D1{N} D1 -- V --> Msg1[Msg] D1 -- F --> D2{N} D2 -- V --> Msg2[Msg] D2 -- F --> D3{N} D3 -- V --> Msg3[Msg] D3 -- F --> D4{N=3} D4 -- V --> C1(()) D4 -- F --> Msg4[Msg] C1 --> D5{N<=0} D5 -- V --> C2(()) D5 -- F --> Msg5[Msg] C2 --> Fim([Fim]) </pre> <p>Fonte: Manzano e Oliveira (2013, p. 67).</p>



Lembre-se

Após a indicação da variável de tipo inteiro N, é solicitada a leitura de um valor para ela. Assim que a leitura é realizada e o valor é fornecido para a variável N, ocorre uma de quatro possibilidades. [...] O programa apresenta para o usuário uma mensagem informando a ocorrência, não importa o valor fornecido. (MANZANO; OLIVEIRA, 2013, p. 67).

Faça valer a pena!

1. Quanto às estruturas de decisão sequenciais e compostas é correto afirmar o que está escrito na alternativa:

- a. estruturas de decisão sequenciais dependem exclusivamente de operadores relacionais.
- b. estruturas de decisão compostas podem ser encadeadas ou simples.
- c. estruturas de decisão não contemplam verificação de condições.
- d. estruturas de decisão sequenciais e encadeadas podem ser substituídas, em sua maioria, pela estrutura caso.
- e. estruturas de decisão encadeadas representam/são logicamente organizadas de forma igual às sequenciais.

2. Assinale a alternativa que apresenta o tipo de estrutura de decisão evidenciada abaixo:

```
se (N =1) entao
    escreval ("Caixa d'agua cheia.")
fimse
```

- a. estrutura condicional simples.
- b. estrutura condicional composta.
- c. estrutura condicional sequencial.
- d. estrutura condicional encadeada.
- e. estrutura de seleção.

3. Assinale a alternativa que apresenta o tipo de estrutura de decisão evidenciada abaixo:

```
se (condição 1) entao
    comandos 1 para condição verdadeira
senao
    comandos 2 para condição falsa
fimse
```

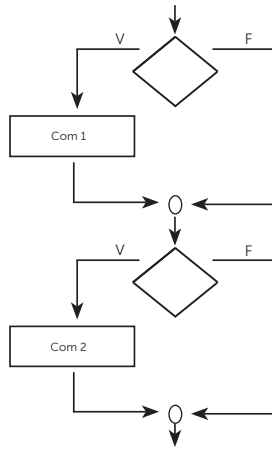
- a. estrutura condicional simples.
- b. estrutura condicional composta.
- c. estrutura condicional sequencial.
- d. estrutura condicional encadeada.
- e. estrutura de seleção.

4. Assinale a alternativa que apresenta o tipo de estrutura de decisão evidenciada abaixo:

```
se (tipo = 3) entao
    escreval("CPF:")
    leia(cpf)
senao
    se (tipo = 4)entao
        escreval("Telefone:")
        leia (telefone)
    fimse
fimse
```

- a. estrutura condicional simples.
- b. estrutura condicional composta.
- c. estrutura condicional sequencial.
- d. estrutura condicional encadeada.
- e. estrutura de seleção.

5. Analise a figura abaixo e assinale:



Refere-se a:

- a. estrutura condicional simples sequencial.
- b. estrutura condicional composta.
- c. estrutura condicional sequencial.
- d. estrutura condicional encadeada.
- e. estrutura de seleção.

6. Defina estruturas condicionais sequenciais.

7. Explique a lógica da estrutura condicional encadeada.