

Comandos condicionais de múltipla escolha (pseudocódigo)

Apresentação

Nesta Unidade de Aprendizagem, estudaremos o desenvolvimento de algoritmos com a utilização de estruturas condicionais de seleção múltipla em pseudocódigo, sua estrutura básica, a análise e o teste de aplicações práticas desenvolvidas.

Bons estudos.

Ao final desta Unidade de Aprendizagem, você deve apresentar os seguintes aprendizados:

- Analisar algoritmos com estruturas de seleção múltipla em pseudocódigo.
- Identificar problemas que podem ser resolvidos com a aplicação de comando de seleção múltipla.
- Desenvolver algoritmos em pseudocódigo que necessitam de comandos de seleção múltipla para a sua solução.

```
Algoritmo "folha"
Var
sal_minimo,sal_inicial, vl_hora, alimenta, sal_final: real
cod, horas: inteiro
categoria, turno: caractere
Inicio
Escreva("Digite o código do funcionário: ")
Leia(cod)
Escreva("Digite o número de horas trabalhadas no mês(inteiras): ")
Leia(horas)
Escreva("Digite o Turno (M-Manhã N-Noite V-Vespertino): ")
Leia(turno)
Escreva("Digite a categoria (G-Gerente ou F-Funcionária): ")
Leia(categoria)
Escreva("Digite o valor do salário mínimo (estadual): ")
Leia(sal_minimo)
Escolha(categoria)
caso "G"
    escolha(turno)
    caso "M"
        vl_hora <- sal_minimo * (4/100)
    caso "V"
        vl_hora <- sal_minimo * (4/100)
    caso "N"
        vl_hora <- sal_minimo * (6/100)
    fimsecolha
    fimsecolha
caso "F"
    escolha(turno)
    caso "M"
        vl_hora <- sal_minimo * (1/100)
    caso "V"
        vl_hora <- sal_minimo * (1/100)
    caso "N"
        vl_hora <- sal_minimo * (2/100)
    fimsecolha
    fimsecolha
sal_inicial <- vl_hora * horas
se (sal_inicial >= 0) e (sal_inicial <= 800) entao
    alimenta <- sal_inicial * (25/100)
senao
    se (sal_inicial > 800) e (sal_inicial <= 1200) entao
        alimenta <- sal_inicial * (20/100)
    senao
        se (sal_inicial > 1200) entao
            alimenta <- sal_inicial * (15/100)
        fimse
    fimse
sal_final <- sal_inicial + alimenta
Escreva("RELATORIO DO FUNCIONARIO") // impressão das saídas solicitadas
Escreva("Codigo do funcionario = ", cod)
Escreva("Número de horas trabalhadas = ", horas)
Escreva("Valor da hora trabalhada = ", vl_hora)
Escreva("Salário inicial = ", sal_inicial)
Escreva("Auxílio alimentação = ", alimenta)
Escreva("Salário final = ", sal_final)
FimAlgoritmo
```

Var

codigo, n_horas: inteiro
turno, categoria: caractere
valor: real

Inicio

Escreva("Código: ")
Leia(codigo)
Escreva("Nº de Horas trabalhadas: ")
Leia(n_horas)
Escreva("Turno: ")
Leia(turno)
Escreva("Categoria: ")
Leia(categoria)
Escreva("Valor hora trabalhada R\$: ")
Leia(valor)

Escreva("O funcionário código:",codigo, "
Categoria:",categoria, " do Turno", turno)
Escreva(" Trabalhou um total de ",n_horas,"
horas a um valor de R\$ ",valor, "a hora ")
Escreva(" Totalizando R\$ ", valor*n_horas)

Fimalgoritmo

Desafio

Sabendo que uma rede hoteleira deseja informatizar sua folha de pagamento, desenvolva em pseudocódigo uma solução conforme regras apresentadas pela empresa, de acordo com as especificações abaixo:

O funcionário possui os seguintes dados de entrada: código, número de horas trabalhadas no mês, turno de trabalho (M – matutino, V – vespertino ou N – noturno), categoria (F – funcionário, G – gerente).

Faça um algoritmo que:

- Leia as informações dos funcionários: código (inteiro), número de horas trabalhadas (inteiro) no mês, turno (caractere) e categoria (caractere). Não leia somente o valor da hora trabalhada (real), pois será calculada.
- Considere sempre a digitação de uma única letra para representar o turno de trabalho e a categoria do funcionário, conforme as tabelas:

Turno	Descrição do turno
M	Matutino
V	Vespertino
N	Noturno

Categoria	Descrição da categoria
G	Gerente
F	Funcionário

- Calcule o valor da hora trabalhada, conforme as regras apresentadas na tabela a seguir. O valor do salário mínimo deve ser solicitado pelo algoritmo, pois ele varia de estado para estado e a rede de hotéis está distribuída por todo o País. Utilizar o comando de seleção múltipla (ou um comando escolha e outro pode ser se encadeado – não utilizar se simples para esse item) para testar a categoria e o turno para calcular o valor da hora trabalhada.

CATEGORIA	TURNO	VALOR DA HORA TRABALHADA
G	N	6% do salário mínimo estadual
G	M OU V	4% do salário mínimo estadual
F	N	2% do salário mínimo estadual
F	M OU V	1% do salário mínimo estadual

- Calcule o salário inicial do funcionário com base no valor da hora trabalhada e no número de horas trabalhadas.

- Calcule o valor do auxílio-alimentação recebido pelo funcionário de acordo com seu salário inicial, conforme a tabela a seguir. Utilizar o comando de seleção composto encadeado.

SALÁRIO INICIAL	AUXÍLIO-ALIMENTAÇÃO
Até 800,00	25% do salário inicial
Acima de 800,00 até 1.200,00	20% do salário inicial
Acima de 1.200,00	15% do salário inicial

- Imprima como saída o código, número de horas trabalhadas, valor da hora trabalhada, salário inicial, auxílio alimentação e salário final (salário inicial + auxílio-alimentação).

a) Represente uma solução no VisuAlg para o problema descrito em forma de pseudocódigo para simular a folha da empresa. Entregue o arquivo .alg desenvolvido.

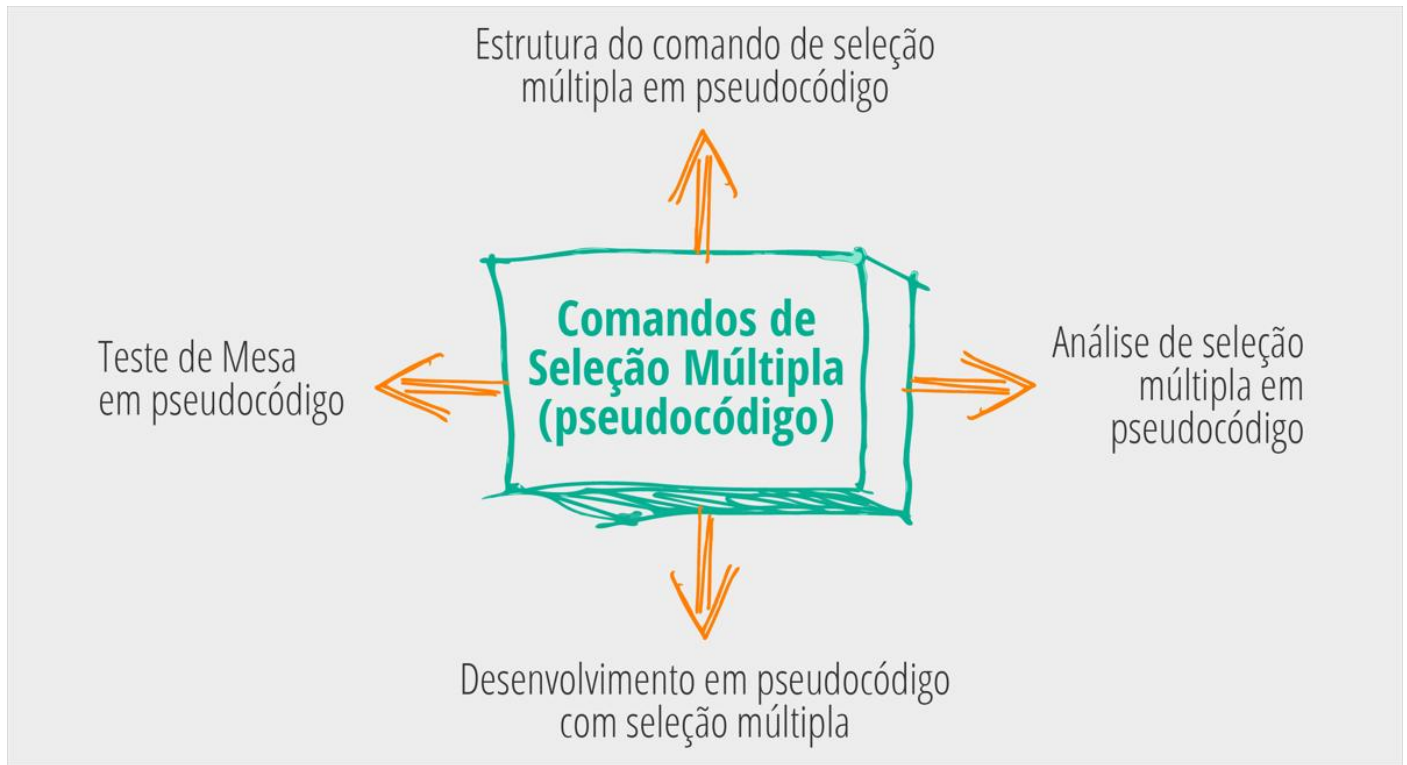
b) Realize o teste de mesa para as seguintes entradas:

Com salário mínimo de R\$ 780,00

Código	Horas	Turno	Categoria
1	100	M	G
2	10	N	G
3	30	V	G
4	20	M	F
5	120	N	F
6	120	V	F

Infográfico

A imagem retrata os diferentes recursos que utilizam o pseudocódigo.



Conteúdo do Livro

As estruturas de seleção múltipla são aplicadas em situações reais nas quais é necessário selecionar uma opção dentre várias disponíveis.

Assim, conforme o resultado de expressão lógica de um conteúdo da variável, é selecionado e executado um comando ou um conjunto de comandos. As opções do comando de seleção múltiplo são mutuamente exclusivas, ou seja, vai executar somente um das opções, nunca mais de uma.

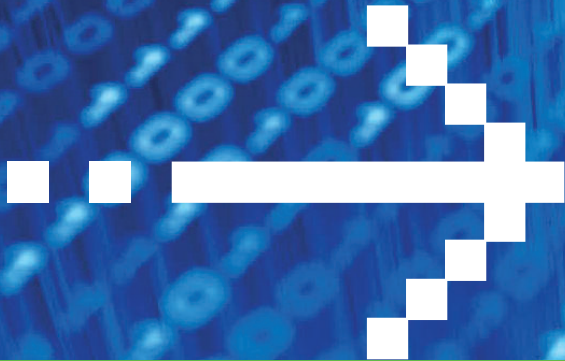
Para melhor compreender a estrutura de controle de seleção múltipla através de pseudocódigo, acompanhe um trecho da seguinte obra: EDELWEISS, N.; LIVI, M.A.C. *Algoritmos e programação com exemplos em Pascal e C* - Vol. 23. Série Livros Didáticos Informática UFRGS. Porto Alegre: Bookman, 2014.

O livro servirá como base para esta Unidade de Aprendizagem. No capítulo selecionado, serão apresentados a estrutura básica de funcionamento do comando de seleção múltipla e alguns exemplos de aplicações desenvolvidas.

Boa leitura.



■ ■ série livros didáticos informática ufrgs ■ ■



algoritmos e programação

com exemplos em Pascal e C

■ ■ nina edelweiss

■ ■ maria aparecida castro livi



→ as autoras

Nina Edelweiss é engenheira eletricista e doutora em Ciência da Computação pela Universidade Federal do Rio Grande do Sul. Durante muitos anos, lecionou em cursos de Engenharia e de Ciência da Computação na UFRGS, na UFSC e na PUCRS. Foi, ainda, orientadora do Programa de Pós-Graduação em Ciência da Computação da UFRGS. É coautora de três livros, tendo publicado diversos artigos em periódicos e em anais de congressos nacionais e internacionais. Participou de diversos projetos de pesquisa financiados por agências de fomento como CNPq e FAPERGS, desenvolvendo pesquisas nas áreas de bancos de dados e desenvolvimento de software.

Maria Aparecida Castro Livi é licenciada e bacharel em Letras, e mestre em Ciência da Computação pela Universidade Federal do Rio Grande do Sul. Desenvolveu sua carreira profissional na UFRGS, onde foi programadora e analista de sistema, antes de ingressar na carreira docente. Ministrou por vários anos a disciplina de Algoritmos e Programação para alunos dos cursos de Engenharia da Computação e Ciência da Computação. Sua área de interesse prioritário é o ensino de Linguagens de Programação, tanto de forma presencial quanto a distância.



E22a Edelweiss, Nina.
Algoritmos e programação com exemplos em Pascal e C
[recurso eletrônico] / Nina Edelweiss, Maria Aparecida Castro
Livi. – Dados eletrônicos. – Porto Alegre : Bookman, 2014.

Editado também como livro impresso em 2014.
ISBN 978-85-8260-190-7

1. Informática. 2. Algoritmos – Programação. I. Livi,
Maria Aparecida Castro. II. Título.

CDU 004.421

Catálogo na publicação: Ana Paula M. Magnus – CRB 10/2052

4.5

→ comando de seleção múltipla

O **comando de seleção múltipla** seleciona uma dentre diversas opções, com base na avaliação de uma expressão. Na pseudolinguagem aqui utilizada, a sintaxe deste comando é:

```

caso <expressão> seja
  <rótulo 1>: <comando 1>
  <rótulo 2>: <comando 2>
  ...
  <rótulo n>: <comando n>
  [ senão <comandos> ]
fim caso

```

onde os símbolos "[" e "]" significam que essa linha é opcional.

O comando inicia com um cabeçalho `caso <expressão> seja`, seguido de uma série de comandos rotulados, ou seja, comandos precedidos por um valor seguido do caractere ":". O resultado da avaliação da expressão utilizada no cabeçalho e os valores representados nos rótulos devem ser todos do mesmo tipo e corresponder a um valor ordinal como, por exemplo, inteiro ou caractere. Cada rótulo corresponde a somente um comando. Um comando composto pode ser utilizado caso se queira executar mais de uma ação para um determinado rótulo.

Depois de avaliada a expressão, seu resultado é comparado com cada um dos rótulos, na ordem em que são definidos. Somente o comando associado ao primeiro rótulo que for igual ao resultado da expressão é executado. Só a igualdade é verificada. Se o valor da expressão não for igual a qualquer dos rótulos, nada será executado pelo comando. Opcionalmente, poderá ser utilizada a cláusula `senão`, que indica o comando a ser executado caso nenhum dos rótulos corresponda ao valor da expressão do cabeçalho. O final do comando de seleção múltipla é indicado pelas palavras reservadas `fim caso`. Por exemplo, supondo que a variável `a` seja uma variável do tipo inteiro:

```

caso a seja
  1: a ← 0          {COMANDO SIMPLES PARA O CASO a = 1}
  2: início        {COMANDO COMPOSTO PARA O CASO a = 2}
    ler(a)
    escrever(a)
  fim
  3: a ← a + 1      {COMANDO SIMPLES PARA O CASO a = 3}
  senão escrever(a) {CLÁUSULA OPCIONAL PARA OUTROS VALORES DE a}
fim caso

```

Um comando de seleção múltipla equivale a um comando de seleção dupla com outros comandos de seleção dupla aninhados nele. O exemplo anterior equivale a:

```

se a = 1
então a ← 0                                {COMANDO SIMPLES PARA O CASO a = 1}
senão se a = 2
    então início                          {COMANDO COMPOSTO PARA O CASO a = 2}
        ler(a)
        escrever(a)
    fim
senão se a = 3
    então a ← a + 1  {COMANDO SIMPLES PARA O CASO a = 3}
    senão escrever(a) {CLÁUSULA OPCIONAL}

```

Uma vantagem de usar o comando de seleção múltipla em lugar desses comandos aninhados está na possibilidade de utilizar somente um nível de indentação, o que torna mais clara a visualização das alternativas existentes.

A Figura 4.5 representa o fluxograma correspondente ao exemplo anterior. Pode-se observar que o fluxo do programa somente passará por um dos comandos associados aos rótulos.

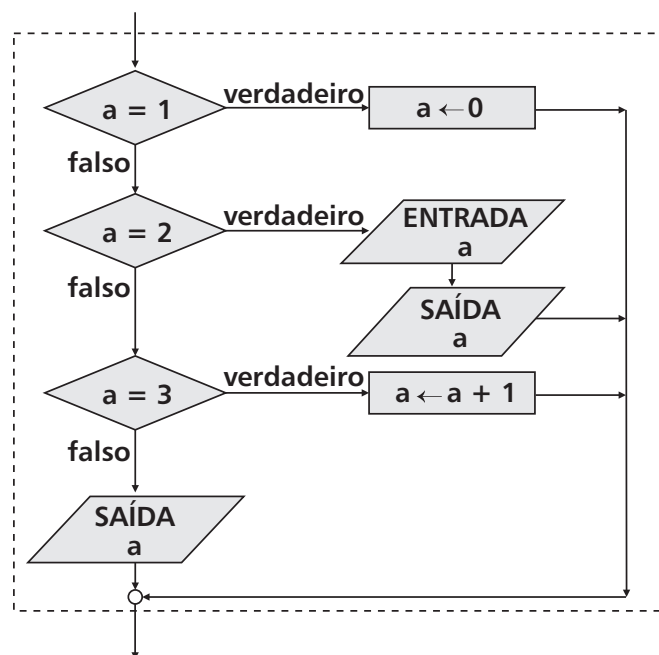


figura 4.5 Fluxograma de um comando de seleção múltipla.

O algoritmo a seguir, que calcula e informa a média e o conceito de um aluno, ilustra a utilização de rótulos do tipo caractere em um comando de seleção múltipla:

Algoritmo 4.5 - MédiaConceito2

```

{INFORMA MÉDIA E CONCEITO DE UM ALUNO}
Entradas: nota1, nota2, nota3 (real)
Saídas: média (real)
        Informação do conceito do aluno
Variável auxiliar: conceito (char)

```

```

início
  ler (nota1, nota2, nota3)                {ENTRADA DAS 3 NOTAS}
  média ← (nota1 + nota2 + nota3) / 3      {CALCULA MÉDIA}
  escrever (média)                        {INFORMA MÉDIA}
  se média ≥ 9                             {CÁLCULO DO CONCEITO}
    então conceito ← 'A'
  senão se média ≥ 7,5
    então conceito ← 'B'
  senão se média ≥ 6,0
    então conceito ← 'C'
  senão conceito ← 'D' {MÉDIA < 6}
  caso conceito seja                      {INFORMA CONCEITO}
    'A': escrever('Conceito A - Parabéns!')
    'B': escrever('Conceito B')
    'C': escrever('Conceito C')
    'D': escrever('Conceito D - Você foi reprovado')
  fim caso
fim

```

A implementação do comando de seleção múltipla varia, dependendo da linguagem de programação utilizada. Algumas linguagens permitem que um comando seja rotulado com uma lista de valores, ou mesmo com um intervalo. No exemplo a seguir, nota somente pode conter um valor inteiro:

```

caso nota seja
  0..5: escrever('Reprovado') {RÓTULO DO TIPO INTERVALO}
  6, 7, 8, 9, 10: escrever('Aprovado') {LISTA DE RÓTULOS}
fim caso

```

O primeiro comando é rotulado com o intervalo 0..5, representando os valores inteiros 0, 1, 2, 3, 4 e 5. O comando associado ao intervalo será executado quando o valor da variável nota for um desses valores. O segundo comando apresenta uma lista de rótulos. Se o valor da variável nota for igual a um deles, o segundo comando será executado. Observar que a utilização desses dois tipos de rótulos gerou um comando conciso e muito fácil de compreender.

4.6

→ exercícios de fixação

exercício 4.1 Ler um número inteiro. Se o número lido for positivo, escrever uma mensagem indicando se ele é par ou ímpar.

Algoritmo ParOuÍmpar

```

{INFORMA SE UM VALOR LIDO DO TECLADO É PAR OU ÍMPAR}
Entrada: valor (inteiro)                {VALOR A SER TESTADO}
Saída: Mensagem de 'par' ou 'ímpar'

```

```
Variável auxiliar: ehPar (lógica)
início
  ler(valor)                                {ENTRADA DO VALOR A SER TESTADO}
  se valor ≥ 0                               {VALOR LIDO É POSITIVO}
  então início
    ehPar ← (valor mod 2) = 0 {VERIFICA SE VALOR É PAR}
    se ehPar                               {VERDADEIRO SE ehPar FOR VERDADEIRO}
    então escrever('Par ! ')
    senão escrever('Ímpar ! ')
    fim
  fim
fim
```

exercício 4.2 Dados os coeficientes de uma equação do 2º grau, calcular e informar os valores de suas raízes.

Algoritmo EquaçãoSegundoGrau

```
{INFORMA OS VALORES DAS RAÍZES DE UMA EQUAÇÃO DO SEGUNDO GRAU}
Entradas: a, b, c (real)                {COEFICIENTES DA EQUAÇÃO}
Saídas: r1, r2 (real)                   {RAÍZES}
Variável auxiliar: disc (real)         {DISCRIMINANTE}
início
  ler(a, b, c) {ENTRADA DOS VALORES DOS COEFICIENTES DA EQUAÇÃO}
  se a = 0
  então início
    escrever('Não é equação do segundo grau! ')
    escrever('Raiz = ', (- c / b ))
    fim
  senão início
    disc ← sqr(b) - 4 * a * c           {CÁLCULO DO DISCRIMINANTE}
    se disc < 0
    então escrever('Raízes imaginárias !')
    senão início
      r1 ← ( - b + sqrt ( disc ) ) / ( 2 * a )
      r2 ← ( - b - sqrt ( disc ) ) / ( 2 * a )
      escrever('Raízes: ', r1, r2)
      fim
    fim
  fim
fim
```

exercício 4.3 Processar uma venda de livros em uma livraria. O cliente poderá comprar diversas unidades de um mesmo tipo de livro. O código que define o tipo do livro vendido (A, B, C) e o número de unidades desse livro são fornecidos como dados de entrada.

Preços: Tipo A – R\$ 10,00
Tipo B – R\$ 20,00
Tipo C – R\$ 30,00

Calcular e informar o preço a pagar. Caso tenham sido vendidos mais de 10 livros, exibir uma mensagem informando isso.

A solução deste problema é mostrada em duas etapas. Inicialmente, é apresentado um algoritmo em passos gerais para dar uma visão global da solução. Depois, cada um dos passos é detalhado, dando origem ao algoritmo completo.

Algoritmo UmaVenda - PASSOS GERAIS

{PROCESSA UMA VENDA DE LIVROS}

Entradas: tipo do livro ('A', 'B' ou 'C')

número de livros

Saídas: preço a pagar

mensagem indicando que foram vendidas mais de 10 unidades

1. Obter dados
2. Calcular preço de venda
3. Emitir mensagem caso necessário
4. Terminar

Detalhamento do algoritmo:

Algoritmo UmaVenda

{PROCESSA UMA VENDA DE LIVROS}

Entradas: código (caractere)

{CÓDIGO DO LIVRO}

numeroUnidades (inteiro)

{NR. UNIDADES VENDIDAS}

Saídas: aPagar (real)

{PREÇO A PAGAR}

{Mensagem indicando que foram vendidas mais de 10 unidades}

início

ler(código, numeroUnidades)

{ENTRADA DE DADOS}

se código = 'A'

{CÁLCULO DO PREÇO DA VENDA}

então aPagar ← numeroUnidades * 10

senão se código = 'B'

então aPagar ← numeroUnidades * 20

senão se código = 'C'

então aPagar ← numeroUnidades * 30

senão início {CÓDIGO ESTÁ INCORRETO}

aPagar ← 0

escrever('Código errado')

fim

se aPagar > 0

{CÓDIGO ERA VÁLIDO}

então início

escrever(aPagar)

{INFORMA VALOR A PAGAR}

se numeroUnidades > 10

então escrever ('Foram vendidas mais de 10 unidades')

fim

fim

exercício 4.4 Processar uma venda em um estabelecimento comercial. São fornecidos o código do produto vendido e seu preço. Deverá ser dado um desconto, de acordo com a seguinte tabela:

Código A – 20%

Código B – 10%

Produtos com código diferente de A ou B não terão desconto. Além disso, se o total a pagar for maior ou igual a R\$ 80,00, deverá ser dado um desconto adicional de 10%. Calcular e informar o preço a pagar e o desconto dado na compra, se for o caso.

Algoritmo UmaVendaComércio

```
{PROCESSA UMA VENDA EM UM ESTABELECIMENTO COMERCIAL}
  Entradas: preço (real)
            código (caractere)
  Saídas: desconto (real)
          aPagar (real)
início
  ler(código, preço)           {ENTRADA DE DADOS}
  aPagar ← preço                {INICIALIZA aPagar}
  desconto ← 0                  {INICIALIZA desconto}
  se código = 'A'               {CALCULA desconto}
  então desconto ← preço / 5
  senão se código = 'B'
  então desconto ← preço / 10
  aPagar ← aPagar - desconto    {CALCULA aPagar}
  se aPagar ≥ 80,00             {VERIFICA SE COMPRA ≥ 80,00}
  então início
    desconto ← desconto + aPagar / 10
    aPagar ← aPagar * 0,9      {MAIS 10% DE DESCONTO}
  fim
  escrever(aPagar)             {INFORMA VALOR A PAGAR}
  se desconto ≠ 0
  então escrever(desconto)     {INFORMA DESCONTO RECEBIDO}
fim
```

exercício 4.5 Escrever um programa que, dados um determinado mês (representado por um número inteiro) e um ano, informe quantos dias tem esse mês. Para determinar o número de dias de fevereiro, verificar primeiro se o ano é bissexto. Um ano será bissexto se terminar em 00 e for divisível por 400, ou se não terminar por 00, mas for divisível por 4.

Algoritmo DiasDoMês1

```
{INFORMA QUANTOS DIAS TEM UM DETERMINADO MÊS}
  Entrada: mês, ano (inteiro)  {DADOS DE ENTRADA - MÊS E ANO}
  Saída: dias (inteiro)        {NÚMERO DE DIAS DO MÊS NESTE ANO}
  Variável auxiliar: ehBissexto (lógica)
```



```

início
  ler(mês, ano)          {ENTRADA DE DADOS}
  se mês = 2              {SE FEVEREIRO, VERIFICA SE ANO EH BISSEXTO}
  início
    eh Bissexto ← falso
    se (ano mod 100) = 0
    então início
      se (ano mod 400) = 0
      então ehBissexto ← verdadeiro
      fim
    senão se (ano mod 4) = 0
    então ehBissexto ← verdadeiro
  fim
  se mês = 2              {FEVEREIRO}
  então se ehBissexto
    então dias ← 29
    senão dias ← 28
  senão se (mês=4) ou (mês=6) ou (mês=9) ou (mês=11)
    então dias ← 30      {ABRIL, JUNHO, SETEMBRO, NOVEMBRO}
    senão dias ← 31      {DEMAIS MESES}
  escrever(dias)         {INFORMA NÚMERO DE DIAS DO MÊS}
fim

```

Algoritmo DiasDoMês2

```

{INFORMA QUANTOS DIAS TEM UM DETERMINADO MÊS}
Entrada: mês, ano (inteiro)  {DADOS DE ENTRADA - MÊS E ANO}
Saída: dias (inteiro)        {NÚMERO DE DIAS DO MÊS NESTE ANO}
Variável auxiliar: ehBissexto (lógica)

início
  ler(mês, ano)              {ENTRADA DE DADOS}
  caso mês seja
    2 : {FEVEREIRO}
      eh_Bissexto ← falso
      se (ano mod 100) = 0
      então início
        se (ano mod 400) = 0
        então ehBissexto ← verdadeiro
        fim
      senão se (ano mod 4) = 0
      então ehBissexto ← verdadeiro
      se ehBissexto
        então dias ← 29      {ANO BISSEXTO}
        senão dias ← 28
      4, 6, 9, 11 : dias ← 30 {ABRIL, JUNHO, SETEMBRO, NOVEMBRO}

```

```
        senão dias ← 31                {DEMAIS MESES}
    fim caso                          {FIM COMANDO DE SELEÇÃO MÚLTIPLA}
        escrever(dias)                {INFORMA NÚMERO DE DIAS DO MÊS}
fim
```

exercício 4.6 Fazer um programa que leia um caractere e exiba uma mensagem informando se ele é uma letra, um dígito numérico ou um caractere de operação aritmética. Se o caractere não for de qualquer desses três tipos, informar que se trata de um caractere desconhecido.

Algoritmo IdentificarCaractere

```
{INFORMA TIPO DE CARACTERE LIDO}
    Entradas: lido (caractere)        {CARACTERE A SER IDENTIFICADO}
    Saídas: Mensagem indicando o tipo de caractere lido
início
    ler(lido)                        {CARACTERE A SER IDENTIFICADO}
    caso lido seja
        'A'..'Z', 'a'..'z' : escrever('Letra')
        '0'..'9': escrever('Dígito')
        '+', '-', '*', '/' : escrever('Operador aritmético')
        senão escrever('Caractere desconhecido')
    fim caso
fim
```

Sugestão: resolva este exercício sem utilizar o comando de seleção múltipla.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Dica do Professor

Em situações nas quais há um grande número de alternativas em um comando de seleção, sempre que possível, deve-se optar pelo comando de seleção múltipla, pois ele também seleciona uma dentre várias opções, com base na avaliação de uma expressão.

O comando de seleção múltipla é uma forma diferente de escrever o comando se-então-senão encadeado. O comando em pseudocódigo é chamado de comando escolha ou comando caso.

Assista ao vídeo para conhecer um pouco mais sobre essa estrutura e as expressões lógicas aplicadas, além de compreender e analisar algumas soluções práticas e o teste de mesa do comando de seleção múltipla em forma de pseudocódigo.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Exercícios

- 1) Observe o fluxograma com a aplicação do comando de seleção múltipla. , Algoritmo "Calculo_dia"

Var

dia : inteiro

valor: real

teste : logico

Inicio

Escreva("Digite o dia da semana (1 até 7): ")

Leia (dia)

teste <- falso

escolha(dia)

caso 1

valor<- 10.0

caso 2

valor<- 5.0

caso 3

valor<- 6.0

caso 4

valor<- 7.0

caso 5

valor<- 8.0

caso 6

valor<- 9.0

caso 7

valor<- 10.0

outrocaso

teste <- verdadeiro

fimescolha

se (teste = verdadeiro) entao

escreval(" Dia da semana inválido")

senao

Escreval(" Valor = ", valor)

fimse

FimAlgoritmo

Analise o algoritmo e selecione a alternativa correta.

- A) Se a variável dia receber o valor 10, imprimirá na tela a frase "Valor = 10".
- B) A estrutura de múltipla escolha não está estruturada corretamente.
- C)** A variável teste controla qual a mensagem que deverá ser exibida, de acordo com o que for digitado.
- D) A variável teste pode ser removida do algoritmo sem afetar nenhuma funcionalidade.
- E) Se a variável dia receber o valor 7, imprimirá na tela a frase "Valor = 5".
- 2) Observe o problema: ler a variável categoria e determinar a categoria do funcionário e qual o valor de imposto que o indivíduo pagará sobre seu salário, conforme tabela apresentada abaixo.

Categoria	Taxa do imposto
A	11%
B	9%
C	Isento

Caso seja informada uma categoria inválida (diferente de A, B e C) a frase a ser exibida deve ser "Categoria inexistente"; senão, deve ser impresso o valor do imposto calculado sobre o salário do funcionário.

Análise os comandos de seleção aplicados nos três algoritmos abaixo para a resolução do problema descrito.

Algoritmo 1	Algoritmo 2	Algoritmo 3
Algoritmo "Algoritmo1" Var sal, valor: real cat: caractere Inicio Leia (sal, cat) Escolha (cat) caso "A" valor <- sal * (11/100) caso "B" valor <- sal * (9/100) caso "C" valor <- 0 outrocaso valor <- 0 Escreval("Categoria invalida") fimescolha Escreval("Valo do imposto = ", valor) Escreval("Total salário = ", sal+valor) finalgoritmo	Algoritmo "Algoritmo2" Var sal, valor: real cat: caractere Inicio Leia (sal, cat) se (cat="A") entao valor <- sal * (11/100) senao (cat= "B") entao valor <- sal * (9/100) senao (cat= "C") entao valor <- 0 senao valor <- 0 Escreval("Categoria invalida") fimse fimse Escreval ("Valor do imposto =", valor) Escreval ("Total salário =", sal+valor) finalgoritmo	Algoritmo "Algoritmo3" Var sal, valor: real cat: caractere Inicio Leia (sal, cat) se (cat="A") entao valor <- sal * (11/100) fimse se (cat= "B") entao valor <- sal * (9/100) fimse se (cat= "C") entao valor <- 0 fimse se (cat <> "A") ou (cat<> "B") ou (cat<> "C") então valor <- 0 Escreval("Categoria inválida") fimse Escreval("Valor do imposto =", valor) Escreval "Total salário = ", sal+valor) fimdoalgoritmo

Avalie as alternativas apresentadas com relação aos algoritmos desenvolvidos e selecione a alternativa CORRETA.

- A) O algoritmo 2 é a pior solução desenvolvida.
- B) Os algoritmos 1 e 2 equivalem em número de testes se o valor de sal=100 e cat="C".
- C) Os algoritmos 2 e 3 equivalem em número de testes se o valor de sal=800 e cat= "B".
- D)** O algoritmo 1 é uma das melhores soluções desenvolvidas.
- E) Nenhum destes algoritmos representa corretamente uma solução para o problema apresentado.

3) Considere o seguinte problema:

O correntista de um banco precisa realizar um empréstimo para apenas 30 dias e buscou um banco para verificar as taxas e os juros. Para simular o financiamento, é necessário informar o estado civil, o valor pretendido de financiamento e a idade do correntista. Após, será calculado e impresso o valor final do financiamento que a pessoa deverá pagar após 30 dias. O valor a pagar do financiamento e a taxa de juro sofrem a influência do estado civil e da idade da pessoa e são calculados conforme a tabela abaixo:

Estado civil	Valor do seguro	Taxa de juro
Casado	0	Até 21 anos, 1,5% do valor Acima de 21 anos, 1%
Solteiro	1% do valor	Até 21 anos, 2,3% do valor Acima de 21 anos e até 35 anos, 4% do valor Acima de 35 anos, 2% do valor
Outros	0	1,5% do valor

O valor do seguro e a taxa de juro são calculados sobre o valor que se deseja financiar para 30 dias. Analise os comandos de seleção aplicados para a resolução do problema descrito. Selecione a alternativa que representa uma solução correta para o problema.

A) [Clique aqui](#)

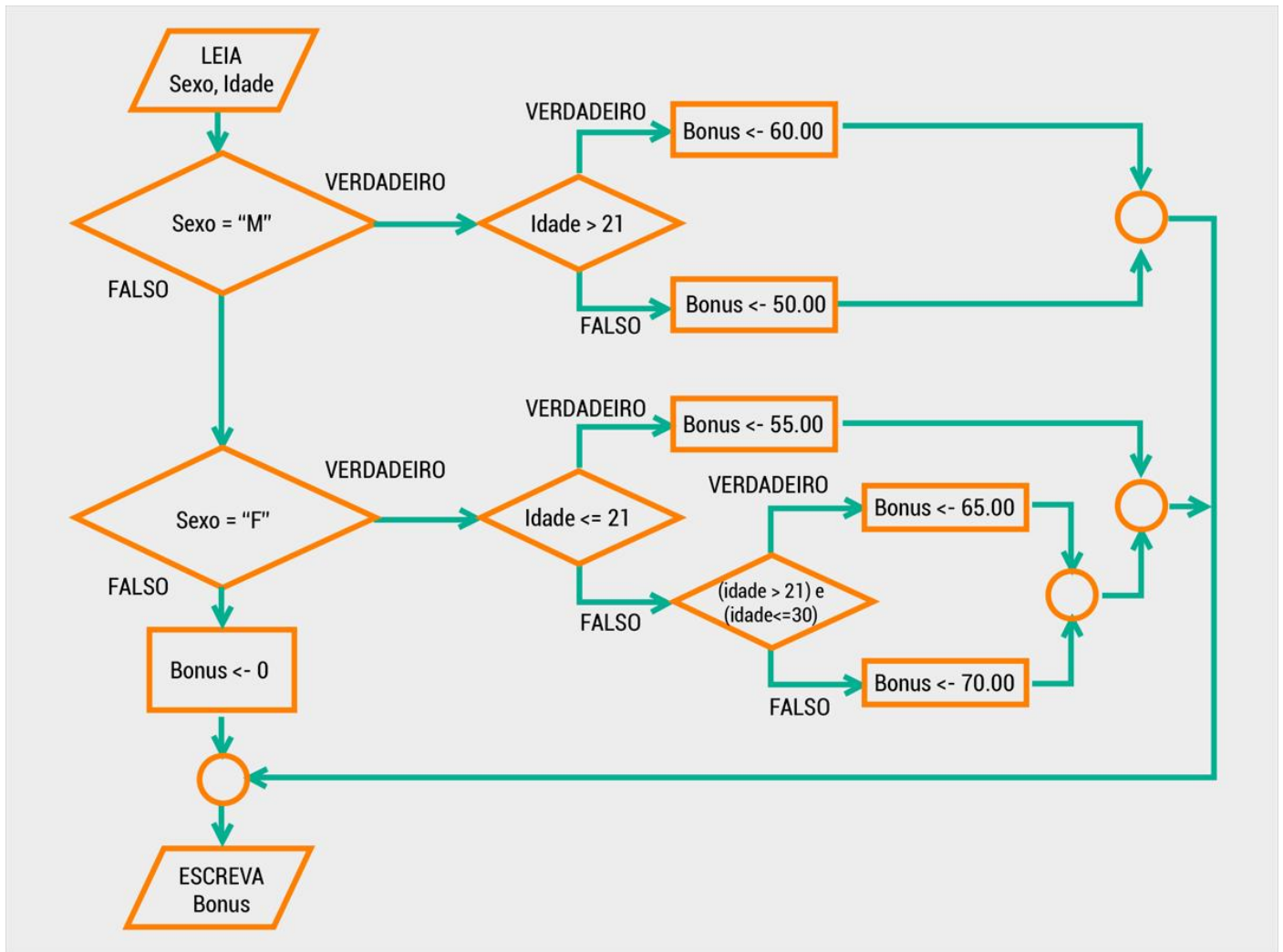
B) [Clique aqui](#)

C) [Clique aqui](#)

D) Nenhuma das alternativas.

E) [Clique aqui](#)

4) Observe o fluxograma:



Selecione a alternativa que representa de forma correta a solução do fluxograma em pseudocódigo.

- A) [Clique aqui](#)
- B) [Clique aqui](#)
- C) [Clique aqui](#)**
- D) [Clique aqui](#)

E) O fluxograma não pode ser transposto em uma solução em pseudocódigo.

5) No nosso cotidiano, precisamos tomar decisões com base em alternativas que nos são apresentadas. Em pseudocódigo, qual é o comando mutuamente exclusivo, ou seja, que executará somente uma das várias alternativas apresentadas e testadas? No momento que uma das condições é verdadeira, ele entra na condição e executa todos os comandos que estão dentro dessa condição; depois, vai para o final do comando, sem testar as outras

alternativas. É também uma forma mais simples e clara de escrever o comando de seleção composto.

Qual é o comando que representa essa funcionalidade?





- A) Se-então.
- B) Se-então-senão.
- C) Se-então-senão encadeado.
- D) Escolha.**
- E) Nenhuma das alternativas.

Na prática

PROMOÇÃO DE LIVRARIA

Uma livraria está realizando uma promoção no centro da cidade com algumas operações básicas de compra. O estabelecimento está fazendo uma liquidação de livros e organizou-os por código de cores, conforme a tabela mostra:



	COR	LETRA QUE IDENTIFICA A COR	DESCONTO
	Preto	P	20%
	Vermelho	V	30%
	Branco	B	40%
	Amarelo	A	50%

Sempre que precisamos selecionar uma dentre várias alternativas, o melhor é utilizar o comando de seleção múltipla.

Vamos desenvolver uma solução para o problema da livraria?

Algoritmo "Algoritmo1"

Var

valor, desconto, total, valor_desconto: real

cor: caractere

Início

Escreval("")

Escreval("Preto - P - 20%")

Escreval("Vermelho - V - 30%")

Escreval("Branco - B - 40%")

Escreval("Amarelo - A - 50%")

Escreva("Digite a cor do produto :")

Leia(cor)

Escreval("Digite o valor do produto:")

Leia (valor)

Escolha(cor)

caso "P"

desconto <- 20/100

caso "V"

desconto <- 30/100

caso "B"

desconto <- 40/100

caso "A"

desconto <- 50/100

outrocaso

desconto <- 0

fimescolha

valor_desconto <- valor * desconto

total <- valor - valor_desconto

Escreval("Valor do desconto = ", valor_desconto)

Escreval("Valor do produto = ", total)

fimalgoritmo



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Saiba mais

Para ampliar o seu conhecimento a respeito desse assunto, veja abaixo as sugestões do professor:

Lógica de Programação - Comando Caso - Escolhas múltiplas



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Escolha caso



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.