

**Questões 8, 20, 21, 22, 23, 24**

**8. Determine se cada uma das seguintes alternativas é verdadeira ou falsa. Se falsa, explique por quê.**

- a. O caso *default* é requerido na instrução de seleção *switch*.**
- b. A instrução *break* é requerida no último caso de uma instrução de seleção *switch*.**
- c. A expressão  $((x > y) \ \&\& \ (a < b))$  é verdadeira se  $(x > y)$  for verdadeiro ou  $(a < b)$  for verdadeira.**
- d. Uma expressão contendo o operador  $\parallel$  é verdadeira se um ou ambos de seus operandos forem verdadeiros.**
- e. Para testar para uma série de valores em uma instrução *switch*, pode-se utilizar um hífen ( – ) entre os valores inicial e final da série em um rótulo *case*.**
- f. Listar casos consecutivamente sem instruções entre eles permite aos casos executar o mesmo conjunto de instruções.**

**Resolução:**

- a. Falsa, o default não é obrigatório dentro da estrutura switch, o mesmo é utilizado como operação de parada, porém pode ser usado mesmo fora do switch.
- b. Falsa, o break é usado em todos os casos, pois ele representa uma interrupção de forma imediata, para assim sair do switch.
- c. Falsa, neste caso o  $\&\&$  respresenta o “e” onde as duas condições precisam ser verdadeiras para atender a condição.
- d. Verdadeira, o  $\parallel$  representa o “ou” logo se apenas uma condição precisar ser atendida.
- e. Falsa, a estrutura switch não oferece ferramentas para testar intervalos de valores, desta forma cada valor deve ser testado de forma separada.
- f. Verdadeiro.

**20. Escreva um modelo para representar uma lâmpada que está à venda em um supermercado.**

VendaLampada
- estadoLampada
+ comprarLampada()
+ devolverLampada()

```

modelo VendaLampada
início do modelo
  dado estadoLampada
  operação comprarLampada()
    início
      se(estadoLampada == perfeito)

        comprarLampada = verdadeiro

        imprime "Venda realizada com sucesso!"

      fim se

    fim

  operação devolverLampada()
    início

      se(estadoLampada == defeito)

        devolverLampada = verdadeiro

        imprime "Devolução realizada com sucesso!"

      fim se

    fim do modelo

```

**21. Escolha um esporte e um campeonato desse esporte. Crie um modelo para representar um time nesse campeonato.**

Futsal
- campeonatoFutsal
+ inscricaoTime()
+ eliminacaoTime()
+ vencedorTime()

```

modelo Futsal
início do modelo
  dado campeonatoFutsal
  operação inscricaoTime()
    início
      inscricaoTime = verdadeiro
      imprime "Inscrição realizada com sucesso!"
    fim

  operação eliminacaoTime()
    início
      se(eliminacaoTime == verdadeiro)
        imprime "Time desclassificado"

      fim se

    fim

  operação vencedorTime()
    início
      se(vencedorTime == verdadeiro)
        imprime "Time vencedor!!!"
      fim se
    fim

  fim do modelo

```

**22. Crie um modelo que represente os dados básicos de um livro de uma biblioteca, que pode ser emprestado a leitores.**

Livros
<ul style="list-style-type: none"> <li>- dadosLivros</li> <li>- nomeLivros</li> <li>- nomeLeitores</li> </ul>
<ul style="list-style-type: none"> <li>+ realizarEmprestimo(data, nome, tituloLivro)</li> <li>+ devolverLivro(data, nome, tituloLivro)</li> <li>+ estenderEmprestimo(data)</li> </ul>

```

modelo Livros
início do modelo
    dado dadosLivros
operação realizarEmprestimo(data, nome, tituloLivro)
    início
        dadosLivros = data
        nomeLivros = tituloLivro
        nomeLeitores = nome
        realizarEmprestimo = verdadeiro
        imprime "Emprestimo realizado com sucesso!"
    fim
operação devolverLivro(data,nome,tituloLivro)
    início
        dadosLivros = data
        nomeLivros = tituloLivro
        nomeLeitores = nome
        devolverLivro = verdadeiro
        imprime "Devolução realizada com sucesso!"
    fim
operação estenderEmprestimo(data)
    início
        dadosLivros = data
        estenderEmprestimo = verdadeiro
        imprime "Emprestimo renovado com sucesso!"
    fim
fim
fim do modelo

```

**23. Crie um modelo que represente uma equação de segundo grau.**

EquacaoSegundograu
<ul style="list-style-type: none"> <li>- delta</li> <li>- idenA</li> <li>- idenB</li> <li>- idenC</li> </ul>
<ul style="list-style-type: none"> <li>+ calcularDelta(a,b,c)</li> <li>+ dadosAbc(a,b,c)</li> <li>+ calcularBhaskara</li> </ul>

```

modelo EquacaoSegundograu
início do modelo
    dado hospede
operação dadosAbc(a,b,c)
    início
        idenA = a
        idenB = b
        idenC = c
    fim
operação calcularDelta(a,b,c)
    início
        delta = b^2 - 4 * a * c
        imprime "Valor de delta: "
    fim
operação calcularBhaskara(delta,x1,x2,a,b)
    início
        calcularBhaskara = (x1,2) = ((-b) ± sqrt (delta))/2*a
        calcularBhaskara = (x1) = (+ b + sqrt(delta))/2*a
        calcularBhaskara = (x2) = (- b + sqrt(delta))/2*a
        imprime "Valor de X1: "
        imprime "Valor de X2: "
    fim
fim
fim do modelo

```

**24. Crie um modelo que represente uma reserva em uma pousada.**

Pousada
<ul style="list-style-type: none"><li>- hospede</li><li>- quartosHotel</li></ul>
<ul style="list-style-type: none"><li>+ realizarReserva(data, nome,nQuarto, checkin)</li><li>+ cancelarReserva(nome,nQuarto)</li><li>+ finalizarReserva(nome,nQuarto,checkout)</li></ul>

modelo **Pousada**

início do modelo

**dado hospede**

**operação realizarReserva**(data, nome, nquarto,checkin)

início

hospede = nome

quartosHotel = nquarto

checkin = verdadeiro

imprime "Reserva realizada com sucesso!"

fim

**operação cancelarReserva**(nome,nQuarto)

início

hospede = nome

quartosHotel = nQuarto

cancelarReserva = verdadeiro

imprime "Cancelamento realizado com sucesso!"

fim

**operação finalizarReserva**((data, nome, nquarto, checkout)

hospede = nome

quartosHotel = nquarto

checkout = verdadeiro

imprime "Obrigado e volte sempre! "

fim

fim do modelo