algoritmos e programação

com exemplos em Pascal e C

nina edelweiss
maria aparecida castro livi

bookman

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

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

representação do comando `para/faça` em fluxogramas.

O comando `para/faça` é representado em fluxogramas por um bloco especial, compreendendo um losango com um retângulo acoplado, onde são mostrados a variável de controle, seu valor inicial, seu incremento ou decremento e o teste que determina o término de sua execução. A partir desse bloco, o fluxo do programa pode seguir dois caminhos: pela linha vertical, que leva ao(s) comando(s) a ser(em) repetido(s); ou pela horizontal, que leva para o comando seguinte quando terminarem as repetições, conforme mostrado na Figura 5.2. Lembrar que, se o laço de repetição contiver mais de um comando, eles deverão ser agrupados em um comando composto. A Figura 5.3 mostra o fluxograma do exemplo anterior, em que a leitura e a escrita de uma variável são repetidas 10 vezes.

valores inicial, final e incremento definidos por expressões. Os valores inicial, final e do incremento podem ser definidos explicitamente ou através do resultado de uma expressão. Caso seja utilizada uma expressão, seu resultado deve ser do mesmo tipo da variável de controle. As expressões contidas no cabeçalho de um comando `para/faça` são avaliadas somente uma vez, no início da sua execução. Por exemplo, considerando as variáveis inteiras `i`, `var1` e `var2`, o cabeçalho:

```
para i de var1 incr 1 até (7 + var2) faça
```

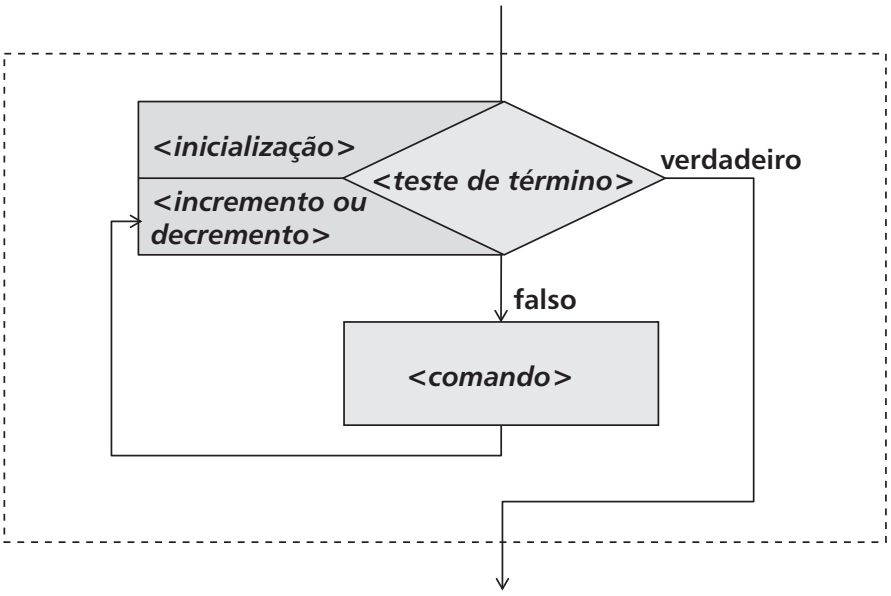

figura 5.2 Fluxograma do comando `para/faça`.

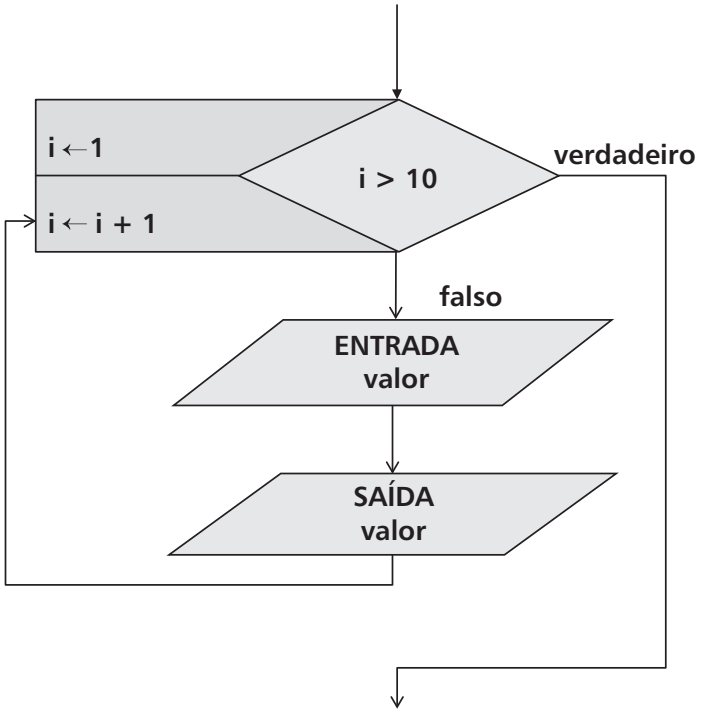

figura 5.3 Exemplo de fluxograma com comando `para/faça`.

indica que a variável de controle `i` é inicializada com o valor contido na variável `var1` e que o valor final é o resultado da expressão `(7 + var2)`, avaliada no início do comando. Mesmo que o valor de `var2` seja alterado durante a execução do comando `para/faça`, o valor calculado para o limite final não será alterado. As expressões contidas no cabeçalho podem também apresentar variáveis iguais, como no exemplo a seguir:

```
para i de var1 até (var1 + 10) faça
```

O valor do incremento deve ser inteiro e diferente de zero, podendo ser positivo ou negativo. Quando o incremento for positivo, o valor inicial da variável de controle deve ser menor ou igual ao valor final; quando for negativo, o valor inicial deve ser maior ou igual ao valor final para que o comando seja executado pelo menos uma vez. Caso o valor do incremento seja 1, sua definição pode ser omitida. Exemplos:

```
para i de 1 incr 2 até 10 faça  {INCREMENTO POSITIVO}
   escrever (i)
para i de 7 incr -1 até 3 faça  {INCREMENTO NEGATIVO}
   escrever (i)
para i de 1 até 10 faça         {INCREMENTO +1 FICA IMPLÍCITO}
   escrever (i)
```

utilização da variável de controle dentro do laço. Como mostrado nos últimos exemplos, a variável de controle pode ser utilizada nos comandos dentro do laço de repetição. Embora seja permitido, não é aconselhável alterar o valor da variável de controle dentro do

laço, pois isso poderia alterar a quantidade de repetições a serem realizadas. A alteração explícita do conteúdo da variável de controle, através de uma atribuição incluída dentro do laço, faz que, em cada repetição, a variável de controle seja alterada duas vezes: a primeira vez através do comando inserido no laço e a segunda controlada pelo comando `para/faça`. Por exemplo, o comando

```
para var_contr de 1 incr 1 até 10 faça
   início
   var_contr ← var_contr + 1
   escrever (var_contr)
   fim
```

exibe somente os valores pares: a variável de controle `var_contr` é inicializada em 1, valor com o qual é executado o comando composto. O primeiro comando contido no comando composto incrementa novamente `var_contr`, que passa a valer 2, valor que é então exibido. O controle volta ao comando `para/faça`, onde `var_contr` é novamente incrementada em uma unidade, passando a valer 3. Novamente é incrementada no corpo do comando, passando a valer 4, e o valor é apresentado. O último valor informado será 10. Observar que, mesmo se utilizado intencionalmente, esse recurso descaracteriza a estrutura do comando de repetição `para/faça`, não sendo, por isso, considerado adequado pela programação estruturada.

exemplo. O Algoritmo 4.4, visto no Capítulo 4, mostra o processo completo para a obtenção das notas de um aluno, bem como cálculo e informação da média e conceito obtidos. Observando as entradas e saídas especificadas nesse algoritmo, vê-se como entradas as variáveis `nota1`, `nota2`, `nota3` e, como saídas, as variáveis `média` e `conceito`. Esses dados se referem a um único aluno. Mas, e se a turma for composta de 30 alunos? Sem utilizar repetição, seriam necessárias cinco variáveis para cada aluno, num total de 150 variáveis. Contudo, os comandos a serem repetidos se referem ao processamento de um único aluno, isto é, a leitura de três notas, o cálculo da média, o cálculo do conceito e a informação da média e do conceito obtidos por esse aluno. No caso de 30 alunos, esse procedimento deverá ser repetido 30 vezes, cada repetição correspondendo ao processamento completo das informações de um aluno. As mesmas áreas de memória podem ser preenchidas em diferentes momentos com diferentes conteúdos, uma vez que os dados de cada aluno são processados isoladamente e de forma independente. Assim, a solução do problema é obtida através da repetição, por 30 vezes, dos mesmos comandos utilizados no Algoritmo 4.4. O Algoritmo 5.1 mostra esse processamento, identificando cada aluno pela sua ordem.

```
Algoritmo 5.1 - Média30
{INFORMA MÉDIA E CONCEITO DOS 30 ALUNOS DE UMA TURMA}
  Entradas: nota1, nota2, nota3 (real)
  Saídas: aluno (inteiro)
          média (real)
          conceito (caractere)
  Variável auxiliar: aluno (inteiro)        {VARIÁVEL DE CONTROLE}
início
  para aluno de 1 incr 1 até 30 faça     {INICIAL, INCREM E FINAL}
```

```
    início
    ler (nota1, nota2, nota3)                {ENTRADA DAS 3 NOTAS}
    média ← (nota1 + nota2 + nota3) / 3           {CALCULA MÉDIA}
    escrever (aluno, média)  {INFORMA ORDEM DO ALUNO E SUA 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}
    escrever (conceito)                        {INFORMA CONCEITO}
    fim {PARA/FAÇA}
fim
```

utilização de constantes no cabeçalho. O valor final utilizado em comandos de repetição por contagem costuma corresponder ao número de elementos envolvidos no problema em questão. No exemplo anterior, o valor final 30 corresponde ao número de alunos a processar. Se o desejado fosse, no mesmo exemplo, calcular a média da turma, seria utilizado esse mesmo valor como divisor da expressão aritmética, onde a soma das médias individuais obtidas por cada aluno seria dividida pelo número de alunos considerados. E esse mesmo valor seria utilizado em outras sequências de instruções envolvendo o processamento de informações referentes a essa turma. Consequentemente, a alteração do número de alunos processados implicaria na alteração do valor 30 em diferentes pontos do programa. Essas alterações, além de trabalhosas, podem gerar erros de execução resultantes da utilização equivocada desse valor em algum ponto do programa. Para evitar esse problema, aconselha-se a declaração e a utilização de uma constante (Seção 2.2.3) contendo esse valor, evitando assim a necessidade de especificações e alterações repetidas em diferentes pontos do programa. Também o valor inicial da variável de controle pode ser definido através de uma constante, facilitando sua alteração.

O Algoritmo 5.2 mostra a utilização de constantes em comandos `para/faça`. Aqui, a constante `NALUNOS` é utilizada como limite do laço de repetições e como divisor no cálculo da média das notas da turma. Notar que o conteúdo atual da variável de controle é usado dentro do laço de repetições para informar o número sequencial do aluno cuja média está sendo mostrada.

Algoritmo 5.2 - MédiaAlunoETurma

```
{INFORMA MÉDIA DOS ALUNOS DE UMA TURMA E A MÉDIA GERAL DESTA TURMA}
  Entradas: nota1, nota2, nota3 (real)
  Saídas:
    média (real)
    média_turma (real)
  Variáveis auxiliares:
    contador (inteiro)    {VARIÁVEL DE CONTROLE}
```

```
    soma_médias (real)
  Constante: NALUNOS = 30 {NÚMERO DE ALUNOS DA TURMA}
início
  soma_médias ← 0  {SOMA MÉDIAS INDIVIDUAIS: VALOR INICIAL ZERO}
  para contador de 1 incr 1 até NALUNOS faça
    início
    ler (nota1, nota2, nota3)                {ENTRADA DAS 3 NOTAS}
    média ← (nota1 + nota2 + nota3) / 3           {CALCULA MÉDIA}
    escrever (contador, média)  {INFORMA NÚMERO DO ALUNO E MÉDIA}
    soma_médias ← soma_médias + média           {SOMA DAS MÉDIAS}
    fim
  média_turma ← soma_médias / NALUNOS            {MÉDIA DA TURMA}
  escrever (média_turma)
fim
```

5.2.1 aninhamento de comandos para/faça

Qualquer comando pode ser incluído no laço de repetição, inclusive outro comando para/faça. Quando ocorre o aninhamento de comandos para/faça, a cada iteração do laço externo, o laço interno é executado de forma completa.

No trecho de algoritmo a seguir, a tabuada dos números 1 a 10 é gerada a partir do aninhamento de comandos para/faça. Observar a inclusão dos comentários sinalizando o final dos laços. Os comentários, embora não obrigatórios, aumentam a legibilidade dos algoritmos e programas que incluem aninhamentos, funcionando de forma complementar à indentação.

```
para multiplicando de 1 incr 1 até 10 faça      {LAÇO EXTERNO}
  início
  escrever ('Tabuada de', multiplicando)
  para multiplicador de 1 incr 1 até 10 faça
    início       {LAÇO DA GERAÇÃO DA TABUADA DE MULTIPLICANDO}
    produto ← multiplicando * multiplicador
    escrever (multiplicando, ' X ', multiplicador, ' = ', produto)
    fim {DO LAÇO DO MULTIPLICADOR}
  fim {DO LAÇO DO MULTIPLICANDO}
```

Nesse outro exemplo de aninhamento de comandos para/faça, a variável de controle do laço externo é incrementada até um valor constante e a variável de controle do laço interno é decrementada até o valor corrente da variável de controle do laço externo:

```
para n1 de 1 incr +1 até 2 faça          {LAÇO EXTERNO}
   para n2 de 3 incr -1 até n1 faça      {LAÇO INTERNO}
      escrever(n1, n2)
```

Os valores de n1 e n2 mostrados pelo comando escrever seriam:

```
1   3
1   2
1   1
2   3
2   2
```

O aninhamento de repetições também ocorre quando é utilizado um comando composto que contenha outro comando para/faça. Nesse caso, a cada repetição são executados os comandos que compõem o comando composto, na ordem definida, e todas as repetições do comando para/faça interno. Por exemplo, no comando:

```
para k de 1 incr 1 até 10 faça          {LAÇO EXTERNO}
   início                                {COMANDO COMPOSTO}
   ler (número)
   para valor de 1 incr 1 até número faça
      {PARA/FAÇA INTERNO AO COMANDO COMPOSTO}
      escrever(valor)
   fim
```

para cada valor do laço externo, a cada uma das 10 execuções do laço externo, é lido um novo valor para a variável número, que serve de limite para a repetição do laço interno.