



■ ■ série livros didáticos informática ufrgs ■ ■

int divpares;

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

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

Os algoritmos que solucionam os problemas apresentados até o momento são puramente sequenciais: todas as instruções são executadas na ordem em que foram definidas, uma após a outra, sem exceção. Este capítulo introduz uma nova classe de problemas, na qual uma ou mais ações podem ou não ser executadas, dependendo da avaliação prévia de condições. Para resolver esses problemas são apresentados três novos comandos que possibilitam a alteração do fluxo sequencial de execução de um programa.

No capítulo anterior, foi resolvido o problema do cálculo da média aritmética de três notas de um aluno em uma disciplina. Estendendo essa aplicação, o professor que utiliza esse programa quer que, além da média das notas obtidas na disciplina, seja informado se o aluno foi aprovado (no caso de sua média ser igual ou superior a 6). Essa informação não pode ser obtida somente com o conjunto de comandos do Capítulo 3, uma vez que requer a análise de uma condição e a execução de uma ação somente se a condição analisada for verdadeira.

Outra possibilidade é fornecer não apenas a informação de aprovação, mas também a de reprovação. Nesse caso, também condicionada ao conteúdo da média calculada, ocorre a execução de apenas uma de duas ações mutuamente exclusivas: se a média for maior ou igual a 6, o programa informa que o aluno foi aprovado; caso contrário, se a média for inferior a 6, informa que o aluno foi reprovado.

Seguindo adiante nesse raciocínio, pode-se escrever um trecho de programa em que, dependendo da média obtida, também é informado o conceito correspondente à média calculada. Aqui, diferentes faixas de valor de uma mesma informação desencadeiam ações diferentes, também mutuamente exclusivas, isto é, a execução de uma ou mais instruções específicas está associada ao valor da informação.

Os comandos que solucionam esses problemas são apresentados neste capítulo.

4.1

→ comando de seleção simples

Um **comando de seleção simples**, também chamado de comando condicional, permite que a execução de um trecho do programa dependa do fato de uma condição ser verdadeira, isto é, vincula a execução de um ou mais comandos ao resultado obtido na avaliação de uma expressão lógica (também denominada expressão condicional). O comando de seleção simples é sempre composto por uma condição e um comando. A condição é expressa por uma expressão lógica, cuja avaliação produz um resultado verdadeiro ou falso. A sintaxe de um comando de seleção simples é:

```
se <expressão lógica>  
então <comando>
```

Observe que o comando somente é executado se o resultado da expressão lógica for verdadeiro; se o resultado for falso, nada é executado.

Por meio do comando de seleção simples pode-se, por exemplo, condicionar a exibição da informação de que um aluno foi aprovado somente para o caso de sua média ser igual ou superior a 6:

```
se média ≥ 6
então escrever('Aprovado')
```

A execução do comando `escrever` ocorre apenas quando a condição for verdadeira, ou seja, quando o conteúdo da média for igual ou superior a 6. Nada é executado se a média for inferior a 6.

A Figura 4.1 representa o fluxograma de um comando de seleção simples. Um novo tipo de bloco, com formato de losango, é utilizado para representar a realização de um teste, escrevendo-se dentro desse bloco a expressão lógica a ser avaliada. Esse bloco tem duas saídas, uma para o caso da expressão ser avaliada como verdadeira, e outra para quando o resultado da avaliação da expressão for falso. As informações que correspondem a cada saída devem estar claramente identificadas. O fluxograma mostra com clareza que nada é executado no caso do resultado da avaliação da expressão ser falso.

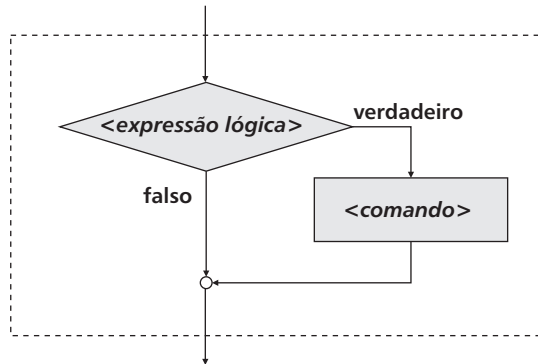


figura 4.1 Fluxograma de um comando de seleção simples.

O algoritmo a seguir informa, além da média de um aluno, se ele foi aprovado:

Algoritmo 4.1 - Média2

```
{ INFORMA A MÉDIA DAS 3 NOTAS DE UM ALUNO E SE ELE FOI APROVADO }
  Entradas: nota1, nota2, nota3 (real)
  Saídas: média (real)
         { Informação de aprovado }
início
  ler (nota1, nota2, nota3)      { ENTRADA DAS 3 NOTAS }
  média ← (nota1 + nota2 + nota3) / 3
  escrever (média)              { INFORMA MÉDIA CALCULADA }
  se média ≥ 6
  então escrever('Aprovado')    { INFORMA SE ALUNO FOI APROVADO }
fim
```

A Figura 4.2 apresenta o fluxograma relativo a esse algoritmo.

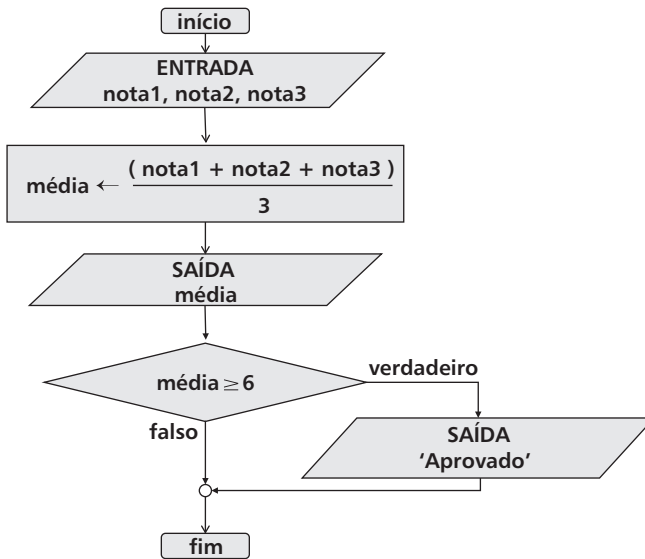


figura 4.2 Fluxograma de um exemplo com comando de seleção simples.

4.2

→ comando composto

Na sintaxe do comando de seleção simples nota-se que somente um comando pode ser executado caso a condição seja verdadeira. Mas o que fazer quando se quer executar vários comandos condicionados à avaliação de uma mesma expressão lógica? Por exemplo, supondo que, na aplicação anterior, se queira saber a média somente no caso das três notas lidas serem iguais ou superiores a 6, então a média deve ser calculada e informada somente se a condição for verdadeira.

Para que isso seja possível, é necessário que dois comandos, o de cálculo da média e o de saída dessa média, sejam executados quando a condição for verdadeira. Como a sintaxe do comando de seleção simples exige a execução de um único comando, faz-se necessária a definição de um novo tipo de comando, denominado **comando composto**.

Na pseudolinguagem aqui utilizada, um comando composto é delimitado pelas palavras reservadas *início* e *fim*. Sintaticamente, trata-se de um único comando. Quaisquer comandos podem ser incluídos dentro de um comando composto. Algumas linguagens de programação permitem, inclusive, a definição de novas variáveis dentro de um comando composto, as quais são alocadas na memória apenas no momento em que inicia a execução desse coman-

do e são liberadas no término da execução do comando. Contudo, essa possibilidade não será considerada neste livro.

O problema proposto no início desta seção pode ser resolvido com a utilização desse novo recurso, conforme mostrado no Algoritmo Média3:

Algoritmo 4.2 - Média3

{ INFORMA MÉDIA DO ALUNO SOMENTE SE SUAS 3 NOTAS FOREM IGUAIS OU SUPERIORES A 6 }

Entradas: nota1, nota2, nota3 (real)

Saídas: média (real)

{ Informação de aprovado }

início

ler (nota1, nota2, nota3)

{ ENTRADA DAS 3 NOTAS }

se (nota1 ≥ 6) e (nota2 ≥ 6) e (nota3 ≥ 6)

então início

{ COMANDO COMPOSTO }

média ← (nota1 + nota2 + nota3) / 3

{ CALCULA MÉDIA }

escrever (média)

{ INFORMA MÉDIA }

fim

fim

O fluxograma desse algoritmo, apresentado na Figura 4.3, mostra com clareza que o conjunto de comandos pode não ser executado, dependendo do resultado da condição.

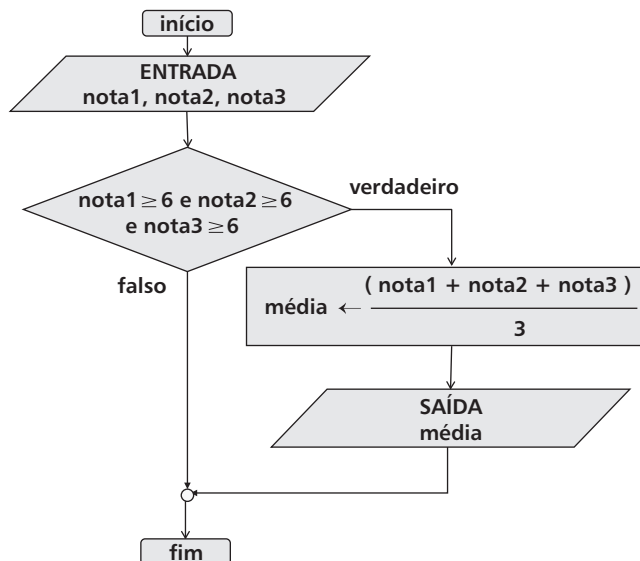


figura 4.3 Fluxograma de comando de seleção simples com comando composto.

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