



Algoritmos e Lógica de Programação

ALGORITMOS E PROGRAMAÇÃO: CONTEXTOS E PRÁTICAS

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 1 Algoritmos e programação contextos e práticas	7
Seção 1.1 - Histórico e definição de algoritmos: perspectivas de linguagem	11
Seção 1.2 - Tipos de dados e expressões: literais, lógicas e aritméticas	25
Seção 1.3 - Representação de algoritmos e o ambiente de programação	39
Seção 1.4 - Declaração de variáveis e constantes	51

Palavras do autor

Olá, aluno, seja bem-vindo.

Nesta unidade de ensino você conhecerá o que é algoritmo e a sua relação com as linguagens de programação, assim como o modo como esses auxiliam no desenvolvimento do raciocínio lógico. A partir de agora, você será apresentado aos conceitos e conteúdos componentes deste tema. Iniciamos com uma breve definição e, em seguida, é apresentado um levantamento dos principais algoritmos que deram origem a esse estudo e compreender o ambiente de programação, bem como conhecer as formas de apresentação e desenvolvimento de um algoritmo. Saber distinguir as expressões literais, lógicas e aritméticas também é importante no âmbito computacional. Além desses, conhecer os tipos de dados, variáveis e constantes, compõem a unidade 1 do seu material de estudos.

Para a unidade 2, o foco está nas operações e comandos que permitem a entrada e saída de dados, o ponto de partida para aprender a manipular as informações. Associadas a esse conceito, estão também a atribuição de valores e a forma de realizar os procedimentos para a declaração das variáveis ou das constantes e, finalmente, iniciam-se os estudos em estruturas de seleção como "if/else" (se/então/senão e CASE). Com ela, você pode compreender a lógica envolvida nas estruturas de programação condicionais simples, compostas e/ou encadeadas.

A Unidade 3 é responsável por promover a imersão em situações reais para aplicar os conhecimentos mais específicos, tais como: *Switch-CASE*, que permite ao usuário escolher dentre as opções predeterminadas e, ainda, estruturas de repetição como "for" (para), "do" (fazer) e "do/while" (faça enquanto). Na Unidade 4, a atenção está sobre o entendimento dos vetores e matrizes, sua estrutura, eficiência e aplicações.

Após todos os conceitos, o material de estudos ainda contribui para que você possa treinar os conhecimentos adquiridos, realizando exercícios orientados, que visam aproximar a situação real e os problemas enfrentados no dia a dia do desenvolvimento de *software*. Desejamos a você, desde já, bons estudos e dedicação para a conclusão desta etapa.

ALGORITMOS E PROGRAMAÇÃO: CONTEXTOS E PRÁTICAS

Convite ao estudo

Para iniciar os estudos em algoritmos e programação, é interessante refletir sobre o que são e quais algoritmos influenciam diretamente nas transações comerciais que se estabelecem na atualidade e que detêm uma relevância no desenvolvimento de soluções práticas e inovadoras. Esses trazem possibilidades de empreendimentos entre os mais diversos segmentos do mercado. A partir de tais informações e compreendendo a relevância do tema, conheça a competência de fundamento de área que se pretende desenvolver com este estudo e os objetivos específicos desta unidade.

Competência de fundamento 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.

Objetivos específicos

- Conhecer o que são, como se aplicam e a quem se destina a elaboração dos algoritmos;
- Conhecer como se desenvolve estruturas computacionais com expressões literais, lógicas e aritméticas, além dos tipos de dados;
- Conhecer o ambiente de programação e como se dá o raciocínio lógico computacional, bem como os tipos e formas de representação de algoritmos através da elaboração do fluxograma e do pseudocódigo;

- Conhecer e saber identificar o que são variáveis e constantes e como se faz a declaração de variáveis.

Importante trazer as definições pertinentes ao tema: algoritmo pode ser definido como um conjunto de processos ou ações, que seguem uma sequência lógica, para executar uma tarefa. Por exemplo, baixar um aplicativo em seu celular requer que você execute um passo-a-passo para realizar aquela instalação e o configure para poder utilizar. A esse passo-a-passo dá-se o nome de algoritmo.

Para desenvolver algumas das atividades aqui apresentadas, você poderá considerar, entre outros, o cenário que envolve os comerciantes do Litoral Sul do país. Suponha que eles estejam se organizando em cooperativa para angariar recursos e melhorar as transações comerciais deste mercado. Contrataram uma consultoria para desenvolver um aplicativo, e suponha que você faz parte dessa equipe. O grande desafio dos comerciantes e da empresa contratada para o desenvolvimento desse aplicativo é achar uma solução que atenda à necessidade de divulgação, organização das informações, facilidade de uso e acesso para os seus clientes o encontrarem, agendarem ou mesmo tomar conhecimento daquele determinado estabelecimento de hotelaria ou gastronomia. Um ponto de atenção é que precisam realizar todos os processos, desde a determinação das ações do sistema e tarefas, até a escolha das ferramentas de análise e desenvolvimento do aplicativo.

Uma das propostas da empresa é disponibilizar um protótipo do aplicativo para celular que informe aos usuários quais são os serviços de hotelaria e gastronomia disponíveis naquela região. A partir desta situação, os comerciantes esperam obter soluções que apresentem uma forma de contabilizar o índice de satisfação do usuário quanto à facilidade de navegação e uso do aplicativo. Para tal, 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.

O primeiro passo é compreender o que realmente será possível fazer e estipular as ações do sistema. Para tal, você precisa entender que é necessário identificar quais são as suas variáveis, bem como as estruturas que precisarão ser desenvolvidas para trabalhar com os dados, armazená-los, atribuir, excluir e alterar valores, verificar se há variáveis ou se serão declaradas como constantes, entre outros conceitos que precisam ser aplicados. Com o estudo de algoritmos e programação, a identificação desses passos será facilitada e, ainda, você poderá compreender as formas de representação, além de sua definição.

Atente-se também aos tipos de dados, às expressões literais, aritméticas e lógicas que serão necessárias para realizar um cálculo específico, como a média dos acessos, por exemplo, ou ainda, se deverá executar algum teste para verificar e informar aos comerciantes os resultados. Estabeleça um processo de verificação e controle de como possibilitar que o sistema contabilize os acessos, realize o cálculo da média através da indicação de sua expressão aritmética, indique uma forma de registrar essas informações e enviá-las aos interessados.

Muito importante, nesta primeira entrega, é que seja especificado o fluxograma dos processos que o sistema deverá executar e os passos a seguir para o seu desenvolvimento. Essa é uma tarefa que requer a leitura dos materiais e indicações componentes dessas referências. Fica a sugestão do desenvolvimento das estruturas do algoritmo em pseudocódigo, o que permite que você identifique os processos e a necessidade de alterações, antes do desenvolvimento efetivo do sistema. Encontre a lógica necessária para otimizar o processamento e uso dos recursos. Então, bons estudos e práticas a você.

Seção 1.1

Histórico e Definição de Algoritmos: perspectivas de linguagem

Diálogo aberto

Você já deve ter observado que todas as ações que realizamos seguem uma sequência, e que essa precisa ser executada para que se consiga atingir o objetivo predeterminado. Por exemplo, quando é necessário trocar o pneu do carro, o que você observa é que existem passos a cumprir. Para resolver uma situação simples ou complexa, uma sequência lógica sempre terá de ser realizada, portanto.



Refleta

"Para programarmos em um computador, devemos conhecer e entender o que são os algoritmos, e como utilizá-los para determinar a sequência de passos necessários para resolvermos determinados problemas ou, em outras palavras, encontrarmos a solução, ou a melhor solução, para a implementação em uma linguagem de programação." (PIVA JR. et al., 2012, p. 3)

Você sabe o que é uma linguagem de programação?

As linguagens de programação visam possibilitar a inserção dos passos ou ações que o *software* deverá executar, através de uma plataforma que permita, desde que respeite a sintaxe de cada uma, expressar a sequência lógica determinada no algoritmo, ou a sua construção propriamente dita.

Por exemplo, para passar um algoritmo de linguagem natural, que é a forma como falamos como precisa ser executada uma tarefa, para uma linguagem de programação estruturada como "C", ou mesmo, orientada a objetos como "Java" ou "C#", é preciso, além de estabelecer o passo-a-passo, seguir a forma como esta interpretará essas instruções, de forma que sejam compreendidas pelo computador.

Esse, por sua vez, utilizará os processos de compilação (tradução e interpretação para a linguagem de máquina), ou seja, codifica e decodifica para permitir que o

resultado seja visualizado em sua tela com o respectivo processamento da informação que foi solicitada.

Em algoritmos é comum utilizar os *softwares* VisualG (pseudocódigo) ou o Scilab (resolução principalmente de problemas matemáticos) para a execução dessas tarefas.

Sendo assim, a partir de agora é preciso definir, seja em linguagem natural, em pseudocódigo ou fluxograma, um algoritmo que deixe clara a sequência lógica que terá de ser cumprida para a solução do problema que os comerciantes do Litoral Sul têm que resolver.

Siga em frente.

Não pode faltar

A fim de compreender esse processo de descoberta e desenvolvimento dos algoritmos, vamos recapitular algumas informações, tais como: qual é a organização básica do computador; o que é linguagem de máquina; como ocorre a comunicação entre elas; o código ASCII; o que são os programas de computadores; como é a lógica de programação e a forma como acontece a resolução de problemas pelo computador, ou seja, como se dá o processamento da informação. Após compreender esses conceitos, você já poderá avançar em seus estudos sobre algoritmos e programação. Você se recorda de como é a organização dos computadores?

Esses são divididos em quatro unidades básicas: unidade de entrada, unidade de saída, unidade de processamento central e memória.



Assimile

- **Unidade de entrada:** em que ocorre a entrada de dados. Ex.: teclado, mouse.
- **Unidade de saída:** há a saída de informações. Ex.: monitor, impressora.
- **Unidade de Processamento Central:** responsável pelo processamento das informações e alocação de recursos.
- **Memória:** armazenamento de dados (RAM, HD, ROM, Cache).

Já o código ASCII (American Standard Code for Information Interchange ou Código Padrão Americano para Intercâmbio de Informações) é o responsável por converter os caracteres para a sua respectiva combinação binária. Cada caractere é composto por

oito bits (8 bits correspondem a 1 byte). Esta se chama linguagem de máquina (binária). Ela possui apenas dois símbolos: 0 e 1, conhecidos por bit (binary digit).



Pesquise mais

A tabela ASCII completa está disponível no site: <http://www.ascii-code.com/>. Acesso em: 25 fev. 2015.

Além destes elementos a considerar, os computadores precisam de *softwares* que auxiliem no desenvolvimento de atividades, sejam elas pessoais ou profissionais. Existem *softwares* para realizar as mais variadas tarefas. O desenvolvimento de um *software* contempla algumas fases, como análise, onde se entende qual o produto de *software* que será gerado; o projeto, em que há as especificações de cada uma das tarefas que ele executará; a implementação, que será realizada com uma linguagem de programação, e os testes, que são realizados a fim de conferir se o *software* atende aos processos solicitados. Para que sejam passíveis de implementação, essas tarefas seguem a sequência lógica, chamada algoritmo.

Tal sequência facilita a transposição das ações que o software precisará executar, mas desenvolvidas de acordo com a sintaxe da linguagem de programação que se escolheu, seja orientada a objetos ou estruturada.



Refleta

"Como a comunicação entre as unidades do computador teria que ser obtida através de fenômenos físicos, os cientistas que conceberam os computadores atuais estabeleceram dois símbolos básicos para a linguagem. Esta quantidade de símbolos foi escolhida pelo fato de que através de fenômenos físicos é muito fácil obter dois estados distintos e não confundíveis, como passar corrente elétrica/não passar corrente elétrica, estar magnetizado/não estar magnetizado, etc., podendo cada um destes estados ser um dos símbolos." (EVARISTO, 2002, p. 6)

Os algoritmos possuem cinco propriedades que os caracterizam. São elas (KNUTH, 1973 apud PIVA JR., 2012):

- Finitude: indica que o algoritmo deve encerrar após um número finito de execução.
- Definição: evita definições que gerem ambiguidade.
- Entrada: atribui valores ao algoritmo, a partir de especificações de variáveis e funções antes de sua iniciação.

- Saída: representa os valores após o processamento, ou seja, a execução das ações.
- Eficácia: todas as ações que o algoritmo realiza precisam ser executáveis em um limite de tempo predeterminado, finito, mesmo que seja sem o auxílio do computador.

O esquema a seguir representa o mecanismo de funcionamento de um algoritmo.



Fonte: Adaptado de Piva Jr, 2012, p. 7.



Pesquise mais

Para auxiliá-lo, sugerimos um *link* em que poderá conhecer os mais diversos tipos de algoritmos e testar a sua eficiência. Disponível em: <<http://nicholasandre.com.br/sorting/>>. Acesso em: 02 mar. 2015.

Antes de aprender a utilizar uma linguagem de programação específica, é importante que você compreenda que programar, de acordo com Ziviane (2007, p. 1), é “basicamente estruturar dados e construir algoritmos”. A partir da análise de um determinado problema, você precisa, então, implementar o algoritmo que foi desenvolvido. Há dois tipos de problemas que envolvem os algoritmos: o primeiro trata da análise de um algoritmo único, parte a parte, e investiga a quantidade de execuções de um determinado processo, buscando o seu aprimoramento para poupar recursos da máquina; o outro é a análise de uma classe de algoritmos, esse estudo implica na análise de vários algoritmos a fim de identificar um que seja viável e compatível para apresentar como sendo a solução de um determinado problema.

O custo de um algoritmo também pode ser mensurado, a partir do princípio da análise de sua complexidade, chamada de “f”, com relação a um algoritmo “n”, ou seja, $f(n)$ representa a função que mede o tempo gasto por um algoritmo para executar um dado problema de tamanho n . A esta função “f” dá-se o nome de complexidade de tempo do algoritmo. Já $f(n)$, que mede a quantidade de recursos de memória utilizados para a sua execução, é chamada de função de complexidade de espaço (ZIVIANE, 2007), ou seja, fatores como tempo e espaço devem ser levados em consideração quando se trata da eficiência de um algoritmo.

Agora veja um exemplo de como se dá a lógica referente à construção de algoritmos. Ele representa a realização das atividades seguindo passos, etapas. O algoritmo a seguir trabalha com o problema das Torres de Hanói.

Observe que há as três hastes, serão aqui chamadas respectivamente de A, B e C. A primeira haste (A) tem anéis com diâmetros diferentes e que estão organizadas em ordem decrescente. O desafio está em transferir os anéis da haste A para a B. Para resolver essa questão, você precisa respeitar algumas regras:

- a. Não pode mover o disco por mais de uma haste por vez, por jogada;
- b. É preciso respeitar a ordem decrescente dos diâmetros, ou seja, do menor para o maior.

Figura 1 | Torres de Hanói



Fonte: Criado pelo autor, adaptado de Ziviane, 2007, p. 6.

Confira abaixo uma das possíveis formas de se resolver esse problema, repetindo quantas vezes for necessário a sequência, de acordo com a quantidade de anéis existentes na torre.

Exemplo 1: Algoritmo Torres de Hanói

Início

1. Mover o primeiro anel de A para B.
2. Mover o segundo anel de A para C;
3. Mover o anel de B para C;
4. Mover o terceiro anel de A para B;
5. Mover o anel de C para A;
6. Mover o anel de C para B;
7. Mover o anel de A para B.

Fim

Note que o algoritmo está escrito em linguagem natural, porém, é necessário que esse seja formalizado de acordo com um conjunto de regras que chamamos de sintaxe. Além desta, há as regras semânticas, ou seja, de lógica. Através de comandos, será possível cumprir o conjunto de regras que o computador interpretará.

A partir do uso dos comandos, será possível escrever as expressões de forma que o computador consiga interpretar essas informações. Essas expressões, que podem ser literais, aritméticas ou lógicas, realizam operações com os dados que são atribuídos. Os comandos recebem o nome de estruturas de programação e podem ser sequenciais, de decisão ou de repetição. Quando você for desenvolver um algoritmo, lembre-se de que, além das estruturas determinadas, também devem ser considerados os tipos de dados, que são as categorias que permitem a aplicação de um conjunto de comando que a máquina irá interpretar e traduzir para valores binários.

Para manipular os dados, é preciso usar as variáveis e os valores constantes. Há também a atribuição de valores para essas variáveis, que determinam quais são os operadores, as funções e os procedimentos necessários para a solução do problema.

Abaixo um outro exemplo de algoritmo, porém, em linguagem não computacional que representa uma tarefa do cotidiano: realizar a troca de uma lâmpada:

Início

- a. Verifica se o interruptor está desligado.;
- b. Pega uma escada;
- c. Posiciona a escada no local;
- d. Sobe a escada;
- e. Retira a lâmpada queimada;
- f. Coloca a lâmpada nova;
- g. Desce da escada;
- h. Acende a lâmpada no interruptor;
- i. Se a lâmpada não acender, então;
- j. Retira novamente a lâmpada queimada;
- k. Troca por uma lâmpada nova.

- l. Senão
- m. Descarta a lâmpada queimada;
- n. Guarda a escada;
- o. Encerra a tarefa.

Fim



Refleta

Lógica é uma área da Matemática cujo objetivo é investigar a veracidade de suas proposições. (SOUZA et al., 2011, p. 19)



Exemplificando

Exemplo em linguagem natural

Início

1. Solicitar ao usuário que digite os valores da base (b) e da altura (h).
2. Calcular a área (A) com a fórmula: $A = (b \cdot h) / 2$.
3. Exibir o valor da área (A).

Fim

Pseudocódigo

(Portugol)





Início

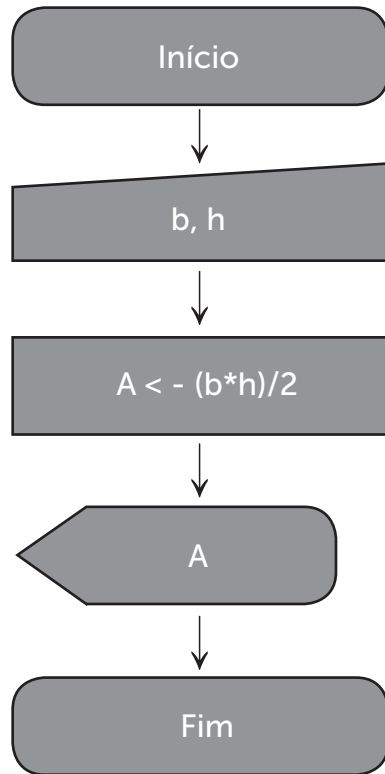
1. Leia (b, h).
2. $A \leftarrow (b * h) / 2$.
3. Escreva (A).

Fim

b. Fluxograma

**Saiba mais**

-  Terminador
-  Processo
-  Entrada manual
-  Exibição



Agora, observe o exemplo abaixo que busca resolver uma situação real no ambiente computacional. Suponha que você precise desenvolver um algoritmo, representado pelo seu pseudocódigo e também por um fluxograma. Ao invés de utilizar a linguagem natural, será preciso iniciar o aprendizado de acordo com uma sintaxe e a lógica computacional. O problema em questão é: algoritmo para calcular a área de um triângulo.

**Saiba mais**

A função "Se" representa a inserção de uma condição a ser testada para que haja o próximo passo, que é o cálculo realizado. Leia o artigo disponível em: http://www.ufpa.br/sampaio/curso_de_icc/icc/aula%2010/comando_se_entao_senao.htm. Acesso em: 24 mar. 2015.



Faça você mesmo

Para especificar um algoritmo, você deve ter em mente que primeiro é preciso conhecer o problema a ser resolvido. Depois, você terá de modelar o problema, isso é, detalhar os processos. Essa modelagem só é possível com o exercício de abstração que é realizado para identificar o passo-a-passo para resolvê-lo. Essa sequência lógica permitirá chegar a uma solução que atenda às convenções para a sua elaboração. Vamos a mais um exemplo.

No bairro São João da Barra, na cidade de Mirandópolis, a companhia telefônica lançou uma promoção em que a cada 30 moradores que fizessem a adesão do seu plano de TV, *internet* e telefone, pagariam apenas o valor de R\$ 100,00 pelas assinaturas e, além disso, ganhariam um bônus de R\$ 67,00 no primeiro mês da assinatura. Descubra qual o valor de cada assinatura.

Algoritmo em linguagem natural	Algoritmo em lógica computacional
Início 1. Inserir valores 30, 67 e 100. 2. Subtrair de 67 de 100. 3. Dividir o resultado da subtração por 30. 4. 4. Exibir o resultado. Fim	Início 1. Leia os valores A, B e C. 2. $s \leftarrow B - C$. 3. Escreva o resultado. Fim

A resolução matemática será uma equação do primeiro grau em que "x" é o custo de cada assinatura, então temos:

$$30x + 67 = 100$$

$$30x = 100 - 67$$

$$x = 33/30$$

$$x = 1.1$$

Porém, as regras matemáticas precisam ser consideradas, e como não há divisão por zero, é preciso inserir essa condição à operação agregando uma estrutura de decisão ou seleção que determine o teste que deverá ser realizado. Vamos lá. Veja o algoritmo correto:

Início

1. Leia A, B e C.
2. **Se** $C > B$ **e** $A > 0$ **e** $B \geq 0$ **e** $C > 0$ **então**.

1. $s \leftarrow C - B$.
2. $\text{valor_ass} \leftarrow s/A$.
3. Escreva o valor da assinatura, valor_ass .

3. Senão

1. Escreva que os valores não permitem a divisão.

Fim



Vocabulário

Sintaxe: ordem e disposição de apresentar as palavras para que sejam interpretadas. Programação estruturada: paradigma de programação que segue a premissa de que contempla estruturas de seleção de informações, decisão e repetição.

Programação orientada a objetos: paradigma de programação que aproxima o mundo real do virtual através da abstração dos dados, vinculação das informações através de herança, encapsulamento e polimorfismo.



Assimile

[...] o papel da lógica na programação: provar que um algoritmo que foi elaborado está correto. Juntamente com a simulação do algoritmo, que consiste em executá-lo com dados reais, é possível saber se está correto e se leva a valores consistentes (SOUZA et al., 2011, p. 21).

Sem medo de errar

Agora, vamos iniciar o desenvolvimento da solução ao caso dos comerciantes do Litoral Sul. Considere que você realizou uma entrevista com os *stakeholders* desse projeto, ou seja, com os comerciantes e demais envolvidos neste trabalho.

Assim, você precisa identificar os processos e escrever em estrutura de algoritmo, mas em linguagem natural, os passos que anotou. Siga em frente. Verifique se o algoritmo a seguir cabe como sugestão de resolução do problema e proponha, também, outras formas de resolver, de maneira que seja otimizado o processamento da informação.

A princípio, estabeleça também quais serão os passos necessários para a sua conclusão. Veja abaixo a proposta.

Algoritmo de execução do projeto

1. Início
 2. Levantar requisitos do aplicativo junto aos comerciantes.
 3. Escrever o plano de ação do projeto.
 4. Obter a aprovação junto aos clientes.
 5. Elaborar o algoritmo do software em questão em linguagem natural e em pseudocódigo.
 6. Desenvolver o fluxograma das ações do aplicativo.
 7. Direcionar à equipe de desenvolvimento do projeto para a implantação.
 8. Desenvolver a solução utilizando uma linguagem de programação.
 9. Realizar os testes.
 10. Disponibilizar o aplicativo.
- Encerrar o projeto.



Atenção!

Busque a compreensão das ações que o sistema executará e que serão entregues nesta primeira fase.



Lembre-se

Você precisa identificar o passo-a-passo de todas as ações que o *software* deverá executar. Isto, é válido tanto para a elaboração de um algoritmo simples como trocar um pneu, como para a resolução de um problema complexo.

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 as compare com a de seus colegas e com o gabarito disponibilizado no apêndice do livro.

Histórico e Definição de Algoritmos: perspectivas de linguagem	
1. Competência de fundamento 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 o que são, como se aplicam e a quem se destina a elaboração dos algoritmos; • Conhecer como ocorre o raciocínio lógico computacional.
3. Conteúdos relacionados	Definição de algoritmos; Histórico e perspectivas para a linguagem; Linguagem natural; Pseudocódigo (Português Estruturado)
4. Descrição da SP	Em uma gincana escolar, o desafio era fazer com que dois grupos de três integrantes cada conseguissem atravessar uma ponte em que só é possível passar dois alunos por vez. No entanto, como são grupos diferentes, não é permitido que em qualquer dos lados da ponte fiquem mais alunos do grupo 2 (G2) que do grupo 3 (G3). Então, elabore o algoritmo que respeite essas regras e demonstre em linguagem natural e faça o teste conforme a indicação que segue.
5. Resolução da SP	1. Atravessem o aluno1 do G1 e aluno1 do G2. 2. Volte o aluno1 do G1. 3. Atravessem o aluno2 do G2 e aluno3 do G2. 4. Volte o aluno1 do G2. 5. Atravessem o aluno1 do G1 e aluno2 do G1. 6. Volte aluno1 do G1 e aluno2 do G2. 7. Atravessem o aluno1 do G1 e aluno3 do G1. 8. Volte o aluno3 do G2. 9. Atravessem o aluno1 do G2 e o aluno2 do G2. 10. Volte aluno2 do G2. 11. Atravessem aluno2 do G2 2 aluno3 do G2.



Lembre-se

Acesse o *link* e repita a sequência com o problema original entre os missionários e os canibais. Boa sorte! <http://rachacuca.com.br/jogos/missionarios-e-canibais/>. Acesso em: 02 abr. 2015.



Faça você mesmo

Experimente fazer manualmente este passo-a-passo: Travessia	ida	volta	Ficam do lado A da ponte	Ficam do lado B da ponte
1ª	1G1 e 1G2	1G1	1G2	1G1, 2G1, 3G1/ 2G2, 3G2

2ª	2G2, 3G2	1G2	2G2, 3G2	1G1, 2G1, 3G1/1G2
3ª	1G1, 2G1	1G1/2G2	2G1/3G2	1G1, 3G1/ 1G2, 2G2
4ª	1G1, 3G1	3G2	1G1, 2G1, 3G1	1G2, 2G2, 3G2
5ª	1G2, 2G2	2G2	1G1, 2G1, 3G1/ 1G2	2G2, 3G2
6ª	2G2, 3G2	-	1G1, 2G1, 3G1/ 1G2, 2G2, 3G2	-

Faça valer a pena

1. De acordo com o material de estudos, complete as lacunas com as palavras correspondentes assinalando a alternativa correta.

i. _____: ordem e forma de apresentar as palavras para que sejam interpretadas.

ii. _____: paradigma de programação que segue a premissa de que contempla estruturas de seleção de informações, decisão e repetição.

iii. _____: paradigma de programação que aproxima o mundo real do virtual através da abstração dos dados, vinculação das informações através de herança, encapsulamento e polimorfismo.

- a. programação estruturada, programação estruturada e algoritmos.
- b. sintaxe, programação estruturada e programação orientada a objetos.
- c. algoritmos, sintaxe e programação orientada a objetos.
- d. sintaxe, algoritmos e programação estruturada.
- e. variáveis, sintaxe e algoritmos.

2. Cite e explique os dois tipos de problemas relacionados aos algoritmos e a forma como esses podem ser analisados.

3. Assinale "V" para verdadeiro e "F" para falso. São características dos algoritmos:

- a. () Finitude: indica que o algoritmo deve encerrar após um número finito de execução.
- b. () Longitude: que indica a capacidade de processamento do algoritmo desenvolvido.
- c. () Definição: evita definições que gerem ambiguidade.
- d. () Entrada: atribui valores ao algoritmo, a partir de especificações de variáveis e funções antes de sua iniciação.
- e. () Saída: representa os valores após o processamento, ou seja, a

execução das ações.

f. () Eficácia: todas as ações que o algoritmo realiza precisam ser executáveis em um limite de tempo predeterminado, finito, mesmo que seja sem o auxílio do computador.

4. A frase abaixo é referente a uma característica do desenvolvimento de qual conceito? Assinale a alternativa correta.

“No princípio designava a forma de resolver problemas matemáticos, depois princípios e teorias matemáticas, migrando finalmente para a área computacional, depois da invenção e consolidação dos computadores.” (PIVA Jr. et al., 2012)

- a. declaração de variáveis
- b. entrada de dados
- c. sistemas de informação
- d. algoritmos
- e. programação orientada a objetos

5. Quando e por quem os algoritmos foram introduzidos? Cite a lógica do algoritmo de Euclides.

6. Assinale a alternativa que representa o mecanismo de funcionamento de um algoritmo.

- a. dados, processamento e informação.
- b. dados, entrada e saída.
- c. entrada, saída e banco de dados.
- d. declaração, comandos e encerramento.
- e. informação, processamento e saída.

7. Desenvolva o algoritmo para calcular a área de uma mesa, solicitando ao usuário que insira os valores da base e da altura. Com base nessas informações e no desenvolvimento do algoritmo, assinale a alternativa correta que representa a lógica que deverá ser implantada.

- a. $A \leftarrow b * h$
- b. escreva $b*h$.
- c. leia (b,h)
- d. digite altura (h)
- e. leia (b)

Seção 1.2

Tipos de dados e expressões: literais, lógicas e aritméticas

Diálogo aberto

Antes de aprender como desenvolver programas de computador utilizando uma linguagem de programação, você será levado a conhecer as formas de apresentação dos elementos que devem ser considerados para efetuar as operações. Nessa área é muito comum que qualquer algoritmo elaborado apresente um contexto lógico-matemático, ou seja, necessite retornar o resultado de um cálculo, por exemplo. Porém, as expressões matemáticas não podem ser representadas nas linguagens computacionais com os mesmos símbolos a todo tempo e com a mesma forma. Precisam ser linearizadas e utilizar os operadores, que você provavelmente já conhece, mas que seguem uma padronização de acordo com a linguagem escolhida para o desenvolvimento.

Veja abaixo a tabela com os operadores matemáticos.

Tabela 1 | Principais operadores matemáticos em linguagem algorítmica

Operações	Operador
Adição	+
Subtração	-
Multiplicação	*
Divisão	/

Fonte: Piva Jr. (2012, p. 59).

Diante deste contexto, veja um exemplo de expressão matemática que precisa ser linearizada para que o computador possa interpretar essa informação: $y = \frac{2x}{5x+7} + 4x$.

De acordo com a linearização da fórmula acima, obtemos: $y \leftarrow ((2*x)/(5*x+7))+(4*x)$. Observe como a representação dessa expressão foi modificada.

A fim de apresentar uma solução em algoritmos, para os comerciantes do Litoral Sul, a sua missão nesta fase é definir as expressões matemáticas, lógicas e literais que serão utilizadas. Ao conjunto de operações é dado o nome de função. Defina, de acordo com o algoritmo apresentado anteriormente, quais serão e como devem ser implementadas. Além das expressões, identifique também quais são os tipos de dados e as respectivas variáveis ou constantes que devem ser declaradas para a realização das operações: “assim como os números, que em algum momento da história tiveram sua forma e utilização padronizadas, aconteceu a mesma coisa com as expressões matemáticas quando utilizadas em algoritmos ou linguagens de programação computacionais.” (PIVA Jr., 2012, p. 58).

Você pode utilizar, para os cálculos mais complexos, operadores e funções em linguagem algorítmica para as mais diversas operações que o aplicativo deverá executar. As expressões, os tipos de dados e variáveis aqui apresentados servem como referência para o desenvolvimento de *software* independente da linguagem ou paradigma de programação, no entanto, precisa se atentar a sintaxe de cada uma delas.

Não pode faltar

Alguns operadores matemáticos são utilizados para cálculos considerados simples mas, e se você precisar calcular, por exemplo, uma equação de segundo grau ou ainda uma integral, derivadas, logaritmos? Enfim, há uma infinidade de funções e suas aplicações são importantes! Vamos conhecer outros operadores matemáticos que auxiliam na elaboração de funções e cálculos mais complexos. Veja as tabelas abaixo e as respectivas descrições:", tal como no original.

Tabela 2 | Operações complexas com operador

Operações	Operador	Exemplo
Exponenciação	\wedge	$a \wedge b$
Divisão inteira	\backslash	$a \backslash b$ retorna o valor inteiro da divisão
Módulo	$\%$	$a \% b$ resto de divisão
Inversão de sinal	$-$	$-a$ ($-(-a)$ resulta em a)

Fonte: Piva Jr. (2012, p. 61).

Tabela 3 | Operações complexas com funções

Operações	Funções	Explicação
Raiz Quadrada	Raizq(x)	Raiz quadrada
Exponenciação	Exp(x,y)	x elevado a y
Valor absoluto	Abs(x)	Valor absoluto de x.
Arco Cosseno	ArcCos(x)	Ângulo em radianos cujo cosseno é representado por x.
Arco Seno	ArcSen(x)	Ângulo em radianos cujo seno é representado por x.
Arco Tangente	ArcTan(x)	Ângulo em radianos.
Cosseno	Cos(x)	Cosseno em radianos do ângulo x.
Cotangente	CoTan(x)	Retorna o ângulo x em radianos.
Parte inteira do número	Int(x)	Retorna a parte inteira do número x.
Logaritmo	Log(x)	Retorna o logaritmo de x na base 10.
Logaritmo Neperiano	LogN(x)	Retorna o logaritmo neperiano de x (base e).
Valor Pi(π)	Pi	Retorna o valor 3.141592.
Quadrado	Quad(x)	Retorna o quadrado de x.
Valor aleatório	Rand	Randômico entre 0 e 1.
Seno	Sen(x)	Retorna o seno do ângulo x em radianos.
Tangente	Tan(x)	Retorna a tangente do ângulo x em radianos.

Fonte: Piva Jr. (2012, p. 61- 62).



Assimile

Um bom exemplo a citar de linearização de funções pode ser representado pela equação de 2º grau. Observe: $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

Após a linearização, a função ficará da seguinte forma: $x \leftarrow (-b + \text{raizq}(b^2 - 4 * a * c)) / (2 * a).$

Além dos operadores matemáticos, conheça também os operadores relacionais. Veja a tabela a seguir.

Tabela 4 | Operadores relacionais

Operações	Operador	Exemplo
Igual	=	a = b (a é igual a b?)
Diferente	< >	a < > b (a é diferente de b?)
Maior que	>	a > b (a é maior que b?)
Menor que	<	a < b (a é menor que b?)
Maior ou igual	>=	a >= b (a é maior ou igual a b?)
Menor ou igual	<=	a <= (a é menor ou igual a b?)

Fonte: Piva Jr. (2012, p. 62-63).

Os operadores lógicos são muito importantes para a composição das expressões necessárias à execução dos programas. Observe na tabela abaixo quais são eles.

Tabela 5 | Operadores lógicos

Operações	Significado
Não	É unário na negação. Tem a maior precedência entre os operadores lógicos. não (VERDADEIRO)= FALSO, e não (FALSO)= VERDADEIRO
Ou	Resulta em verdadeiro quando um dos seus operandos lógicos for verdadeiro.
E	Operador que resulta VERDADEIRO apenas se seus dois operandos lógicos forem verdadeiros.
Xou	Operador que resulta VERDADEIRO se seus dois operandos lógicos forem diferentes, e FALSO se forem iguais.

Fonte: Piva Jr. (2012, p.63).

Para verificar se as proposições são verdadeiras ou falsas de acordo com o operador lógico que é utilizado, é aconselhável que você saiba como realizar o teste. Para tal, a sugestão é o uso da tabela verdade. Então, a considerar os operadores lógicos, confira a seguir os testes que podem ser realizados. Para a expressão: $A = B \text{ e } D$, observe:

B	D	$A = B \text{ e } D$
Falso	Falso	Falso
Falso	Verdadeiro	Falso
Verdadeiro	Falso	Falso
Verdadeiro	Verdadeiro	Verdadeiro

Exemplo: para que a condição seja verdadeira e atenda à solicitação do operador lógico, então, suponha que uma concessionária tenha uma meta por vendedor de 18 carros por mês e o mínimo de R\$ 58.000,00 de valor bruto de vendas. Se esse vendedor atingir a meta, então, ele receberá 10% de comissão. Se não, o vendedor recebe

apenas 0,08% do total como participação nas vendas. Seja A o valor da comissão, B o total de carros vendidos e D o valor mínimo de vendas, elabore a expressão lógica que atenda essa operação. **Resposta:** a expressão que representa essa operação é: $A = ((B \geq 18) \text{ e } (D \geq 58.000))$.

Agora, vamos fazer a tabela verdade para a expressão com o operador lógico "ou".

B	D	A = B ou D
Falso	Falso	Falso
Falso	Verdadeiro	Verdadeiro
Verdadeiro	Falso	Verdadeiro
Verdadeiro	Verdadeiro	Verdadeiro

Exemplo 2: para o mesmo problema apresentado acima, considere que o vendedor, para ganhar a comissão total, precisa atingir ou o valor da meta de vendas que é de 18 carros por mês, ou o valor mínimo em reais que é de R\$ 58.000,00. Elabore a expressão que representa essa condição. **Resposta:** a expressão que representa essa operação é: $A = ((B \geq 18) \text{ ou } (D \geq 58.000))$.

Observe a ação do operador lógico **XOU** ou, operador lógico exclusivo, é derivado das conjunções, disjunções inclusivas e negação. É preciso que **ao menos uma** das condições seja verdadeira para que resulte em verdadeiro. Veja a sua tabela verdade.

B	D	A = B xou D
Falso	Falso	Falso
Falso	Verdadeiro	Verdadeiro
Verdadeiro	Falso	Verdadeiro
Verdadeiro	Verdadeiro	Falso

Exemplo 3: Considere ainda o problema sugerido no exemplo 1 e elabore a expressão que a representa. A diferença desse operador está em aceitar ao menos uma das condições para que resulte em verdadeiro. **Resposta:** $A = ((B \geq 18) \text{ xou } (D \geq 58.000))$.

Abaixo a tabela verdade do operador lógico "não".

B	A = não B	A = B xou D
Falso	Verdadeiro	Falso
Verdadeiro	Falso	Verdadeiro

Exemplo 4: ainda considerando a situação-problema do exemplo 1, elabore a expressão que representa essa operação. **Resposta:** $A = (\text{nao}(B \geq 18) \text{ e } (\text{nao}(D \geq 58.000)))$. Os materiais de referência podem trazer o sinal "~" como indicação da notação para "não".



Pesquise mais

O [link](http://pichiliani.com.br/2013/04/visualizando-algoritmos/) abaixo contém informações sobre algoritmos e aplicações. Acesse e teste os exemplos disponíveis. Disponível em: <http://pichiliani.com.br/2013/04/visualizando-algoritmos/>. Acesso em: 25 mar. 2015.

Os operadores lógicos auxiliam a execução das operações, porém, estas afetam diretamente o valor contido no que se chama de “variável”. Por exemplo, nas expressões acima nós temos as variáveis A, B e D. Os operadores relacionais “>” maior e “=” igual e, os valores referentes a cada uma das variáveis. Note também que os valores das variáveis B e D não são alterados durante a execução, por esse motivo são chamadas de constantes. É importante que o valor da variável seja atribuído logo no momento de sua declaração.

Além desse aspecto, você precisa considerar a forma de como se faz para determinar o nome da variável, pois este será o seu identificador. São permitidos identificadores com letras maiúsculas, exemplo “COMISSAO_INTEGRAL”, no entanto, é recomendado que seja objetivo e transmita a informação que será ali armazenada ou manipulada, então, a sugestão é: “comissaoIntegral”.



Refleta

O espaço reservado previamente em memória, e devidamente rotulado com um nome (identificador), chamamos de variável. Caso o valor não venha a se alterar durante o programa, chamamos esse espaço de constante. (PIVA JR., 2012, p. 85).

- O espaço em branco também é um caractere especial.

As variáveis armazenam valores que são classificados quanto ao seu tipo e características. Veja na tabela abaixo os tipos de dados existentes.

Tipos de dados	Características	Exemplos
Inteiro	Contempla todos os números inteiros relativos (positivos, negativos e nulos).	12, -56, 852, 146698
Real	Conjunto dos números reais nulos, negativos e positivos.	23.5, 85.4, -354.8, -74, 3.141618

Caracter	Composta por todos os caracteres alfanuméricos e especiais.	0 – 9, A-Z, a-z, #,\$%,&,*@
Lógico ou booleanos	Assumem apenas a situação de verdadeiro ou falso.	0 ou 1, verdadeiro ou falso, sim ou não.

Fonte: Adaptado de Piva Jr. (2012, p. 86).

A seguir, há um exemplo para você praticar. Siga em frente.



Exemplificando

Observe o exemplo abaixo e as respectivas explicações.

Teste a transposição do algoritmo de soma de dois números (Scilab ou no Visualg-pseudocódigo) e observe como aparecem as entradas e saídas de dados e informação:

algoritmo "soma dois números"

// Função: algoritmo que exhibe o resultado da soma entre dois números.

// Autor : JJJ

// Data : 05/03/2015

// Seção de Declarações

var

x, y: inteiro

inicio

// Seção de Comandos

escreval("Digite o primeiro número: ")

leia(x)

escreval("Digite o segundo número: ")

leia(y)

escreva("A soma dos números é: ",x+y)

fimalgoritmo

As variáveis x e y são do tipo de dado "inteiro".

Dados de entrada: o comando "escreval" indica que o dado é **solicitado ou exibido** ao usuário, e o "l" que há a **quebra de linha**.

O comando "leia" indica que o dado é verificado na variável indicada.

Saída: soma (informação)



Faça você mesmo

Agora, observe o mesmo exemplo na transposição para uma linguagem de programação (C, implementada no Dev C++):

`/* Programa deverá apresentar a soma entre dois números*/`

`#include <stdio.h>`

Primeiramente é preciso estabelecer quais são as bibliotecas utilizadas por este programa em C.

`#include <stdlib.h>`

`main()`

Em seguida, há a declaração das variáveis, especificando o tipo de dado e a solicitação da entrada de dados.

`{ int a, b, soma;`

`printf("Informe o primeiro numero:\n");`

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

`printf("Informe o segundo numero:\n");`

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

A variável "soma" recebe o valor da adição das variáveis "a" com "b".

`soma = a + b;`

`printf("A soma dos numeros e: %d", soma);`

"%" sucedido da letra "d" representa a exibição do valor do dado do tipo inteiro contido na variável "soma".

`fflush (stdin);`

fflush: limpa o buffer

Os comandos **"printf"** e **"scanf"** são, respectivamente, o "escreva" e "leia" do exemplo anterior, porém, escritos em pseudocódigo.

Sem medo de errar

Aplicação dos procedimentos de atuação convenientes à SP

1. Observe que no caso apresentado para resolver a solicitação de desenvolvimento do aplicativo para os comerciantes do Litoral Sul, será proposto o algoritmo ao lado. O pseudocódigo apresenta inicialmente o nome do algoritmo:

"Litoral Sul".

2. Em seguida, traz a especificação da fase do projeto e o que o algoritmo apresenta.

3. Indicar o autor do algoritmo também é recomendável, bem como, a sua data de elaboração. Fatores que atribuem maior controle das alterações desse projeto inclusive.

4. Observe que antes de iniciar efetivamente o bloco de declaração de variáveis, há a indicação desse com a palavra "var", sintaxe válida para o VisualG, que visa demonstrar a sequência em que as operações precisam ser apresentadas nos algoritmos e consequentemente nos programas.

5. São declaradas três variáveis do tipo caracter, que corresponde ao tipo de dado char em outras linguagens. A variável média foi declarada com o tipo de dado real, o que em linguagem de programação significa *float*. As demais variáveis receberão valores do tipo inteiro, equivalente ao tipo de dado integer nas demais linguagens.

6. Observe que em seguida à declaração das variáveis há efetivamente o início das operações que o algoritmo deverá executar para atender a demanda solicitada.

7. Os comandos "leia" e "escreva" representam, respectivamente, entrada e saída de dados. Neste caso, houve a necessidade de uso de uma estrutura de seleção, que será melhor descrita a seguir. Nesse caso, ela serve para inserir o teste da condição que foi imposta e que, a partir desta escolha do usuário, será possível verificar e seguir adiante com as execuções previstas.

• Pseudocódigo

algoritmo "Litoral Sul"

// Função : representação do algoritmo de ações do aplicativo- Fase 1.

// Autor : JJJ

// Data : 12/03/2015

// Seção de Declarações

var

login, agenda, email: caractere

média: real

satisfação, satisfeito, insatisfeito, categoria, g, h: inteiro

inicio

// Seção de Comandos

escreva("Clique no ícone login para acessar o sistema.")

leia(login)

escreva("Digite 1 para escolher serviços de gastronomia ou 2 para hotelaria.")

leia(categoria)

se categoria = 1 então

$g \leftarrow g + 1$

senão

$h \leftarrow h + 1$

fimse

média $\leftarrow (g + h)/2$

escreva("A quantidade média de acessos é de:", média)

escreva("A quantidade de acessos para gastronomia é de:", g)

escreva("A quantidade de acessos para hotelaria é de:", h)

fimalgoritmo



Atenção!

Outro ponto de atenção, aqui, é para a atribuição de valores nas variáveis, que não têm característica de constante justamente porque podem sofrer alterações durante a execução do programa, de acordo com a opção escolhida.



Lembre-se

Você tem total autonomia para melhorar os processos e incrementar as funcionalidades, então, fique de olho no cronograma desse projeto 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 as compare com a de seus colegas e com o gabarito disponibilizado no apêndice do livro.	
Tipos de dados e expressões: literais, lógicas e aritméticas	
1. Competência de fundamento 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 o que são, como se aplicam e a quem se destina a elaboração dos algoritmos; • Conhecer como ocorre o raciocínio lógico computacional.
3. Conteúdos relacionados	<ul style="list-style-type: none"> • Expressões literais, lógicas e aritméticas; • Tipos de dados, variáveis e constantes.
4. Descrição da SP	Nas turmas da 6ª série da escola "Estudar Faz Bem", a professora estava reforçando com os alunos as operações matemáticas. Para integrar as tecnologias de informação e comunicação nas aulas para essa turma, a professora solicitou o desenvolvimento de um aplicativo simples que permitisse aos alunos testar em uma calculadora amigável e interativa tais operações. O intuito é fazer com que os alunos, desde as primeiras etapas de seu ensino,
4. Descrição da SP	compreendam a lógica computacional e, dessa forma, realizem a prova real manuscrita e também visualizem outras possibilidades de se realizar essas operações. Sua missão é desenvolver o algoritmo que apresente as operações adição, subtração, multiplicação e divisão, e de forma que a interface apresentada seja fácil para esse público compreender.
5. Resolução da SP	Faça você mesmo: algoritmo "Operações matemáticas" // Função : Elabore um algoritmo que receba dois números e a partir desses, exiba os respectivos resultados com as operações matemáticas adição, subtração, multiplicação e divisão. // Seção de Declarações var x, y: real início // Seção de Comandos escreva("Digite o primeiro número: ") leia(x) escreva("Digite o segundo número: ") leia(y) escreval("A soma é: ",x+y) escreval("A subtracao é: ",x-y) escreval("A multiplicacao é: ",x*y) escreval("A divisao é: ",x/y) finalgoritmo



Lembre-se

"A definição de um tipo de dado para uma determinada variável indica a reserva de um determinado espaço em memória. Caso esse espaço seja insuficiente, ocorrerá um erro de **time mismatch** ou **overflow**, levando à interrupção da execução do programa." (PIVA JR., 2012, p. 101)

Faça valer a pena

1. Elabore um algoritmo que permita ao lojista inserir o valor do produto e o valor do desconto. Em seguida, apresente o valor do produto e o seu valor com o desconto. Assinale a alternativa que indica corretamente a expressão para este cálculo.

- a. `desconto <- v_produto*pdesconto/100`
- b. `desconto <- desc/100 - valor_prod`
- c. `v_produto<- desconto * desconto/100`
- d. `desconto <- v_produto* desconto/0,1`
- e. `desconto <- v_produto* desconto/1,0`

2. Desenvolva um algoritmo que receba o valor de um ângulo em graus e exiba o seu valor em radianos: apresente o pseudocódigo e, em linguagem C.

3. Elabore um algoritmo para calcular o consumo de combustível de um carro. Assinale a alternativa correta, após a análise da linha de comando abaixo.

Escreval ("O consumo de combustível do seu veículo é: ",distancia/combustivel)

- a. não é possível realizar esta operação.
- b. a sintaxe está incorreta.
- c. a sintaxe está correta, pois apresenta o valor médio de combustível que o tanque do carro comporta.
- d. o comando está correto, pois é possível efetuar uma operação matemática seguida da exibição da mensagem.
- e. está incorreta, pois não existe o comando escreval.

4. Elabore um algoritmo que realize a cotação do dólar.

5. Em um processo de precificação, o empreendedor precisa de uma ferramenta que o auxilie a estabelecer a sua margem de lucro e a calculá-la de forma mais rápida e facilitada. Para tal, desenvolva um algoritmo que receba o valor do produto, o valor da margem de lucro desejada em percentual, e exiba o valor final. Assinale a alternativa que representa as variáveis que são necessárias para a realização desse algoritmo e o seu respectivo tipo de dado.

- a. preço1, preçoTotal e preçoLíquido: caractere.
- b. valorProduto, margemLucro, valorFinal: real.
- c. valorProduto, margemLucro, valorFinal: inteiro.
- d. valorProduto, margemLucro, valorFinal: caractere.
- e. preço1, preçoTotal e preçoLíquido: inteiro.

6. Dadas as sequências de tipos de dados abaixo, assinale a alternativa correta.

- a. inteiro, real e caractere.
- b. real, strong e integer.
- c. char, íntegro e real.
- d. strong, íntegro e short.

7. Assinale a alternativa que contém apenas os operadores relacionais.

- a. and, or, xor
- b. >, <, >=, <=
- c. e, ou, xou
- d. *, /, -, +
- e. >, /, <=, *

Seção 1.3

Representação de algoritmos e o ambiente de programação

Diálogo aberto

No cenário descrito no "Convite ao Estudo", a empresa de desenvolvimento "Think Now" precisa considerar inicialmente:

a. Os processos que o aplicativo deverá executar e como é o seu algoritmo. Além de buscar informações de soluções e algoritmos existentes que possam servir de parâmetro para esse desenvolvimento.

b. O ambiente de programação e como se dá a interpretação das ações para a linguagem de computador;

c. A lógica existente entre os processos que o aplicativo deverá executar e a sua respectiva ilustração com a elaboração do fluxograma;

d. A competência técnica da equipe de desenvolvimento para a identificação das expressões literais, lógicas e aritméticas, além de saber quando deve ser utilizada uma ou outra;

e. Além disso, será necessário que a equipe já identifique os tipos de dados que serão utilizados, o que será variável ou constante e inicie o seu processo de declaração, em ambiente computacional.

É preciso estabelecer qual será o passo-a-passo para realizar as consultas e reservas, e de forma que seja simples, rápido e fácil ao usuário final. A empresa também precisa identificar quais são os riscos que o projeto pode enfrentar para que seja concluído, ou seja, entregue.

Sendo assim, é possível inferir que devem ser apresentados:

- Para o passo-a-passo, além da observação do processo inicialmente informado, será preciso interpretar e identificar se todos os envolvidos compreendem e aprovam as etapas definidas;

- Elaborar o fluxograma de ações do *software*, ou seja, a representação do seu algoritmo é componente fundamental desse processo;
- A equipe responsável pelo desenvolvimento do projeto deverá alinhar as solicitações de funcionalidades do aplicativo que realmente são passíveis de realização dentro do prazo estipulado.

Além desses, você, que faz parte do quadro de colaboradores da empresa dedicados a esse projeto, deverá considerar, para o desenvolvimento de suas atividades, todos os tópicos de “a” a “e”, descritos na situação acima, e auxiliar a sua equipe na resolução desta tarefa. Bom trabalho!

Não pode faltar

Considere que os algoritmos, de forma geral, visam estabelecer a sequência lógica existente entre as ações que devem ser executadas. Para tal, símbolos podem ser utilizados de forma que representem a informação desejada. Veja um exemplo de como pode ser representado um processo a partir do uso de operadores lógicos (conjunção “e”, disjunção ou condicional “ou”), nesse caso, é descrito como exemplo o operador “e” (PIVA JR., 2012, p. 9):

p: $1 + 1 = 2$

r: $2 + 1 = 3$

q: p e r

Na representação acima, o “q” é uma proposição verdadeira resultante de “p” e “r”, que também o são. Essa lógica nada mais é do que uma representação lógica formal ou simbólica.

Observe o exemplo de um algoritmo que define o cálculo da média dos alunos, a considerar duas notas a obter um *status* de aprovação ou reprovação, caso a média seja inferior a 6,0:

Inserir nota 1.

Inserir nota 2.

Calcular a média sendo que: $(\text{nota } 1 + \text{nota } 2)/2$.

Exibir a média.

Verificar se a média é menor do que “6,0”.

Se sim, indicar "aluno reprovado".

Se não, indicar "aluno aprovado".

Outro exemplo, um algoritmo de um programa que calcule a soma de dois números (mesma lógica do primeiro exemplo):

$a \leftarrow 1$

$b \leftarrow 2$

$c \leftarrow a + b$

$c = 3$



Refleta

"Saber as fases e as estratégias de resolução de um problema auxilia a construir novas ferramentas mentais para compreender e resolver novos problemas. Quanto mais praticar, mais ferramentas terá e mais problemas conseguirá resolver." (PIVA JR., 2012, p. 33)

A seguir, o exemplo de cálculo da média em C (Dev C++):

```
/* Programa deverá apresentar a média de duas notas*/
#include <stdio.h>
#include <stdlib.h>
main()
{
    float a, b, media;
    printf("Informe duas notas:\n");
    scanf("%f%f%f", &a, &b, &c);
    media = (a + b + c)/3;
    printf("A media e: %f, %f e %f igual a %f", a, b, c, media);
    fflush (stdin);
    getch();
    return 0;}

```

As prioridades de execução respeitam as propriedades matemáticas.

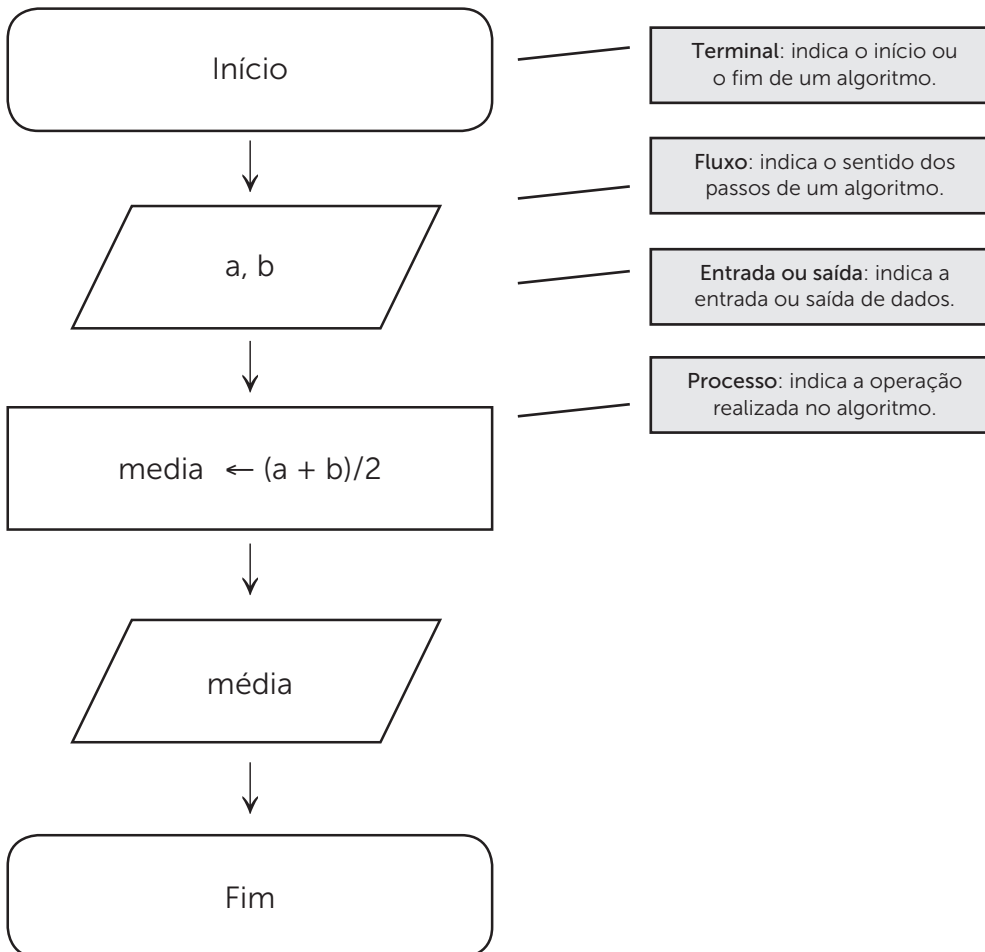


Pesquise mais

No site C Programming, você encontrará mais informações sobre a linguagem C. Disponível em: <<http://www.cprogramming.com/>>. Acesso em: 15 mai. 2015.

É possível a utilização de símbolos e também de uma linguagem não computacional, como no caso do algoritmo de cálculo da média. Nesse último caso, as entradas de dados são as "nota 1" e "nota 2". Processamento é o cálculo efetivamente; e a média é a saída que se obtém do processamento.

Outra forma de representar um algoritmo é através do seu fluxograma. O exemplo será do algoritmo elaborado para realizar a soma de dois números. Veja abaixo como fazer e o que significa cada símbolo utilizado.



Analise o fluxograma e identifique que há a necessidade da declaração das variáveis, seguido da indicação da operação que será realizada, ou seja, o processamento e, logo após, há a apresentação do resultado desse processamento, uma saída, portanto, e o encerramento do processo do algoritmo, que representou, nesse caso, a soma de dois números.



Assimile

Representação do algoritmo pode ser:

- Algoritmo não computacional: é escrito em linguagem natural.
- Algoritmo computacional: é escrito em pseudocódigo ou português estruturado ou em uma linguagem de programação, ou seja, implementado.
- Fluxograma.

Para facilitar a sua compreensão acerca do que é algoritmo, e como ele pode auxiliar na resolução de problemas, tenha em mente os passos:

1. Identifique o problema;
2. Imagine como pode resolver;
3. Analise as soluções e escolha a que for mais viável;
4. Reavalie se necessário.

Vamos exercitar mais um pouco o conteúdo. Suponha que em uma partida de vôlei, o sistema registra os pontos a partir do lançamento manual, ou seja, realizado pelo juiz ou assistente da partida, que atualiza os valores ponto a ponto. Nesse caso, para estabelecer qual é o ganhador, o sistema faz a contagem dos pontos do jogador A e do jogador B, o que tiver a maior quantidade de pontos é o vencedor. Faça o algoritmo em linguagem natural, em pseudocódigo e o seu fluxograma.



Exemplificando

Algoritmo em linguagem natural:

1. Identifique os jogadores;
2. Lance os pontos do jogador A;

3. Lance os pontos do jogador B;
4. Verifique a maior quantidade de pontos;
5. Se jogador A com maior quantidade de pontos, então, Jogador A é o campeão, senão, Jogador B;
6. Encerra a partida.

Algoritmo em pseudocódigo (Visualg):

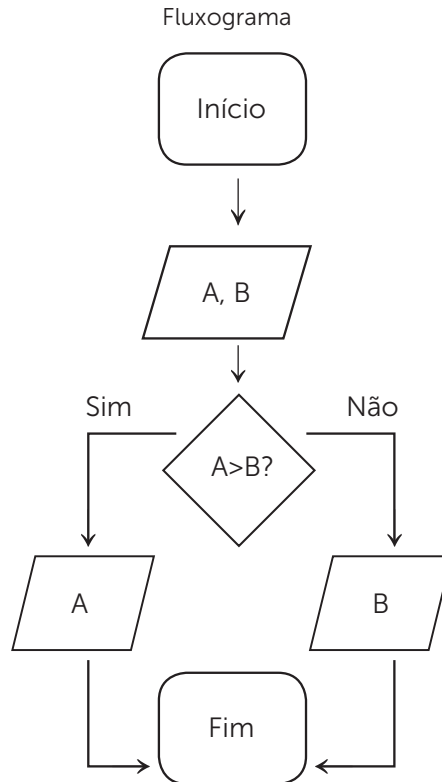
```

algoritmo "Verificar valor maior ou menor"
    // Função: Cálculo de pontos: maior ou menor
    // Autor: "Nós"
    // Data: 06/01/2015
    // Seção de Declarações
    var
        A, B: inteiro
    inicio
        // Seção de Comandos
        escreval("Informe os pontos do jogador A: ")
        leia (A)
        escreval ("Informe os pontos do jogador B: ")
        leia (B)
        se A > B entao
            escreva ("O campeão é o jogador A:", A)
        senao
            escreva ("O campeão é o jogador B:", B)
        fimse
    fimalgoritmo

```

**Faça você mesmo**

Agora tente você mesmo fazer o diagrama de bloco ou fluxograma do algoritmo acima.

**Sem medo de errar**

Agora é o momento de iniciar a resolução do projeto que a Think Now assumiu, e você está responsável por mapear os processos que o aplicativo deverá contemplar. Você e sua equipe, ao se reunirem com os comerciantes, identificaram que a maior preocupação deles estava em identificar de forma contínua, ou seja, desde a primeira fase da implantação, qual é a percepção do cliente em relação ao aplicativo, sua aceitação e usabilidade. Sugeriram um processo inicialmente mais simples, porém, que já retornasse a média de pessoas que acessaram. Então, a sua tarefa agora é mapear os processos para o desenvolvimento e apresentar o algoritmo desse aplicativo em linguagem natural, fluxograma e pseudocódigo. Então, mãos à obra.

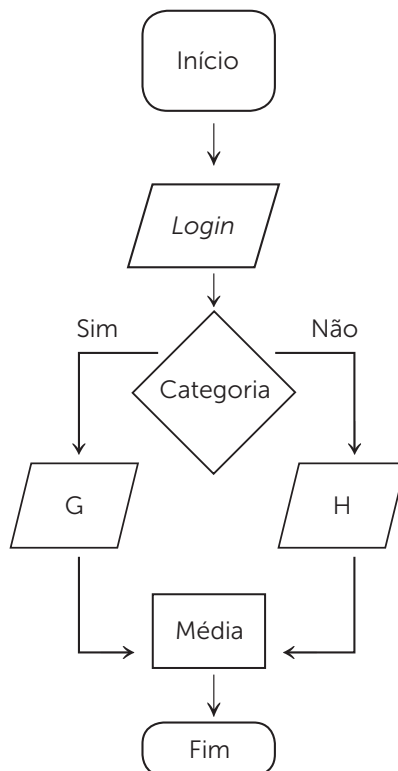
**Atenção!**

Para compreender os processos, é importante interpretar a solicitação e, com isso, escrever o passo-a-passo para a execução sem deixar de contemplar os processos-chave dessa operação.

A seguir, uma sugestão de transposição dos dados levantados, em **linguagem natural**.

1. Início
2. O aplicativo deverá permitir o login por cadastro ou por rede social.
3. Após logar, o usuário será direcionado a um processo para escolher a categoria: gastronomia ou hotel.
4. Aplicativo contabiliza acessos de gastronomia e hotelaria e gera a média.
5. Fim.

A definição dos processos inicialmente identificados, representados em **fluxograma**:





Lembre-se

Os fluxogramas precisam apresentar as ações do sistema, os processos, as entradas e saídas de dados. Você pode, inclusive, inserir as estruturas de seleção, repetição e decisão na representação do algoritmo. Pesquise mais na bibliografia básica e resolva os exercícios para praticar todas as formas de representação dos processos. Norma ISO 5807/1985.

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 as compare com a de seus colegas e com o gabarito disponibilizado no apêndice do livro.	
Representação de algoritmos e o ambiente de programação	
1. Competência de fundamento 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 o que são, como se aplicam e a quem se destina a elaboração dos algoritmos; • Conhecer o ambiente de programação e como se dá o raciocínio lógico computacional.
3. Conteúdos relacionados	<ul style="list-style-type: none"> • O ambiente de programação; • Formas de representação de algoritmos;
4. Descrição da SP	Suponha que no restaurante francês JacquesBistrô, além do valor do salário, os garçons não podem receber gorjetas dos clientes. Então, há uma calculadora especial que está programada para executar essa operação. Assim que o garçom encerra a conta no sistema, a calculadora solicita que ele digite dois números inteiros menores que 20. A comissão será calculada pela soma dos dois números e de duas formas distintas. Veja a fórmula $c1 = v1 + v2/100$ e a outra fórmula é: $c2 = (v1+v2)/100$. Para encerrar, o sistema calcula a soma das duas operações e exibe o valor que será pago ao garçom como comissão. Faça o algoritmo dessa calculadora em pseudocódigo e em linguagem C.
5. Resolução da SP	<pre> algoritmo "JacquesBistrô" var n1,n2: inteiro c1,c2,comissaoFinal: real inicio // Seção de Comandos escreva("1º Número= ") leia(n1) escreva("2º Número= ") leia(n2) c1 <- n1 + n2 / 100 </pre>

5. Resolução da SP

```
c2 <- (n1 + n2)/100
comissaoFinal<- c1 +c2
escreval("Comissão final= ", comissaoFinal:7:2)
finalgoritmo
```

**Lembre-se**

O título do algoritmo deve ser o mais objetivo possível, expressando exatamente a função realizada. O espaço reservado para declaração de variáveis vem antes do início dos processos de execução em VisuAlg.

**Faça você mesmo**

```
#include <stdio.h>

#include <conio.h>

void main()

{ int n1, n2;

  float c1,c2, cf;

  printf("n1 = "); scanf ("%d", &n1);

  printf("n2 = "); scanf ("%d", &n2);

  c1 = n1 + n2 / 100;

  c2 = (n1 + n2)/100;cf= c1 + c2;printf("\n cf = %7.2f", cf);getch();}
```

Faça valer a pena

1. Uma loja de artes e decoração, de porte pequeno, está precisando de um módulo de cálculo que indique, a partir do valor total da compra, o valor das prestações e parcela em até cinco vezes. Assinale a alternativa que representa o cálculo correto a ser inserido.

- a. prestação <- compra/5
- b. leia (compra)
- c. compra = prestação/5
- d. escreva(compra <- prestação)
- e. leia (prestação = compra/5)

2. Seguindo o raciocínio lógico requerido, analise o algoritmo abaixo e assinale a alternativa que melhor representa o seu processamento.

```
var
distancia, combustivel: real
inicio
// Seção de Comandos
escreval("* Cálculo de consumo de combustível *")
escreva("Informe a distância a percorrer: ")
leia(distancia)
escreva("Informe a quantidade (em litros) de combustível gasto: ")
leia(combustivel)
escreval("O consumo de combustível será de: ",distancia/combustivel)
finalgoritmo
```

- a. o algoritmo acima calcula o valor que se pagará ao completar o tanque de combustível.
- b. o algoritmo acima calcula a quantidade de consumo de combustível do automóvel.
- c. o algoritmo representa a seção de comandos para calcular o valor do combustível.
- d. o algoritmo representa a situação real usada para estabelecer os preços de mercado do combustível.
- e. o algoritmo informa simplesmente quais são as variáveis utilizadas para o cálculo do combustível.

3. Em uma empresa de automóveis, os vendedores, além do seu salário fixo, recebem uma comissão de 15% a cada venda realizada. A empresa precisa de um programa que, a partir da entrada do nome do vendedor, mostre o seu salário fixo e o valor total de vendas que ele fez no período. O valor da comissão se aplica ao valor total de vendas. Elabore o algoritmo equivalente a esta operação.

4. Considere duas variáveis, X e Y. Desenvolva um algoritmo que realize a troca dos valores das variáveis, ou seja, a variável X recebe o valor da variável Y e vice-versa e, ao final, o algoritmo deve apresentar também os valores trocados.

5. Analise a expressão: "v_real <- v_dolar/cotação" e assinale a alternativa que substituiria sem perdas o resultado se estivesse escrito na linguagem de programação C.

- a. `v_rs← v_us/cot;`
- b. `v_real == v_dolar/cotação`
- c. `v_real = v_dolar/cotação;`
- d. `real = dólar/5.`

6. Assinale a alternativa que indica o comando de entrada em C.

- a. `var`
- b. `get`
- c. `printf`
- d. `scanf`
- e. `fflush`

7. Assinale a alternativa que indica o comando de saída em C.

- a. `main()`
- b. `stdin.h`
- c. `set`
- d. `scanf`
- e. `printf`

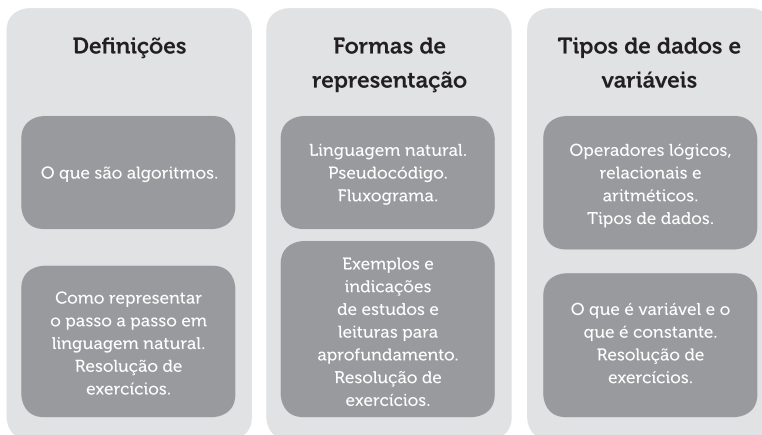
Seção 1.4

Declaração de variáveis e constantes

Diálogo aberto

A fim de sistematizar o conteúdo visto até o momento, nesta seção você pode identificar todos os pontos de atenção que precisam ser trabalhados para que seja possível a elaboração de algoritmos, ou seja, ampliar o seu contato com as informações dos tópicos anteriores e reforçar com os exercícios. Então, veja abaixo um quadro esquemático da abordagem conceitual realizada até o momento. Vamos lá.

Quadro 1 | Esquema de conteúdos estudados



Fonte: O autor (2015).

Leve em consideração que a primeira fase do projeto que os comerciantes do Litoral Sul solicitaram já está com as diretrizes e funcionalidades esclarecidas. A partir de agora, cabe à equipe da Think Now elaborar o algoritmo final que contemple a identificação dos processos e de todas as variáveis, constantes e os seus respectivos tipos de dados bem definidos e mapeados. Essa representação também deverá contemplar a elaboração do seu fluxograma.

Lembre-se de que a sequência a seguir retoma a importância da sua compreensão

acerca dos processos que o aplicativo executará para a elaboração do protótipo em algoritmos:

1. Entenda o processo através do levantamento de requisitos. Pergunte-se: o que os comerciantes precisam e o que solicitaram? Em que tempo?
2. Elabore protótipo do aplicativo e apresente o seu algoritmo em linguagem natural; se preciso for, revise o material e utilize as dicas!
3. Identifique quais são as variáveis necessárias e o seu respectivo tipo de dado e elabore uma tabela com as especificações. Boas práticas.

Não pode faltar

Caro aluno, vamos em frente com a apresentação de situações que podem auxiliar no desenvolvimento da solução para a entrega do aplicativo. Vamos rever alguns conteúdos e ver o quanto conseguimos progredir com as práticas.

Primeiramente, vimos como se deu a concepção de algoritmos e citamos o de Euclides, como uma forma de manter as propriedades matemáticas da operação envolvida e, ainda, estabelecer a sequência de execução. Veja abaixo o exemplo em pseudocódigo.



Exemplificando

//seção de declarações

var

a, b, mdc, resto: inteiro;

//seção de comandos, lembre-se de que você pode solicitar ao usuário para inserir os dados de a e b (escreva e leia)

início

resto ← a mod b; //esta operação indica que o comando "mod" fará a divisão de "a" por "b" e apresentará o resto de divisão, que será atribuído à variável "resto".

enquanto resto ≠ 0 faça //o comando de repetição enquanto, representado por while em linguagem c, indica que o programa fará uma sequência de

repetições enquanto a condição determinada for atendida.

$a \leftarrow b$;

$b \leftarrow \text{resto}$;

$\text{resto} \leftarrow a \bmod b$;

fimenquanto //toda estrutura iniciada, seja de seleção, repetição ou mesmo de decisão, deverá ser finalizada

$\text{mdc} \leftarrow b$;

fimalgoritmo.

Note que no algoritmo apresentado acima, há uma estrutura de repetição sobre a qual você poderá aprofundar os estudos nas próximas unidades, porém, esta faz uma verificação do resto de divisão, e enquanto este for diferente de "0" (zero), implica na sequência de divisões imposta pelo algoritmo. Observe e tente você mesmo fazer a implementação do algoritmo em linguagem C.



Faça você mesmo

```
#include <stdio.h>

int main( void )

{int a,b,resto,mdc, dividendo, divisor;

printf( "Numero a: "); scanf( "%d", &a);
printf( "Numero b: "); scanf( "%d", &b);

dividendo = a;
divisor = b;

while ( resto !=0 ){

resto = dividendo % divisor;

dividendo = divisor;
divisor = resto; };

mdc = dividendo;

printf( "mdc [%d, %d] = %d ", a, b, mdc); return 0;}
```

Observe o uso dos operadores relacionais e sua importância na execução dos algoritmos, bem como, dos operadores lógicos. As expressões matemáticas representadas pelas variáveis indicam claramente como acontece a alocação de valores e como o computador interpreta essas informações para que se possa imprimir, ou exibir, o resultado da operação.

Você também pode notar que, de acordo com o ambiente de programação em que se encontra, há uma forma específica de representação das variáveis necessárias ao programa. Por exemplo, em Visualg, os tipos de dados comuns são: caractere, lógico, inteiro e real. Veja a tabela a seguir com as características de cada um deles.

Tabela 1 | Tipos de dados em VisuAlg

Tipos de dados em VisuAlg	Descrição	Tamanho
Caractere	Corresponde a um caractere qualquer.	1 byte
Lógico	Apresenta apenas valores para verdadeiro ou falso, sim ou não.	1 byte
Inteiro	Compreende os números inteiros do intervalo, inclusive: -32768 a +32767.	2 bytes
Real	Compreende todos os números de 2.9×10^{-39} a 1.7×10^{38} .	6 bytes

Fonte: Adaptado de Piva Jr. (2012, p. 90).

Uma característica interessante é que em VisuAlg a declaração das variáveis precede a seção de comandos e é indicada pela palavra “var”. Abaixo são apresentados os tipos de dados em linguagem C. A diferença, nesse sentido, é que na linguagem de programação C, as variáveis podem ser declaradas em qualquer ponto do programa, e também é possível atribuir um valor a ela. No entanto, há duas recomendações que, por convenção, são sugeridas quanto à declaração das variáveis: antes das declarações de funções, ou seja, como variáveis globais; ou, logo no início do código da própria função, o que faz com que a variável seja local.

Tabela 2 | Tipos de dados em C

Tipos de dados em C	Descrição	Tamanho
Char	Compreende de -127 a 126 caracteres.	1 byte
int	Compreende de -32768 a +32767.	2 bytes
Float	Compreende de $3.4\text{E}-38$ a $3.4\text{E}+38$	4 bytes
Double	Compreende de $1.7\text{E}-308$ a $1.7\text{E}308$	8 bytes

Fonte: Adaptado de Piva Jr. (2012, p. 93).

Fonte: Adaptado de Piva Jr. (2012, p. 90).

O equivalente ao tipo de dado "real" em linguagem C são os tipos de dados "float" e também "double". O seu uso depende do tamanho do dado e de que forma deverá ser manipulado pelo programa. Outros tipos de dados em C também podem ser usados, tais como os que seguem na Tabela 3.

Tabela 3 | Tipos de dados em C e em linguagens comerciais

Tipos de dados em C	Descrição	Tamanho
signed char	-128 a 127	1 byte
unsigned char	0 a 255	1 byte
signed int	-32768 a 32767	2 bytes
unsigned int	0 a 65535	2 bytes
short int	-32768 a 32767	2 bytes
long int	-2.147.483.648 a 2.147.483.647	4 bytes
long double	3.4xE-4932 a 1.1xE4932	10 bytes

Fonte: Adaptado de Piva Jr. (2012, p. 93).

Retomando os conceitos, temos então a sequência apresentada para o desenvolvimento de algoritmos em que há, além da descrição do nome, seguido da declaração das variáveis, o início do algoritmo efetivamente. Veja mais um exemplo.

Em uma empresa de desenvolvimento de *software* há um módulo que realiza a cotação e autoriza a compra. O algoritmo dessa operação, que é realizada manualmente a partir do lançamento pelo usuário, considera dois orçamentos para a tomada de decisões. A regra é adotar o menor valor. Acompanhe a resolução abaixo:

```
//algoritmo: autorização para compra de software
```

```
//seção de declaração
```

```
var
```

```
software1, software2 : real
```

```
//seção de declaração
```

```
inicio
```

```
    escreval("Informe o valor do software 1:")
```

```
    leia (software1)
```

```
    escreval("Informe o valor do software 2:")
```

```
    leia (software2)
```

```
    se (software 1 > software 2) entao
```


`escreval("Autorização de compra para o software 2 liberada!")`

`senao`

`escreval("Autorização de compra para o software 1 liberada!")`

`fimse`

`fimalgoritmo`



Refleta

“Uma variável em um fluxograma representa uma área na memória onde se pode armazenar um valor”. (SOUZA et al., 2011, p. 75).

Por enquanto, a abordagem focou na apresentação das formas de representação dos algoritmos, a linearização de expressões matemáticas, a importância dos operadores aritméticos, relacionais e lógicos. Porém, além desses, você também precisa utilizar as cadeias de caracteres, ou seja, aprender algumas funções de como realizar as atribuições de texto às variáveis e visualizar esse resultado.

A seguir, estão especificadas algumas funções utilizadas para que seja facilitada a manipulação de informações textuais, também chamadas de **expressões literais**. Também será comum ouvir a palavra de referência **string** para identificar o tipo de dado de que trata uma cadeia de caracteres.

Tabela 4 | Funções e procedimentos para as cadeias de caracteres

Operação	Significado	Exemplos
<code>length(s)</code>	Fornece como resultado o número de caracteres que compõem uma cadeia S.	$N \leftarrow \text{length}(\text{'Olá'})$ O valor N é 3.
<code>concat(S1,S2)</code>	Une duas cadeias, a 2ª (S2) no final da 1ª (S1), formando uma nova cadeia.	$S \leftarrow \text{concat}(\text{'Bom'}, \text{'Dia'})$ O valor de S é 'BomDia'.
<code>copy(S, ini, num)</code>	Retorna (copia) a uma nova cadeia de caracteres com os num elementos da cadeia S a partir da posição ini.	$S \leftarrow \text{copy}(\text{'Turbo Pascal'}, 76)$ O valor de S é 'Pascal'.
<code>insert(S1, S2, ini)</code>	Insere uma nova cadeia (S1) na posição ini da S2, deslocando para a esquerda o resto da cadeia original (S2).	$S \leftarrow \text{'Turbo 70'}$ $\text{insert}(\text{'Pascal'}, S, 7)$ O valor de S é 'Turbo Pascal 70'

$pos(S1, S2)$	Fornece como resultado a posição, na qual a cadeia S1 começa dentro da cadeia S2. Se a cadeia S1 não existir em S2, o resultado será zero.	$I \leftarrow pos('Pascal', 'Turbo Pascal')$ Aqui o valor de I é 7, mas $I \leftarrow pos('pascal', 'Turbo Pascal')$ O valor de I é 0. Existe diferença entre maiúsculas e minúsculas nessa função.
---------------	--	--

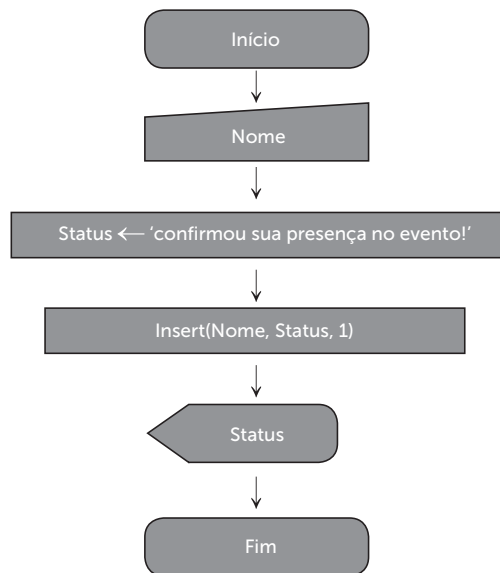
Fonte: Souza et al. (2011, p. 104).

Note que os caracteres em algoritmos, por convenção, são acompanhados de apóstrofes. Cada caractere corresponde a uma única letra ou símbolo. Essa forma de representação ('x') evita que um caractere seja confundido com o nome da variável x, por exemplo. Encontra-se nessa categoria de representação também o espaço em branco (' '). Por convenção, o tamanho da cadeia de caracteres é limitado a 255. Você sabia que um nome de variável pode ter no máximo 63 caracteres? Pois bem, identifique as variáveis e constantes com nomes objetivos que realmente representem ou remetam à informação que será atribuída.



Assimile

Observe a representação no fluxograma de uma sequência lógica para a manipulação de uma cadeia de caracteres que informe um Nome e exiba o Status: "Valter confirmou presença no evento".



Além da função *insert*, você pode testar os resultados obtidos com o uso das demais funções. Utilize a tabela acima e este fluxograma como

exemplos, e inicie os testes com outras variáveis, cenários e objetivos. Tais funções podem ser muito úteis no desenvolvimento do protótipo da aplicação que você apresentará junto à Think Now.



Pesquise mais

Leia o artigo do site Tecmundo, e veja os sete principais algoritmos que foram selecionados e cabem no contexto descrito. Disponível em: <http://www.tecmundo.com.br/tecnologia/56148-forca-invisivel-7-tipos-algoritmos-dominam-nosso-mundo.htm>. Acesso em: 04/03/2015.

Sem medo de errar

A proposta abaixo oferecida pela Think Now, com a especificação das ações iniciais que o aplicativo deverá executar, está descrita no algoritmo em linguagem natural.

Algoritmo de ações iniciais do aplicativo

1. Início
2. Cadastrar estabelecimentos comerciais.
3. Logar usuário por cadastro em formulário próprio ou por redes sociais.
4. Solicitar que o usuário selecione: categoria (gastronomia ou hotelaria), cidade.
5. Exibir de acordo com a seleção, os estabelecimentos cadastrados.
6. Disponibilizar a opção ao usuário para “conhecer” ou “agendar”.
7. Se o usuário simplesmente optar por conhecer, então,
 - 7a. Direcionar para o website do estabelecimento.
8. Senão
 - 8a. Exibir calendário e horários disponíveis.
 - 8b. Solicitar o preenchimento dos campos pelo usuário: local, data e hora.
 - 8c. Exibir a mensagem de confirmação: “Aguarde o contato de confirmação de reserva”.
 - 8d. Abrir a pesquisa de satisfação.

- 8e. Solicitar a resposta do usuário.
- 8f. Calcular a média de acessos e respostas à pesquisa por estabelecimento cadastrado.
- 8g. Enviar mensagem para o e-mail do estabelecimento com cópia ao usuário final com a solicitação de reserva e a resposta da pesquisa.
9. Encerra estrutura de seleção.
10. Retornar à página inicial.



Atenção!

A partir dessas definições, é possível identificar quais serão as variáveis necessárias para a sua implementação. Fazemos o exercício de identificação das variáveis que deverão ser declaradas, para que os processos abaixo identificados sejam realizados.



Lembre-se

Variáveis	Tipo de dado em VisuAlg	Descrição
local, data, hora, G, H, msg_confirmacao (constante)	caractere	G – gastronomia H- hotelaria
satisfacao, conhecer, agenda	lógico	Variáveis aceitam 1 para sim e 0 para não
cont_satisfacao, qtd_sim, qtd_nao	inteiro	A cada indicação de satisfeito ('1') ou insatisfeito ('0') contar e armazenar
Media	real	Efetua o cálculo da média de acessos

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 as compare com a de seus colegas e com o gabarito disponibilizado no apêndice do livro.

Declaração de variáveis e constantes	
1. Competência de fundamento 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 o que são, como se aplicam e a quem se destina a elaboração dos algoritmos; • Conhecer como ocorre o raciocínio lógico computacional.
3. Conteúdos relacionados	<ul style="list-style-type: none"> • Tipos de dados, variáveis e constantes; • Declaração de variáveis e constantes;
4. Descrição da SP	Um triatleta, em fase de readaptação, está treinando para competir nas Olimpíadas. Sabendo que a prova pode ter no máximo 51,5 km, ele precisa atingir bom desempenho para realizar os 1,5 km de natação, 40 km de ciclismo e 10 km de corrida. O atleta busca um dispositivo que indique os tempos ideais para superar cada fase com tranquilidade e melhorar os seus tempos. Faça um algoritmo que receba um número, que indica a quantidade de minutos mínima para cumprir a prova e, também, considere o intervalo, que seria a indicação do tempo ideal. O algoritmo deve verificar se esse número está no intervalo entre 1 e 20 min.
5. Resolução da SP	<ul style="list-style-type: none"> • Pseudocódigo <pre> algoritmo "Cálculo deTempo" // Função : Faça um algoritmo que receba um número que indica a quantidade de minutos mínima para cumprir a prova e, também, considere o intervalo, que seria a indicação do tempo ideal. O algoritmo deve verificar se esse número está no intervalo entre 1 e 20 min.// Autor :xxx // Data : xxx // Seção de Declarações var numero: inteiro inicio // Seção de Comandos escreva("Digite um número: ") leia(numero) se numero >= 1 entao se numero <= 20 entao escreva("O número está no intervalo entre 1 e 20") senao escreva("O número não está no intervalo entre 1 e 20") fimse senao escreva("O número não está no intervalo entre 1 e 20") fimse finalgoritmo </pre>



Lembre-se

Todo algoritmo precisa ser definido como exemplificado no exercício ao lado. Note que há as seções de: declaração de variáveis; de inserção dos

comandos e funções. É preciso indicar o início e o fim do algoritmo. A palavra descrita como "var" indica as variáveis que serão utilizadas para realizar as operações e essas necessariamente são declaradas. Os comandos "leia" e "escreva" representam, respectivamente, as entradas e saídas de dados no algoritmo.



Faça você mesmo

- em linguagem C

```
#include<stdio.h>

main(){

int num;

printf("Informe um numero: ");

scanf("%d",&num);

if ((num > 1) && ( num < 20)){

    printf("O valor: %d, está no intervalo compreendido entre 1 e 20.", num);
}

else{

    printf("Informe valor que esteja no intervalo de 1 a 20.");  }

fflush(stdin);getch(); }
```

Observe, no código-fonte acima, o uso do operador lógico "**&&**", que indica que a condição é inclusiva (e), ou seja, precisa atender às duas condições impostas na operação.

Você também já pode se familiarizar com os operadores relacionais: (**>**) maior, (**<**) menor, (**>=**) maior ou igual, (**<=**) menor ou igual.

Faça valer a pena

1. Elabore o fluxograma do algoritmo do protótipo a ser desenvolvido do aplicativo para os comerciantes do Litoral Sul.

2. Complete a tabela com os respectivos tipos de dados que precisam ser associados aos conceitos. Assinale a alternativa correta.

Tipos de dados em VisuAlg	Descrição
	Corresponde a um caractere qualquer.
	Apresenta apenas valores para verdadeiro ou falso, sim ou não.
	Compreende os números inteiros do intervalo, inclusive:
	-32768 a +32767.
	Compreende todos os números de 2.9×10^{-39} a 1.7×10^{38} .

- a. real, lógico, inteiro e caractere.
- b. Lógico, real, integer, e char.
- c. real, char, int, integer.
- d. string, booleano, inteiro e real.
- e. caractere, lógico, inteiro e real.

3. Escreva um algoritmo que armazene o valor informado pelo usuário em uma variável A e um outro valor em uma variável B. O algoritmo precisa apresentar a lógica necessária para realizar a inversão dos valores da variável A para B e de B para A. Por fim, exiba o valor final das variáveis.

4. Para a expressão "correcao < - 1.2 * deposito", identifique qual é o melhor tipo de dado a ser relacionado a estas variáveis. Assinale a alternativa correta.

- a. inteiro
- b. real
- c. long
- d. lógico
- e. caractere

5. Com base no exercício 3 desta lista, elabore o fluxograma do algoritmo proposto e assinale a alternativa que melhor representa as variáveis utilizadas nesse algoritmo.

- a. menos, mais e altera.
- b. qtd1, qtd2 e qtd3.

- c. a, b e aux
- d. raiz1, raiz2 e delta
- e. num1, num2 e num3.

6. A considerar ano com 365 dias e mês com 30 dias, elabore um algoritmo que leia a idade de uma pessoa e escreva a sua idade em dias e em meses. A expressão "escreval("A sua idade em dias é de:", dias, " dias")", apresenta qual identificação direta de variável?

- a. real
- b. dias
- c. dia
- d. inteiro
- e. idade

7. Elabore um algoritmo que exiba as unidades, dezenas e centenas de um número qualquer fornecido pelo usuário. Analise o algoritmo e assinale a alternativa correta.

- a. não é possível extrair o resto de divisão do número quando o tipo de dado for inteiro.
- b. mod é um comando que exibe o resto de divisão.
- c. div exibe resto de divisão.
- d. mod e div não são utilizados em operações com tipos de dados inteiros.
- e. não é possível exibir unidades, dezenas e centenas de um número

Referências

EVARISTO, Jaime. **Aprendendo a programar programando na linguagem C: para iniciantes**. 3. ed. Maceió: Instituto de Computação Universidade Federal de Alagoas, 2002.

PIVA JUNIOR, Dilermando et al. **Algoritmos e programação de computadores**. Rio de Janeiro: Campus, 2012.

ZIVIANI, Nivio. **Projeto de Algoritmos: com implementações em Java e C++**. São Paulo: Cengage Learning, 2011.

SOUZA, Marco Antonio Furlan de et al. **Algoritmos e lógica de programação**. 2. ed. Cengage, 2011.

Referência(s) bibliográfica(s) complementare(s):

1. EVARISTO, Jaime. **Aprendendo a Programar: programando na linguagem C para iniciantes**. Rio de Janeiro: Ed. Book Express, 2001.

2. SOUZA, Marco. **Algoritmos e lógica de programação**. 2ª ed. São Paulo: Cengage Learning, 2011.

3. MANZANO, José A. N. G. **Algoritmos: Lógica para Desenvolvimento de Programação de Computadores**. 24ª ed. São Paulo: Érica, 2010.

4. AGUILAR, Luis J. **Programação em C++**. 2. ed. Porto Alegre. McGraw Hill, 2008.

5. SZWARCFITER, Jayme Luiz; MARKEZON, Lilian. **Estruturas de dados e seus algoritmos**. 3ª ed. Rio de Janeiro: LTC – Livros Técnicos e Científicos, 2010.