

Respostas dos Exercícios

Lógica de Programação **Conhecendo Algoritmos e Criando Programas**

Daniel Hayashida Simão
Wellington José dos Reis



editora
VIENA

1ª Edição
Bauru/SP
Editora Viena
2015

Capítulo 1

1. Fazer um sanduíche

Exemplo de Algoritmo:

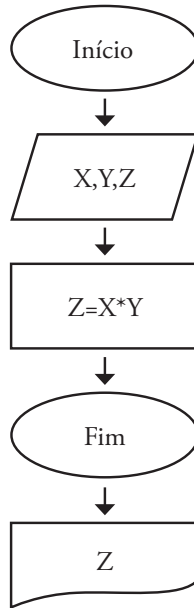
1. Pegar o pão.
2. Cortar o pão ao meio Pegar a maionese.
3. Algoritmos 3.
4. Passar a maionese no pão.
5. Pegar e cortar alface e tomate.
6. Colocar alface e tomate no pão.
7. Pegar o Hambúrguer.
8. Fritar o Hambúrguer.
9. Colocar o Hambúrguer no pão.

1. Trocar Pneu

Exemplo de Algoritmo:

1. Desparafusar a roda.
2. Pegar o pneu estepe.
3. Suspender o carro com um macaco.
4. Retirar a roda com o pneu furado.
5. Colocar o pneu estepe.
6. Abaixar o carro.
7. Parafusar a roda.

Capítulo 2

2.

Capítulo 3

1. Assinale os dados que são do tipo inteiro:

- a) 042
- d) 4000
- f) -85

2. Assinale os dados que são do tipo caractere:

- a) "11,6"
- b) -75
- c) "media"
- d) Verdadeiro
- e) 0,56
- f) -23,5983
- g) "Par OU Impar"
- h) Falso

3. Assinale os nomes válidos para variáveis:

- a) lnome
- b) RUA
- c) (Y)
- d) “ano”
- e) &resposta
- f) KM/H
- g) data_aniversario
- h) V52
- i) valor\$

4. Assinale os dados que são do tipo real:

- a) “73,2”
- b) -833
- c) “dez”
- d) Verdadeiro
- e) 0,56
- f) 78,12579
- g) “numero”

5. Assinale os dados que são do tipo lógico:

- a) “76978”
- b) Verdadeiro
- c) “Rua 7 de setembro”
- d) 4569
- e) 0,056
- f) -3,687
- g) “preço \$”
- h) Falso

Capítulo 4

1.

- V.
- F.
- V.
- V.

F.
F.
V.
V.
V.
V.
F.
F.
F.

2.

Nome do programa: Media

Variáveis:

```
real nota1;  
real nota2;  
real = media;
```

Início:

```
media = (nota1+nota2)/2;  
exiba (media);
```

Fim

Capítulo 5

1.

Nome do programa: Verifica_idade

Variáveis:

```
inteiro idade;
```

Início:

```
se idade >= 18 então;  
exiba (“maior de idade”);  
senão  
exiba (“menor de idade”);
```

Fim

2.

Nome do programa: Cardápio

Variáveis:

inteiro pratoAlimento;

Início:

selecione (diadasemana)

caso 1:

exiba (“feiojada”)

caso 2:

exiba (“panquecas”)

caso 3:

exiba (“pastel”)

caso 4:

exiba (“lasanha”)

caso 5:

exiba (“esfiha”)

caso senão:

(“prato indisponível!”)

Fim

3. Exibir números pares de 0 a 100:

Variáveis

Inteiro par;

Início

enquanto par de 0 até 100 faça

se (par%2)=0 então

exiba (par)

Fim

Capítulo 6

1. Exibir um vetor de 400 posições

Variáveis:

inteiro nome [400]

inteiro i

Início

```
para (i=0;i<=100;i++)  
  faça  
    exiba ("digite o nome do aluno")  
    leia nome[i]  
  para (i=0;i<=100;i++)  
    faça  
      exiba nome[i]
```

Fim

2. Criar Vetor de caractere que armazene 15 valores

Variáveis

Inteiro nome[1, 5, 10, 15]

Inteiro a, b, c, d

Início

```
para (a=0; a<=1; a++)  
  faça  
    exiba ("digite o nome do aluno: ")  
    leia nome[a]  
  para (b=0; b<=1; b++)  
    faça  
      exiba ("digite o nome do aluno: ")  
      leia nome[b]  
  para (c=0; c<=10; c++)  
    faça  
      exiba ("digite o nome do aluno: ")  
      leia nome[c]  
  para (d=0; d<=15; d++)  
    faça  
      exiba ("digite o nome do aluno: ")  
      leia nome[d]
```

3. Faça um programa que leia uma matriz A de duas dimensões

Variáveis:

inteiro mat [3,6]

inteiro a

inteiro b

Início

para (a=1; a<=3; a++) faça

para (b=1; b<=6; b++) faça

leia mat [3,6]

para (a=1; a<=3; a++) faça

para (b=1; b<=6; b++) faça

exiba (mat [a,b])

Fim

4. Ler duas matrizes com 20 linhas e 15 colunas e calcular a soma de seus elementos.

Variáveis:

Inteiro mat1 [20,15], mat2 [20,15], soma;

Início:

leia mat1;

leia mat2;

soma = mat1+mat2;

exiba (“a soma das matrizes será”, soma);

Fim

5. Faça um programa que leia os dados de entrada para um registro

Variáveis:

registro cadastro

caractere nome

caractere endereço

inteiro telefone

caractere email

real salário

fim registro

Início

```
leia cadastro.nome
exiba cadastro.nome
leia cadastro.endereco
exiba cadastro.endereco
leia cadastro.telefone
exiba cadastro.telefone
leia cadastro.email
exiba cadastro.email
leia cadastro.salario
exiba cadastro.salario
```

Fim

6. Crie um conjunto de registros que armazene 20 funcionários, onde cada registro deverá ficar dentro de um vetor.

Tipo:

```
registro colaboradores
caractere nome
caractere endereço
inteiro telefone
caractere email
real salário
fim registro
```

Variáveis:

```
cadastro funcionario [20]
```

Início

```
leia cadastro.nome
exiba cadastro.nome
leia cadastro.endereco
exiba cadastro.endereco
leia cadastro.telefone
exiba cadastro.telefone
leia cadastro.email
exiba cadastro.email
leia cadastro.salario
exiba cadastro.salario
```

Fim

Capítulo 7

1. Desenvolva um programa que leia um número qualquer e chame um procedimento que verifica se ele é positivo ou negativo.

procedimento verifica (real numero)

Variáveis
real numero

Início
 exiba (informe um número)
 leia numero
 se numero > 0
 faça
 exiba (“numero positivo”)
 senão
 exiba (“numero negativo”)
Fim

2. Elabore um programa que leia um número positivo qualquer e chame um procedimento que calcule e mostre o fatorial do valor lido.

procedimento fatorial (inteiro, num, i, fatorial)

Início
 exiba (“digite um número positivo”)
 leia (num)
 fatorial < -1
 para i de 1 ate num faça
 fatorial <-1 fatorial*1
 exiba (“o fatorial é:”, fatorial)
Fim

3. Elabore um programa que leia dois valores e chame uma sub-rotina que calcule a multiplicação entre eles e retorne o resultado

procedimento calculo (real num1, num2, mult)

Início

 mult = num1*num2

Fim

Início

 exiba (“digite o primeiro numero”)

 leia (num1)

 exiba (“digite o segundo numero”)

 leia (num2)

 exiba (“o resultado da multiplicação e: ”, mult)

Fim

4. Crie um programa que crie a conversão de reais em dólar. O usuário deve informar o valor da cotação do dólar e a quantidade de reais que deseja converter

Variáveis

real c, n

Início

 exiba (“informe o valor da cotação em dólar ”);

 leia (c)

 exiba (“informe a quantidade de reais que deseja converter”)

 leia (n)

Fim

real função conversao

Início

 conversao = c*n

 exiba (“o valor em dolar e ”, conversao)

Fim

Capítulo 8

1. Escreva o que você entende sobre fila no contexto da programação.

No contexto da programação, as filas seguem o mesmo conceito das filas da vida real, como as filas que enfrentamos em bancos, escolas, mercados, entre outras situações. Ou seja, o primeiro elemento a entrar na fila será o primeiro a sair ou o que está a mais tempo será o primeiro a sair.

2. Escreva o que você entende sobre pilha de programação.

No contexto da programação, o conceito de pilhas é semelhante ao de uma pilha de papéis, onde o último que entra é o primeiro que sai, pois, por exemplo, quando se coloca uma pilha de papéis sobre uma mesa, o papel que toca a mesa é o primeiro papel da fila e o papel da superfície é o último da fila, mas será o primeiro a sair para uso.

3. Qual a principal diferença entre o conceito de pilha e fila?

Pilha e fila possuem propostas de funcionamento inversos, sendo que nas filas o primeiro elemento a entrar é o primeiro a sair e nas pilhas, o último a entrar é o primeiro a sair.

4. Analise a árvore a seguir e responda as seguintes questões:

a) Quais os nós terminais?

F, I.

b) Qual é a altura da árvore?

A árvore possui altura 5.

c) Qual é o grau de cada nó?

Grau A: 8 / Grau B: 3/ Grau C: 3/ Grau D: 2/ Grau E: 1/ Grau F: 0/Grau G: 0
Grau H: 1/ Grau I: 0

d) Qual o grau da árvore?

Grau 2

Capítulo 10

1. Criar as classes Escola, Uniforme e Aluno. Definir todos os atributos e métodos dessas classes. Após isso criar 1 objeto escola, 1 objeto uniforme e 2 objetos alunos

Classe Escola:

Atributos: Nome, endereço, nível de ensino, ano fundação.

Objeto Escola	
Nome	E.E Getulio Vargas
Endereço	Rua dos Limoeiros
Nível de Ensino	Ed. Básica
Ano Fundação	1958

Classe Uniforme

Atributos: Cor, tamanho, tecido.

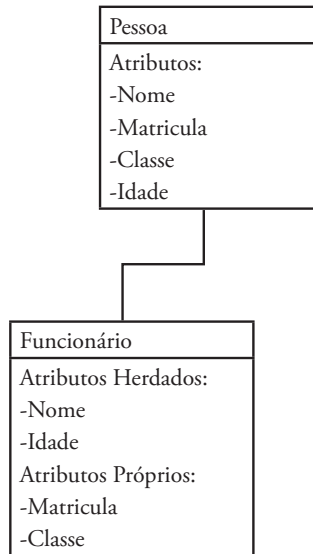
Objeto Uniforme	
Cor	Azul
Tamanho	M
Tecido	Algodão

Classe Aluno

Atributos: Nome, matricula, classe, idade

Objeto Aluno	
Nome	Alan da Silva
Matricula	2145-6
Classe	1ª Série
Idade	7 Anos

2. Desenvolver uma representação de herança simples



3. Assinale abaixo a alternativa que melhor defina o principal objetivo do Encapsulamento.

Alternativa B.

4. De acordo com a representação, responda:

- As classes Designer, Diagramador e Revisor são especializações da classe funcionário, pois todo colaborador no caso tratado, todo colaborador será um funcionário.
- As classes Designer, Diagramador e Revisor são generalizações da classe funcionário, pois as 3 categorias se tratam de um funcionário.

5. Desenvolver um relacionamento de classes do tipo Associação utilizando os atributos de ligação

