

Curso – Tecnologia em Sistemas para Internet

Disciplina – Lógica de Programação

Prof. Nelio Alves – www.nelioalves.com

Lista de Exercícios 5 – Modularização (funções)

PARTE 1 – Funções que não retornam valor (procedimentos)

Parte 1.1	Variáveis: (x) variáveis “simples” () vetores
	Instruções: (x) sim () não

1) Fazer um programa para imprimir na tela os nomes e cargas horárias das disciplinas do primeiro semestre do seu curso. Antes de imprimir na tela estes dados, imprimir também um cabeçalho com seus dados acadêmicos tais como seu nome, número de matrícula, ano, turno (este programa é meramente informativo – não será preciso entrar com nenhum valor).

Instruções:

- Implemente as seguintes funções:

```
/**
 * Imprime na tela um cabeçalho com seus dados acadêmicos tais como
 * seu nome, número de matrícula, ano, turno
 */
public static void mostrarCabeçalho()

/**
 * Imprime na tela os nomes e cargas horárias das disciplinas do
 * primeiro semestre do seu curso
 */
public static void mostrarDisciplinas()
```

2) Fazer um programa para ler a quantidade de glicose no sangue de uma pessoa e depois mostrar na tela a classificação desta glicose de acordo com a tabela de referência ao lado.

Classificação	Glicose
Normal	Até 100 mg/dl
Elevado	Maior que 100 até 140 mg/dl
Diabetes	Maior de 140 mg/dl

Instruções:

- Implemente uma função que atenda à seguinte especificação:

```
/**
 * Imprime na tela a classificação da glicose recebida, de acordo com
 * a seguinte tabela de referência: (1) Até 100mg/dl = Normal;
 * (2) Maior que 100 até 140mg/dl = Elevado; (3) Maior que 140mg/dl = Diabetes
 * @param g Medida de glicose em mg/dl
 */
public static void classificarGlicose(double g)
```

3) Fazer um programa para ler os dados de um financiamento de imóvel (valor do imóvel, valor de entrada, número de prestações financiadas). Em seguida, mostrar na tela o valor de cada prestação, sem considerar juros.

- Implemente a seguinte função:

```
/**
 * Imprime na tela o valor de cada prestação do financiamento, sem considerar juros
 * @param va Valor do imóvel
```

```

* @param ve Valor da entrada
* @param p Quantidade de prestações
*/
public static void mostrarValorDaPrestacao(double vi, double ve, int p)

```

Parte 1.2	Variáveis: (x) variáveis “simples” () vetores
	Instruções: () sim (x) não

- 4) Fazer um programa para solicitar do usuário um número inteiro N. Depois, imprimir na tela a frase “Online ou offline?” N vezes.
- 5) Fazer um programa para imprimir na tela a tabuada de multiplicação de 1 a 10 de um certo número N fornecido pelo usuário.
- 6) Fazer um programa para ler os dados do resultado de um exame de sangue (glicose, triglicerídeos e colesterol). Em seguida, mostrar na tela um relatório com as classificações para este exame de acordo com as tabelas de referência abaixo.

Classificação	Glicose
Normal	Até 100 mg/dl
Elevado	Maior que 100 até 140 mg/dl
Diabetes	Maior de 140 mg/dl

Classificação	Triglicerídios (mg/dl)
Desejável	Até 200 mg/dl
Aumentado	Maior de 200 mg/dl

Classificação	Colesterol (mg/dl)
Desejável	Até 200 mg/dl
Limiar	Maior que 200 até 240 mg/dl
Elevado	Maior que 240 mg/dl

- 7) Fazer um programa para classificar o peso de uma pessoa, dados seu peso e altura. A classificação é baseada no IMC (Índice de Massa Corporal) e as faixas de classificação são dadas abaixo:

$$IMC = \frac{PESO}{ALTURA^2}$$

Classificações:

IMC	Classificação
abaixo de 20	“abaixo do peso”
20 até 25 exclusive	“peso normal”
25 até 30 exclusive	“sobre peso”
30 em diante	“obeso”

Parte 1.3	Variáveis: () variáveis “simples” (x) vetores
	Instruções: (x) sim () não

- 8) Fazer um programa para ler um vetor de N números reais a partir do teclado. Em seguida, imprimir apenas os números positivos do vetor lido.

Instruções:

- Implemente as seguintes funções:

```

/**
 * Faz a leitura do vetor inteiro a partir da entrada padrao
 * @param vet Um vetor JA INSTANCIADO
 */
public static void lerVetor(double[] vet)

/**
 * Imprime na tela somente os valores positivos do vetor
 * @param vet Um vetor ja preenchido com numeros
 */
public static void imprimirPositivos(double[] vet)

```

- 9) Fazer um programa para ler um vetor de N números reais a partir do teclado. Somar cinco unidades a cada um dos elementos do vetor e imprimir o vetor assim modificado na tela.

Instruções:

- Reproveite a função **lerVetor**

- Implemente as seguintes funções:

```
/**
 * Altera o conteudo do vetor dado, somando o numero x a cada um
 * dos elementos do vetor
 * @param vet Um vetor ja preenchido com numeros
 * @param x Um valor numerico qualquer
 */
public static void somarValor(double[] vet, double x)

/**
 * Imprime o vetor dado na saida padrao
 * @param vet Um vetor ja preenchido com numeros
 */
public static void imprimirVetor(double[] vet)
```

Parte 1.4	Variáveis: () variáveis “simples” (x) vetores Instruções: () sim (x) não
------------------	---

10) Fazer um programa para ler três vetores A, B e C de números reais de N elementos cada um. Em seguida, gerar um vetor D onde cada elemento de D é a média aritmética dos elementos correspondentes de A, B e C. Imprimir o vetor D.

11) Fazer um programa para ler um vetor de N números inteiros e, em seguida, mostrar na tela apenas os números pares deste vetor, depois imprimir na tela apenas os números negativos.

12) Fazer um programa para ler os nomes e idades de N pessoas, armazenando estes dados em dois vetores. Em seguida, mostrar na tela os nomes das pessoas que possuem entre 12 e 20 anos (inclusive).

PARTE 2 – Funções que retornam valor

Parte 2.1	Variáveis: (x) variáveis “simples” () vetores Instruções: (x) sim () não
------------------	---

13) Fazer um programa para ler o peso e altura de uma pessoa. Em seguida, mostrar o valor do Índice de Massa Corporal (IMC) dessa pessoa.

Instruções:

- Implemente uma função que atenda à seguinte especificação:

```
/**
 * Calcula o IMC de uma pessoa
 * @param peso Uma medida de peso em kg
 * @param altura Uma medida de altura em metros
 * @return O IMC calculado
 */
public static double imc(double peso, double altura)
```

- Fórmula: $IMC = \frac{PESO}{ALTURA^2}$

14) Fazer um programa que solicite um valor de temperatura medida em Fahrenheit. Em seguida, mostrar na tela a temperatura convertida para Celsius.

Instruções

- Implemente uma função que atenda à seguinte especificação:

```
/**
 * Converte uma temperatura de Fahrenheit para Celsius
 * @param F Uma medida de temperatura em Fahrenheit
 * @return A temperatura correspondente em Celsius
 */
public static double fahrenheitParaCelsius(double F)
```

- Fórmula: $C = \frac{5}{9}(F - 32)$

15) Fazer um programa que faça a leitura do preço do dólar e, em seguida, peça para o usuário digitar uma quantidade de dinheiro em reais e mostre na tela quantos dólares é possível comprar com a quantidade de reais informada.

Instruções

- Implemente uma função chamada **realParaDolar** que atenda à seguinte especificação:

```
/**
 * Realiza a conversao de real para dolar
 * @param reais Um valor em reais
 * @param cambio O valor da cotacao do dolar
 * @return O valor correspondente em dolar
 */
public static double realParaDolar(double reais, double cambio)
```

- Dica: para resolver este cálculo, basta fazer uma regra de três

16) Fazer um programa para ler os valores de quatro números N1, N2, N3 e N4. Calcule e imprima o valor da média aritmética dos mesmos.

Instruções:

- Implemente uma função que atenda à seguinte especificação:

```
/**
 * Calcula a media aritmetica entre quatro numeros
 * @param N1 Primeiro numero
 * @param N2 Segundo numero
 * @param N3 Terceiro numero
 * @param N4 Quarto numero
 * @return A media aritmetica dos numeros
 */
public static double mediaEntreQuatro(double N1, double N2, double N3, double N4)
```

Parte2.2	Variáveis: (x) variáveis “simples” () vetores
	Instruções: () sim (x) não

17) Em uma escola, a nota final anual é dividida em três trimestres, sendo que cada trimestre vale 25, porém a nota do último trimestre tem o dobro do peso das duas primeiras (perceba que a nota final vale 100). Fazer um programa para ler as três notas obtidas durante os três trimestres do ano. Em seguida, mostrar na tela a nota final do estudante, bem como uma mensagem se ele foi aprovado ou reprovado. O estudante é reprovado quando sua nota final for inferior a 70.

18) Fazer um programa para mostrar na tela o maior dentre quatro números fornecidos pelo usuário (em caso de empate, basta mostrar uma vez o maior).

19) Fazer um programa para ler as medidas (base e altura) de um retângulo. Em seguida, mostrar na tela os valores da diagonal, da área e do perímetro do retângulo.

20) A operadora de energia de uma cidade aplica, para cada casa, o valor básico de R\$ 20,00, para um consumo mensal de até 45 Kwh. Para cada Kwh excedente é cobrado o valor de R\$ 0,50. Fazer um programa para ler a quantidade de Kwh consumida no mês por uma casa e depois mostrar o valor da conta de energia.

Parte 2.3	Variáveis: () variáveis “simples” (x) vetores
	Instruções: (x) sim () não

21) Fazer um programa para ler um vetor de números reais de N elementos e, em seguida, mostrar a soma, a média dos elementos desse vetor, bem como mostrar a posição do maior elemento deste vetor.

Instruções:

- Implemente as seguintes funções:

```
/**
 * Faz a leitura de um vetor inteiro a partir da entrada padrao
 * @param N A quantidade desejada de elementos a serem lidos
 * @return O vetor contendo os valores lidos
 */
public static double[] nextVetor(int N)

/**
 * Calcula a soma dos valores de um vetor de numeros
 * @param vet Um vetor ja preenchido com numeros
 * @return A soma dos valores
 */
public static double soma(double[] vet)

/**
 * Calcula a media aritmetica dos valores de um vetor de numeros
 * @param vet Um vetor ja preenchido com numeros
 * @return A media aritmetica dos valores
 */
public static double media(double[] vet)

/**
 * Encontra o maior elemento de um vetor de numeros
 * @param vet Um vetor ja preenchido com numeros
 * @return O maior elemento do vetor
 */
public static double maiorElemento(double[] vet)
```

22) Fazer um programa para ler um vetor de N números reais. Depois, solicitar que o usuário forneça um número real a ser procurado no vetor. Caso o número seja encontrado na posição P, imprimir na tela: “Número encontrado na posição P”. Caso contrário, imprimir na tela: “Número não encontrado”.

Instruções:

- Reaproveite a função **lerVetor** ou **nextVetor**
- Implemente a seguinte função:

```
/**
 * Realiza a busca de um elemento x no vetor
 * @param vet Um vetor ja preenchido
 * @param x Um valor a ser procurado
 * @return A primeira ocorrencia de x no vetor. Se o valor x nao
 * existir, retorna -1
 */
public static int busca(double[] vet, double x)
```

Parte 2.4

Variáveis:	() variáveis “simples” (x) vetores
Instruções:	() sim (x) não

23) Fazer um programa para ler um vetor de números reais de N elementos e, em seguida, mostrar a média aritmética somente dos números positivos do vetor. Mostrar também a quantidade de números negativos do vetor.

24) Fazer um programa para ler os nomes e idades de N pessoas, armazenando estes dados em dois vetores. Em seguida, mostrar na tela o nome da pessoa mais velha. Mostrar também a percentagem de pessoas menores de 18 anos em relação ao total de pessoas.