

Nome:	Turma:
Professor(a): Graça Marietto	Data:

Lista de Exercícios

Objetivo

Implementação de programas na linguagem Java com vetor e módulo.

Exercícios

- 1) Escreva um programa na linguagem de programação Java que, no método `main()`, peça ao usuário que digite o tamanho de um vetor a ser criado. No método `main()` deve ser criado um vetor com o tamanho informado pelo usuário, mas ainda sem nenhum valor inserido pelo usuário.

Este vetor criado deve ser passado como parâmetro para o método `lerVetor()`, que permitirá que o usuário digite cada número do vetor. O método `lerVetor()` deve retornar ao método `main()` o vetor atualizado, ou seja, com os valores inseridos pelo usuário. Em seguida, no método `main()` o vetor atualizado deve ser passado como parâmetro para o método `encontrarMaiorElemento()`. Este método irá encontrar o maior elemento do vetor recebido como parâmetro, e depois deve retornar o maior elemento encontrado. No método `main()`, o maior elemento encontrado deve ser impresso na tela.

Logo após esta implementação, o método `main()` deve passar o vetor atualizado para o método `ordenarVetor()`. Este método deve colocar em ordem crescente o vetor recebido (do menor elemento para o maior elemento), e deve retornar o vetor já em ordem crescente. No método `main()` o vetor recebido, já em ordem crescente, deve ser impresso na tela do computador.

```
import java.util.Scanner;

public class MaiorElementoVetor
{
    public static void main(String args[])
    {
        //Declaração e inicialização de variáveis
        double numeros[];
        double maiorElemento=0;
        int tamanho=0;
        Scanner entradaNum = new Scanner( System.in );

        //Entrada de dados
        System.out.print("Digite o tamanho do vetor a ser lido: ");
        tamanho = entradaNum.nextInt();
        numeros = new double[tamanho];
        numeros = lerVetor(numeros);

        maiorElemento=encontrarMaiorElemento(numeros);
    }
}
```

```
        imprimirLinha();
        System.out.println("Maior elemento = " + maiorElemento);

        double vetorOrdenado[] = new double[tamanho];
        vetorOrdenado=ordenarVetor(numeros);
        System.out.println("Vetor ordenado em ordem crescente");
        imprimirVetor(vetorOrdenado);
    }//Fim do método main()

    public static double[] lerVetor(double[] num)
    {
        Scanner ent = new Scanner(System.in);

        imprimirLinha();
        System.out.println("DIGITE OS VALORES DOS ELEMENTOS DO VETOR");

        for (int i=0; i<num.length; i++)
        {
            System.out.print("Digite o valor de num["+i+"]= ");
            num[i]=ent.nextDouble();
        }
        return num;
    }//Fim do método lerVetor()

    public static double encontrarMaiorElemento(double[] vetor)
    {
        double maior=Double.NEGATIVE_INFINITY;
        for (int i=0; i<vetor.length; i++)
        {
            if (vetor[i]>maior)
                maior = vetor[i];
        }
        return maior;
    }//Fim do método encontrarMaiorElemento()

    public static double[] ordenarVetor(double[] vet)
    {
        for (int i=0; i<vet.length-1; i++)
        {
            for (int k=1;k<vet.length-i; k++)
            {
                if (vet[k-1]>vet[k])
                {
                    double aux;
                    aux=vet[k-1];
                    vet[k-1]=vet[k];
                    vet[k]=aux;
                }
            }
        }
        return vet;
    }//Fim do método ordenarVetor()
```

```
public static void imprimirVetor(double[] valores)
{
    imprimirLinha();
    for (int i=0; i<valores.length; i++)
    {
        System.out.println("valores["+i+"]= " + valores[i]);
    }
} //Fim do método imprimirVetor()

public static void imprimirLinha()
{
    System.out.println("-----");
} //Fim do método imprimirLinha()
} //Fim da classe
```

EXEMPLOS DE TESTE

```
Digite o tamanho do vetor a ser lido: 12
-----
DIGITE OS VALORES DOS ELEMENTOS DO VETOR
Digite o valor de num[0]= 3
Digite o valor de num[1]= 45
Digite o valor de num[2]= -12
Digite o valor de num[3]= 0
Digite o valor de num[4]= 32
Digite o valor de num[5]= 27
Digite o valor de num[6]= 63
Digite o valor de num[7]= 12
Digite o valor de num[8]= -5
Digite o valor de num[9]= 8
Digite o valor de num[10]= 97
Digite o valor de num[11]= 74
-----
Maior elemento = 97.0
Vetor ordenado em ordem crescente
-----
valores[0]= -12.0
valores[1]= -5.0
valores[2]= 0.0
valores[3]= 3.0
valores[4]= 8.0
valores[5]= 12.0
valores[6]= 27.0
valores[7]= 32.0
valores[8]= 45.0
valores[9]= 63.0
valores[10]= 74.0
valores[11]= 97.0
```

- 2) Escreva um programa na linguagem de programação Java que, no método `main()`, peça ao usuário que digite o tamanho de um vetor a ser criado. No método `main()` deve ser criado um vetor com o tamanho informado pelo usuário, mas ainda sem nenhum valor inserido pelo usuário. Este vetor criado deve ser passado como parâmetro para o método `lerVetor()`, que permitirá que o usuário digite cada número do vetor. O usuário deve digitar valores entre

zero até 10 no vetor, de tal forma que o vetor contenha uma sequência de números entre zero à 10. O método `lerVetor()` deve retornar ao método `main()` o vetor atualizado, ou seja, com os valores inseridos pelo usuário.

Em seguida, no método `main()` o vetor atualizado deve ser passado como parâmetro para o método `maiorFreq()`. Este método irá calcular a frequência de cada um de seus elementos, armazenando estas frequências em um vetor. Depois, este vetor com as frequências deve ser enviado ao método `imprimirVetor()`, para ser impresso. O método `imprimirVetor()` não deve retornar nenhum valor. Após a impressão do vetor, o método `maiorFreq()` deve determinar qual o valor de maior frequência. Caso haja mais de um valor tenha a maior frequência, o método deve mostrar todos estes valores. O método `maiorFreq()` não deve retornar nenhum valor.

```
import java.util.Scanner;

public class MaiorFrequencia
{
    public static void main(String args[])
    {
        //Declaração e inicialização de variáveis
        int numeros[];
        int tamanho=0;
        Scanner entradaNum = new Scanner( System.in );

        //Entrada de dados
        System.out.print("Digite o tamanho do vetor a ser lido: ");
        tamanho = entradaNum.nextInt();
        numeros = new int[tamanho];
        numeros = lerVetor(numeros);

        maiorFreq(numeros);
    }

    public static int[] lerVetor(int[] num)
    {
        Scanner ent = new Scanner(System.in);

        imprimirLinha();
        System.out.println("DIGITE OS VALORES DOS ELEMENTOS DO VETOR");
        imprimirLinha();

        for (int i=0; i<num.length; i++)
        {
            System.out.print("Digite o valor de num["+i+"]= ");
            num[i]=ent.nextInt();
        }
        return num;
    } //Fim do método lerVetor()

    public static void maiorFreq(int[] vetor)
    {
        int []vetMaiorFreq = new int[11];
```

```
int []vetIndiceMaiorFreq = new int[11];

for (int i=0; i<vetor.length; i++)
{
    vetMaiorFreq[vetor[i]] = vetMaiorFreq[vetor[i]] + 1;
}

imprimirLinha();
System.out.println("FREQUÊNCIAS DOS VALORES DO VETOR:");
imprimirVetor(vetMaiorFreq);

//Colocar em ordem crescente os valores da frequência, para
encontrar as maiores frequências
int tamanho=vetor.length, nVezes=vetor.length;

for (int i=0; i<tamanho;i++)
    vetIndiceMaiorFreq[i]=i;

for (int i=0; i<tamanho;i++)
{
    for (int k=1;k<tamanho-i; k++)
    {
        if (vetMaiorFreq[k-1]>vetMaiorFreq[k])
        {
            int aux;
            aux=vetMaiorFreq[k-1];
            vetMaiorFreq[k-1]=vetMaiorFreq[k];
            vetMaiorFreq[k]=aux;

            aux=vetIndiceMaiorFreq[k-1];
            vetIndiceMaiorFreq[k-1]=vetIndiceMaiorFreq[k];
            vetIndiceMaiorFreq[k]=aux;
        }
    }
}

//imprimirLinha();
//System.out.println("ORDEM CRESCENTE DAS FREQUÊNCIAS DOS VALORES
DO VETOR:");
//imprimirVetor(vetMaiorFreq);

//imprimirLinha();
//System.out.println("FREQUÊNCIAS DOS ÍNDICES DOS VALORES DO
VETOR:");
//imprimirVetor(vetIndiceMaiorFreq);

int i=10;
do
{
    System.out.println("Número " + vetIndiceMaiorFreq[i] + "
Frequência: " + vetMaiorFreq[i]);
    i=i-1;
} while (vetMaiorFreq[i]==vetMaiorFreq[i+1]);
```

```
//Fim do método maiorFreq()

public static void imprimirVetor(int[] valores)
{
    imprimirLinha();
    for (int i=0; i<valores.length; i++)
    {
        System.out.println("valores["+i+"]= " + valores[i]);
    }
}

public static void imprimirLinha()
{
    System.out.println("-----");
}

//Fim do método imprimirLinha()
//Fim da classe
```

EXEMPLOS DE TESTE

```

Digite o tamanho do vetor a ser lido: 11
-----
DIGITE OS VALORES DOS ELEMENTOS DO VETOR
-----
Digite o valor de num[0]= 0
Digite o valor de num[1]= 1
Digite o valor de num[2]= 1
Digite o valor de num[3]= 1
Digite o valor de num[4]= 2
Digite o valor de num[5]= 2
Digite o valor de num[6]= 2
Digite o valor de num[7]= 3
Digite o valor de num[8]= 3
Digite o valor de num[9]= 3
Digite o valor de num[10]= 4
-----
FREQUÊNCIAS DOS VALORES DO VETOR:
-----
valores[0]= 1
valores[1]= 3
valores[2]= 3
valores[3]= 3
valores[4]= 1
valores[5]= 0
valores[6]= 0
valores[7]= 0
valores[8]= 0
valores[9]= 0
valores[10]= 0
Número 3   Frequência: 3
Número 2   Frequência: 3
Número 1   Frequência: 3

```

3) Uma doceria possui o seguinte cardápio:

Código	Descrição	Preço Kg (R\$)
401	Pudim de Leite	R\$12,50
402	Bolo de Chocolate	R\$14,00
403	Torta de Morango	R\$15,00
404	Torta Holandesa	R\$16,50

Escreva um programa na linguagem Java que, no método `main()`, armazene o cardápio da doceria utilizando vetores.

OBS: Os valores do cardápio não devem ser digitados pelo usuário. Devem ser inseridos diretamente nos vetores pelo programador.

Ainda no método `main()`, deve-se permitir que o usuário escolha qual produto deseja comprar. O programa deverá exibir o cardápio para o usuário na tela, antes que ele escolha seu pedido. Para realização do pedido o programa deverá solicitar ao usuário o código do item e sua quantidade.

Considere que o usuário pode comprar mais de um produto apresentado no menu, e que ele decidirá não comprar mais nada caso digite a opção 9. Se o código digitado não for encontrado, o algoritmo deverá exibir a mensagem "Esse código não existe".

Após cada escolha do usuário, ainda no método `main()` deve ser chamado o método `calcularPedido()` que irá calcular quanto o usuário irá pagar em cada pedido.

Ao final o programa deverá calcular o valor da compra e exibir o valor na tela do computador.

```
import java.util.Scanner;

public class Doceria
{
    public static void main(String args[])
    {
        //Declaração e inicialização de variáveis
        String []codigo = {"401", "402", "403", "404"};
        String []descricao = {"Pudim de Leite", "Bolo de Chocolate", "Torta de
Morango", "Torta Holandesa"};
        double preco[] = {12.5, 14.0, 15.0, 16.5};
        int []totOpcao = new int[codigo.length];
        int opcao=0, qdade=0;
        double totAPagar=0;
        Scanner entradaNum = new Scanner( System.in );

        //Entrada de dados
        imprimirMenu(codigo,descricao, preco);
        System.out.print("Qual sua opção do menu (1, 2, 3 ou 4)? Ou digite 9 para sair:
" );
        opcao = entradaNum.nextInt();
```

```
while ( opcao != 9 )
{
    System.out.print("Quantas unidades deseja de " + descricao[opcao-1] + "?");

    qdade = entradaNum.nextInt();
    switch (opcao)
    {
        case 1:
            totAPagar=calcularPedido(opcao, preco, totOpcao, qdade, totAPagar);
            break;
        case 2:
            totAPagar=calcularPedido(opcao, preco, totOpcao, qdade, totAPagar);
            break;
        case 3:
            totAPagar=calcularPedido(opcao, preco, totOpcao, qdade, totAPagar);
            break;
        case 4:
            totAPagar=calcularPedido(opcao, preco, totOpcao, qdade, totAPagar);
            break;
        default:
            System.out.println("Opcao " + opcao + " escolhida não está cadastrada no nosso sistema.");
    } //fim do switch-case

    System.out.print("Qual sua opção do menu (1, 2, 3 ou 4)? Ou digite 9 para sair: " );
    opcao = entradaNum.nextInt();
} //fim do while

imprimirLinha();
System.out.println("Total a pagar de seu pedido: " + totAPagar);
} //Fim do método main()

public static double calcularPedido(int opcao, double[] p, int [] totOp, double q, double totPag)
{
    totOp[opcao-1] = totOp[opcao-1]++;
    totPag = totPag+q*p[opcao-1];
    return totPag;
} //Fim do método calcularPedido()
```

```
public static void imprimirMenu(String[] cod, String[] desc, double[] p)
{
    System.out.printf("%-7s %-7s %-20s %10s \n", "OPÇÃO", "CÓDIGO", "DESCRIÇÃO",
"PRECO Kg (R$)");
    for (int i=0; i<cod.length; i++)
    {
        System.out.printf("%-7s %-7s %-20s %10.2f \n", (int)i+1, cod[i], desc[i],
p[i]);
    }
}

//Fim do método imprimirMenu()

public static void imprimirLinha()
{
    System.out.println("-----");
}

//Fim do método imprimirLinha()

//Fim da classe
```

EXEMPLOS DE TESTE

```
OPÇÃO    CÓDIGO  DESCRIÇÃO          PRECO Kg (R$)
1         401    Pudim de Leite      12,50
2         402    Bolo de Chocolate   14,00
3         403    Torta de Morango    15,00
4         404    Torta Holandesa     16,50
Qual sua opção do menu (1, 2, 3 ou 4)? Ou digite 9 para sair: 4
Quantas unidades deseja de Torta Holandesa? 2
Qual sua opção do menu (1, 2, 3 ou 4)? Ou digite 9 para sair: 2
Quantas unidades deseja de Bolo de Chocolate? 3
Qual sua opção do menu (1, 2, 3 ou 4)? Ou digite 9 para sair: 1
Quantas unidades deseja de Pudim de Leite? 5
Qual sua opção do menu (1, 2, 3 ou 4)? Ou digite 9 para sair: 9
-----
Total a pagar de seu pedido: 137.5
```