

Documentação Trabalho Prático - AEDS I

Alunos: Izabela Cecília Silva Barbosa, Vinícius Tavares Coimbra Ribeiro

Professor: Rosilene Mota

Vídeo no Youtube: <https://youtu.be/R6t7sFgo2Es>

Explicação do código de controle de vendas de máscaras

Main:

```
public static void main(String[] args) {  
  
    Scanner ent = new Scanner(System.in);  
    int[][] produtos = new int[4][5];  
    String linha = null;  
    String[] data;  
    int quantidadeP, tipo, vendas, custo, pvenda;  
    int control = 1;  
  
    StringBuilder sb = new StringBuilder("s");  
    char resposta = sb.charAt(0);  
    String diretorio = "C:\\Users\\sucas\\Desktop\\trabalho\\baseProdutos.txt";  
    try {  
        File arqProdutos = new File(diretorio);  
        Scanner entArqProdutos = new Scanner(arqProdutos);  
        while (entArqProdutos.hasNextLine()) {  
            linha = entArqProdutos.nextLine();  
            // separa os ; que estão na linha e guarda em um vetor as posições separadas.  
            data = linha.split(";");  
            /*  
             * /*  
            * Os comandos abaixo salvam os tipos na matriz.  
            */  
            tipo = Integer.parseInt(data[0]);  
            quantidadeP = Integer.parseInt(data[1]);  
            vendas = Integer.parseInt(data[2]);  
            custo = Integer.parseInt(data[3]);  
            pvenda = Integer.parseInt(data[4]);  
            produtos[tipo][0] = tipo;  
            produtos[tipo][1] = quantidadeP;  
            produtos[tipo][2] = vendas;  
            produtos[tipo][3] = custo;  
            produtos[tipo][4] = pvenda;  
  
            // a quantidade de produto é a unica coisa que no programa em si vai alterar,  
            // logo, vamos guarda-la em um vetor para acesso posterior.  
        }  
        entArqProdutos.close();  
    } catch (IOException r) {  
        System.out.println("Falha na leitura");  
    }  
}
```

O trecho de código inicia o scanner de entrada de dados do teclado;

Inicia as variáveis necessárias;

Realiza o try para iniciar o arquivo;

Lê os dados do arquivo base, separa eles com base no ";", transforma para inteiro e guarda em uma matriz.

```

    }
    /*
     * verifica se usuario quer continuar no programa
     */
    while (resposta == 's') {
        menu(ent, produtos);
        System.out.println("Você deseja continuar\n(S)Sim --- (N)Nao");
        ent.nextLine();
        sb.append(ent.nextLine().toLowerCase());
        resposta = sb.charAt(control);
        control++;
    }
    criaRelatorioVendas(produtos);

    ent.close();

```

Após isso é executada a função Menu e após a função é realizada a verificação para saber se o usuário deseja continuar utilizando o programa.

Na saída do while é utilizada a função de criar relatórios, que será explicada posteriormente.

```

public static void menu(Scanner ent, int[][] produtos) {
    // variavel para quantidade que a pessoa deseja comprar
    int qntCompra, opt;
    System.out.println("=====Menu=====");

    System.out.println(
        "0-Mascara Infantil Lisa\n1-Mascara Infantil Estampada\n2-Mascara Adulta Lisa\n3-Mascara Adulta Estampada\n4-Generar Relatório de vendas");
    opt = ent.nextInt();

    /*
     * Se o valor retorna for true ele vai na matriz na linha onde se encontra o
     * produto entra na coluna que está o estoque, subtrai o valor a ser comprado
     * e depois vai no local onde está os produtos a serem vendidos e adiciona o
     * valor a ser comprado, no vetor total de vendas, ele está colocando na posição
     * dos produtos
     * o total de vendas feitas, posteriormente será usado para gerar o relatório de
     * vendas.
     */
    if (opt == 0 || opt == 1 || opt == 2 || opt == 3) {
        qntCompra = desejaComprar(ent);
        if (verificaEstoque(qntCompra, opt, produtos)) {
            manipulaProdutos(produtos, opt, qntCompra);
        }
    } else if (opt == 4) {
        criaRelatorioVendas(produtos);
        System.out.println("Relatório de Vendas gerado");
    } else {
        System.out.println("Nenhum produto selecionado");
    }
}

```

A função menu dispõe as opções de escolha para venda de máscaras e uma opção para gerar o relatório de vendas.

Caso o usuário escolha as opções 1 a 3 é executada a função desejaCompar

```
public static int desejaComprar(Scanner ent) {  
    int qntCompra;  
    System.out.println("Quantos você deseja comprar?");  
    qntCompra = ent.nextInt();  
    return qntCompra;  
}
```

A função pergunta quantas máscaras o usuário deseja comprar.

Após isso é executada uma função para verificar se quantidade comprada está disponível em estoque.

```
public static boolean verificaEstoque(int qntCompraN, int tipoProduto, int[][] produtos) {  
    boolean temEstoque = false;  
    if (produtos[tipoProduto][1] >= qntCompraN) {  
        temEstoque = true;  
        return temEstoque;  
    }  
    System.out.println("Fora de estoque");  
    return temEstoque;  
}
```

A função verifica se na matriz criada o número em estoque é maior do que o número comprado, e habilita a compra retornando um True, caso contrário retorna False por padrão.

```
public static void manipulaProdutos(int[][] produtos, int opt, int qntCompra) {  
    produtos[opt][1] -= qntCompra;  
    produtos[opt][2] += qntCompra;  
}
```

Dentro do If é executada a função manipula produtos, que realiza a adição da quantidade de produtos vendidos na matriz, e realiza a subtração do estoque.

Caso o usuário escolha a opção 4:

```

public static void criaRelatorioVendas(int[][] produtos) {

    // em cada linha desse vetor vai ficar o lucro de um produto, sendo a linha 5 o
    // lucro total

    int[] lucro = new int[5];

    calculaLucro(produtos, lucro);
    String nomeArquivo = "RelatorioVendas.txt";
    int i = 1;
    while (arquivoExiste(nomeArquivo)) {
        nomeArquivo = "Relatorio Vendas n" + i + ".txt";
        System.out.println(nomeArquivo);
        i++;
    }
    try (BufferedWriter arquivoRelatorio = new BufferedWriter(new FileWriter(new File(nomeArquivo)))) {
        for (int p = 0; p < 4; p++) {
            for (int j = 0; j < 5; j++) {

                arquivoRelatorio.write(produtos[p][j] + ";");

            }

            arquivoRelatorio.write("Lucro no produto: " + lucro[p]);
            arquivoRelatorio.newLine();
        }
        arquivoRelatorio.write("Lucro total: " + lucro[4]);
        arquivoRelatorio.flush();
    } catch (IOException e) {
        System.err.println(nomeArquivo);
    }
}

```

A função criaRelatorioVendas é executada, nessa função é chamada a função calculaLucro

```

public static void calculaLucro(int[][] produtos, int[] Lucro) {
    int lMascara;
    // lucro por mascara
    for (int ln = 0; ln < 4; ln++) {
        lMascara = produtos[ln][4] - produtos[ln][3];
        Lucro[ln] = lMascara * (produtos[ln][2]);
    }
    // resetando
    lMascara = 0;
    // lucro total
    for (int i = 0; i < 4; i++) {
        lMascara += Lucro[i];
    }
    Lucro[4] = lMascara;
}

```

A função necessita da matriz de produtos e de um vetor para ser alocado o lucro. Na função é feito o cálculo de (Preço de venda - Preço de compra) * Quantidade de venda, para calcular o lucro em cada tipo de máscara. Na posição 4 do vetor é calculado o Lucro total de todas as vendas.

```
private static boolean arquivoExiste(String nomeArquivo) {
    Path end = Paths.get(nomeArquivo); // converte o nome de arquivo em um endereço
    if (Files.exists(end)) { // se o arquivo for encontrado no endereço
        return true;
    }
    // se o arquivo não foi encontrado
    return false;
}
```

Após isso é executada uma função para verificar se o arquivo com o nome destinado a ele existe, caso ele exista a função retorna True.

```
while (arquivoExiste(nomeArquivo)) {
    nomeArquivo = "Relatorio Vendas n" + i + ".txt";
    System.out.println(nomeArquivo);
    i++;
}
try (BufferedWriter arquivoRelatorio = new BufferedWriter(new FileWriter(new File(nomeArquivo)))) {
    for (int p = 0; p < 4; p++) {
        for (int j = 0; j < 5; j++) {
            arquivoRelatorio.write(produtos[p][j] + ";");
        }

        arquivoRelatorio.write("Lucro no produto: " + lucro[p]);
        arquivoRelatorio.newLine();
    }
    arquivoRelatorio.write("Lucro total: " + lucro[4]);
    arquivoRelatorio.flush();
} catch (IOException e) {}
System.err.println(nomeArquivo);
```

Com a existência do nome de arquivo é gerado um relatório com um nome diferente, sendo ele baseado em números de ordem crescente.

Após a criação do nome do arquivo é feito um try para gerar um BufferedWriter, que cria o arquivo e escreve nele a matriz final, o lucro obtido em cada máscara e o lucro total.