

INSTITUTO FEDERAL DE MINAS GERAIS
CAMPUS OURO BRANCO

**Documentação de Tópicos em Desenvolvimento de
Software**

Aluna: Isabella Alane Darin Melo

Conselheiro Lafaiete
28/09/2020

1.0 INTRODUÇÃO

Esse trabalho tem como intuito auxiliar os diversos empreendedores na administração estoques por meio de uma aplicação prática, a qual o usuário consegue comprar o produto inserindo seu código e alterando a quantidade desejada. Já o administrador, consegue alterar o preço dos produtos, removê-los da compra e ainda finalizá-la atualizando o arquivo referencial de produtos, possibilitando um controle de estoque muito eficaz e prestativo.

2.0 IMPLEMENTAÇÃO DAS CLASSES

2.1 FakeBD

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;

/**...4 lines */
public class FakeBD {
    //vetor que será armazenado os produtos
    private static Vector<Produto> produtos;

    //leitura do arquivo externo
    private static void cargaArquivo() {...52 lines }
    public static Produto consultaProdutoCod(int codigo) {...17 lines }
    //alterando informações na unidade de armazenamento secundária
    public static void atualizaArquivo() {...28 lines }
}
```

A Classe FakeBD é usada como uma forma de substituir um banco de dados, que efetua leitura de arquivos externos a partir dos objetos FileReader, BufferedReader, atualiza os arquivos em caso de modificações através de objetos BufferedWriter e FileWriter e consulta a existência dos produtos através de seu código percorrendo o vector Produtos.

Métodos, Funções e objetivos:

- **cargaArquivo()** - Pega o arquivo externo na memória e cria um vetor para produtos.
- **consultaProdutoCod()** - O código do produto é inserido como parâmetro e a função retorna o objeto produto
- **atualizaArquivo()** - Modifica e otimiza o arquivo externa

2.2 CompraGUI

Código	<table><tr><th>Title 1</th><th>Title 2</th><th>Title 3</th><th>Title 4</th></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>	Title 1	Title 2	Title 3	Title 4																	Remove
Title 1	Title 2	Title 3	Title 4																			
Nome																						
Quantidade	1																					
Preço Unit.		Estoque																				
Add produto	Valor total R\$ 0,00	Finalizar																				

Na Classe CompraGUI, é possível a partir do código, encontrar o nome do produto, o preço e selecionar a quantidade desejada para inserir no carrinho de compra. Isso é possível pois a Classe busca métodos da FakeBD, cria um produto e um novo modelo de tabela para uma nova compra. Caso a quantidade inserida de produtos desejados for maior que a quantidade disponível no estoque, o programa exibe mensagem de produtos insuficientes. Já caso o usuário tenha adicionado um produto e não queria mais completar a compra desse, é possível através do botão

remover retirá-lo da tabela, assim como também pode-se alterar a quantidade do produto desejada. Ao longo da compra, o preço total da compra é implementado juntamente com a adição de produtos. Ao finalizar a compra, o arquivo com os produtos é atualizado e uma nova compra pode ser realizada após todos os campos serem limpos e receberem novos valores.

Métodos, Funções e objetivos:

- **codigoTextKeyTyped()** - Após o código ser digitado, o método faz uma consulta na lista de produtos e retorna seu nome, e preço unitário.
- **quantidadeTextKeyReleased()** - Após a quantidade desejada ser inserida e conferida, calcula-se o preço parcial do produto e o grava no textField do preço.
- **addProdutoButtonMouseClicked()** - Ao clicar no botão adicionar, o método insere no carrinho de compras o produto requisitado e exibe informações para o usuário sobre o estoque e quantidades disponíveis conferido pela função `verificaDado()` e após a finalização da compra
- **removerButtonActionPerformed()** - Neste método, é possível após a inserção de um produto, removê-lo da compra, mas nesta funcionalidade é requisitada a senha do gerente para validação da atividade.
- **verificaDado()** - É uma função chamada no método `addProdutoButtonMouseClicked()` para fazer a conferência da quantidade de produtos no estoque disponível.
- **finalizarButtonActionPerformed()** - Ao clicar no botão finalizar, a tabela do carrinho de compra é atualizada e os campos de preenchimento de informações tornam-se vazios.
- **atualizaDados()** - Atualiza a tabela carrinho e calcula o preço parcial dos produtos.

2.3 JanelaPrincipal

```
import java.awt.BorderLayout;
import java.awt.CardLayout;
import javax.swing.JPanel;
import javax.swing.JScrollPane;

/**...4 lines */
public class JanelaPrincipal extends javax.swing.JFrame {

    //representa a barra de rolagem
    private static JScrollPane barraRolagem;

    //auxilia na transição de painéis existentes
    private static JPanel trocaInformacao;

    //gerencia qual painel estará visível
    private static CardLayout paineisLayout;
    /** Creates new form JanelaPrincipal ...3 lines */
    public JanelaPrincipal() {
        initComponents();
        meuInitComponents();
    }
    private void meuInitComponents() {...24 lines }
    public static void efetuaTransicao(JPanel novoPainel, String nome){
        trocaInformacao.add(novoPainel, nome);
        paineisLayout.show(trocaInformacao, nome);

        //corrigindo o bugg do tamanho das barras de rolagens
        trocaInformacao.setPreferredSize(novoPainel.getPreferredSize());
    }
}
```

Na Classe da Janela Principal, é criado um JPanel para a barra de rolagem de forma que o frame se adapte com o tamanho do painel e , para trocar informações, a troca de painéis é feita por conta do CardLayout.

Métodos, Funções e objetivos:

- **meuInitComponents()** - Adiciona o novo painel no frame e o posta os paineisLayout.
- **formWindowClosing()** - Chama o método atualizaArquivo() da classe FakeBD para a execução.

2.4 ModeloTabelaCompra

```
public class ModeloTabelaCompra extends AbstractTableModel {
    //vetor responsavel por adicionar produtos no carrinho de compras
    private Vector<Produto> carrinhoCompra;
    private CompraGUI painel;

    public ModeloTabelaCompra(CompraGUI painel) {...4 lines }

    //linhas
    @Override
    public int getRowCount() {...4 lines }

    //colunas
    @Override
    public int getColumnCount() {...4 lines }

    //valor das células
    @Override
    public Object getValueAt(int linha, int coluna) {...16 lines }

    //adicionar conteúdo no carrinho
    public void addProduto(Produto vendido) {...3 lines }

    //remoção do produto do carrinho
    public void removeProduto(int indice) {...3 lines }
```

É responsável por organizar o conteúdo das tabelas, adicionar e remover produtos.

Métodos, Funções e objetivos:

- **ModeloTabelaCompra(CompraGUI painel)** - Inicializa o carrinho de compra e o painel.
- **getRowCount()** - Retorna a quantidade de produtos variados.
- **getColumnCount()** - Retorna o número 4, contabilizado pelo nome, código do produto, quantidade e preço unitário.
- **getValueAt()** - Usa os parâmetros da linha e coluna para armazenar temporariamente o produto procurado.
- **addProduto()** - Adiciona o produto vendido no carrinho de compras.

- **removeProduto()** - Remove o produto através do índice da coluna.
- **limpaCarrinho()** - Efetua a função clear() no carrinho de compras.
- **getColumnName()** - Retorna o nome das respectivas colunas: nome, código, quantidade e preço unitário.
- **isCellEditable()** - Determina a célula quantidade como editável.
- **setValueAt()** - Determina o novo valor para quantidade do produto após a verificação de uma senha criada pelo gerente.
- **getColumnClass()** - Efetua a conversão de Strings para os respectivos tipos das variáveis
- **calculaPrecoParcial()** - Calcula o preço parcial dos produtos após a busca no vetor de carrinho de compra é calculado pela multiplicação do preço unitário pela quantidade desejada.
- **produtosCarrinho()** - retorna o carrinho de compras.

2.5 Produto

```
public class Produto {
    private int codigo;
    private String nome;
    private double preco;
    private int quantidade;

    public Produto(int codigo, String nome, double preco, int quantidade) {...6 lines }

    public int getCodigo() {...3 lines }

    public String getNome() {...3 lines }

    public double getPreco() {...3 lines }

    public void setPreco(double preco) {...3 lines }

    public int getQuantidade() {...3 lines }

    public void setQuantidade(int quantidade) {...3 lines }

    @Override
    public String toString() {...3 lines }

}
```

Na Classe Produto, o objeto é criado com seus devidos atributos: código, nome, preço e quantidade no estoque.

Métodos, Funções e objetivos:

- **Produto(int codigo, String nome, double preco, int quantidade)** - Construtor de classe inicializando as variáveis.
- **getNome()** - Retorna o nome do produto.
- **getPreco()** - Retorna o preço do produto.
- **getQuantidade()** - Retorna a quantidade disponível de certo produto
- **setPreco()** - Passa por parâmetro o preço do produto, sendo capaz de alterá-lo.
- **setQuantidade()** - Passa por parâmetro o quantidade de um produto específico produto, sendo capaz de alterá-la..
- **toString()** - Representa o objeto instantaneamente através de uma String que é separada por “;” assim como no arquivo.csv.

3.0 CONCLUSÃO

De modo geral, o trabalho foi bem produtivo, sendo o tratamento de arquivos uma solução incrível e muito prática para armazenamento de informações e foi um rico aprendizado. Houve muitas dificuldades em relação à manipulação de frames e painéis, já que o conteúdo foi pouco estudado pela aluna. A ideia do programa, apesar de já existente, deveria ser implementada em diversas lojas e balcões para facilitar a organização e manipulamento de produtos.

4.0 REFERÊNCIAS BIBLIOGRÁFICAS

<https://docs.oracle.com/javase/7/docs/api/>

Vídeo aulas do professor Saulo publicadas no Moodle -

<https://ead2.ifmg.edu.br/ourobranco/mod/page/view.php?id=4588>

https://www.youtube.com/watch?time_continue=4&v=UbjLOn1PAKw&feature=emb_title

<https://www.youtube.com/watch?v=OCIKwp-NgVE>