

Roteiro InfoSys - Aula 02

Docentes: Caique Zaneti Kirilo, Luiz Machi Lozano, Ricardo Drudi

Passo 33 - Criar a classe java Produto dentro do Pacote Model, com os atributos abaixo:

```
private String descricao = null;
private int estoque;
private boolean ativo = false;
private double valor = 0;
private double custo = 0;
```

Criar gettes/settes da Classe Java Produto.

Criar o método construtor da Classe Java Produto, conforme abaixo:

```
19 public Produto(String descricao, int estoque, double valor, double custo){
    setDescricao(descricao);
    setEstoque(estoque);
    setValor(valor);
    setCusto(custo);
    setAtivo(true);
25 }
26 }
```

Fazer a sobrescrita do método toString dentro da classe Java Produto, conforme imagem abaixo:

```
68 @Override
    public String toString() {
        String ret = null;
        if (isAtivo()) ret = "[Ativo] - ";
        else ret = "[Fora de Linha] - ";
        ret = ret + "[Item: " + getDescricao() +
71         " - [Estoque: " + getEstoque() +
72         " - [Custo: " + getCusto() +
73         " - [Preço de Venda: " + getValor() + "]]";
74         return ret;
75     }
76 }
77 }
78 }
```

Passo 34 - Devemos configurar um evento para o Spinner. Clicar com o botão direito do mouse no Spinner spProEstoque / Eventos / Change / stateChanged e programar o seguinte evento:

```
private void spProEstoqueStateChanged(javax.swing.event.ChangeEvent evt) {
    int v = (int) spProEstoque.getValue();
    if (v < 0) spProEstoque.setValue(0);
    // TODO add your handling code here:
}
```

Obs.: Esse evento irá pegar em tempo de execução o valor do Spinner e não deixará o Spinner ficar com valor negativo.

Passo 35 – Agora, igualmente fizemos com o Cadastro de Clientes, iremos criar um método chamado `gravaProduto`, conforme imagem abaixo:

Obs.: Note que damos um tratamento diferente ao spinner.

```
private void gravaProduto() {
    String descricao = tfProDescricao.getText();
    int estoque = (int) spProEstoque.getValue();
    double custo = Double.parseDouble(tfProCusto.getText());
    double valor = Double.parseDouble(tfProValor.getText());
    System.out.println(new Produto(descricao, estoque, custo, valor));
    limpaProduto();
}
```

Passo 36 – Agora iremos programar o método `limpaProduto`, note que método `limpaProduto` tratamos o Spinner e o CheckBox também de forma diferente:

```
private void limpaProduto() {
    tfProDescricao.setText("");
    spProEstoque.setValue(0);
    cbAtivo.setSelected(true);
    tfProCusto.setText("");
    tfProValor.setText("");
    tfProDescricao.requestFocus();
}
```

Passo 37 - Os alunos deverão tentar programar os botões Gravar e Limpar para o cadastro de Produtos, os mesmos serão trabalhados no início da próxima aula.

Passo 38 - Para o sistema verificar qual painel está selecionado quando o botão Limpar for clicado devemos alterar o evento do botão:

```
private void btLimparActionPerformed(java.awt.event.ActionEvent evt) {
    if (tbTela.getSelectedComponent() == pnCliente) limpaCliente();
    else if (tbTela.getSelectedComponent() == pnProduto) limpaProduto();
    // TODO add your handling code here:
}
```

Passo 39 - Agora façamos que foi realizado no passo anterior, porém para o botão Gravar.

```
private void btGravarActionPerformed(java.awt.event.ActionEvent evt) {
    if (tbTela.getSelectedComponent() == pnCliente) gravaCliente();
    else if (tbTela.getSelectedComponent() == pnProduto) gravaProduto();
    // TODO add your handling code here:
}
```

Passo 40 - Criar um pacote chamado Util.

Passo 41 - Dentro do pacote Util criar uma classe chamada Configura e inserir o seguinte bloco de códigos pra configurar o LookAndFeel.

```

public class Configura {
    public static void LookAndFeel(String laf) {
        try {
            for (UIManager.LookAndFeelInfo info : UIManager.getInstalledLookAndFeels()) {
                if (laf.equals(info.getName())) {
                    UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (UnsupportedLookAndFeelException | ClassNotFoundException | InstantiationException | IllegalAccessException e) {
            System.out.println("Erro: " + e.getMessage());
            e.printStackTrace();
        }
    }
}

```

Passo 42 - Dentro do pacote controller, criar uma classe chamada Principal. Nela chamar o método do LookAndFeel e instanciar a TelaCadastro:

*Obs.: fazer as importações das classes Util.Configura e View.TelaCadastro.

```

package Controller;

import Util.Configura;
import View.TelaCadastro;

/**
 *
 * @author luizm
 */
public class Principal {
    public static void main(String[] args) {
        Configura.lookAndFeel("Windows Classic");
        new TelaCadastro().setVisible(true);
    }
}

```

Passo 43 - Fazer o cadastro de Técnicos de acordo com o levantamento de requisitos.

Nome:

Salário:

Valor da Hora:

Obs.: Esse código não será passado no roteiro.

Criar uma classe chamada Mascara dentro do pacote Util. Implementar as mascaras dos campos cpf, telefone e celular da tela de cadastro de clientes.