


O diagrama ilustra a arquitetura MVC (Model-View-Controller). No topo, há uma barra azul com o logo 'FIRJAN' e o slogan 'INFORMA, FORMA, TRANSFORMA.'. Abaixo, um ícone de uma pessoa (usuário) está conectado por uma seta dupla horizontal ao bloco 'view'. O bloco 'view' está conectado por setas duplas verticais aos blocos 'controller' (acima) e 'model' (ao lado). O bloco 'controller' também está conectado por uma seta dupla vertical ao bloco 'model'. Todos os blocos são retângulos azuis com texto branco.

Projeto *GUI06_MVC*

Faremos um projeto simples seguindo o padrão MVC, onde será implementada uma calculadora que realiza a operação matemática de soma.

A imagem mostra uma janela de uma calculadora simples criada em Java. O título da janela é 'Calculadora:'. Ela possui dois campos de entrada para números, um botão '+' entre eles e um botão '=' para calcular o resultado, que é exibido em um campo de saída. Abaixo dos campos, há um botão 'Calcula'. A janela tem uma barra de título padrão com botões de minimizar, maximizar e fechar.

Crie o referido projeto, e dentro dele crie os pacotes *model*, *view* e *controller*.

 4

Classe CalculaModel

```
package model;

public class CalculaModel {
    private int primeiroNumero;
    private int segundoNumero;
    private int valorCalculo;

    public void setPrimeiroNumero(int primeiroNumero) {
        this.primeiroNumero = primeiroNumero;
    }

    public void setSegundoNumero(int segundoNumero) {
        this.segundoNumero = segundoNumero;
    }

    public int getValorCalculo(){
        return valorCalculo;
    }

    public int somaDoisNumeros(){
        valorCalculo = primeiroNumero + segundoNumero;
        return valorCalculo;
    }
}
```



5

Classe CalculaView

Faça a classe receber herança de JFrame.

Crie os seguintes objetos globais:

```
private JTextField primeiroNumero = new JTextField(10);
private JTextField segundoNumero = new JTextField(10);
private JTextField resultado = new JTextField(10);
private JLabel sinalMais = new JLabel(" + ");
private JLabel sinalIgual = new JLabel(" = ");
private JButton botaoCalcula = new JButton("Calcula");
```



6



Classe CalculaView

Digite o seguinte código dentro do construtor:

```
super("Calculadora:");

resultado.setEditable(false);

JPanel painel = new JPanel();
painel.add(primeiroNumero);
painel.add(sinalMais);
painel.add(segundoNumero);
painel.add(sinalIgual);
painel.add(resultado);

Container container = getContentPane();
container.add(painel, BorderLayout.CENTER);
container.add(botaoCalcula, BorderLayout.SOUTH);

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
pack();
```



7

Classe CalculaView

```
public int getResultado() {
    return Integer.parseInt(resultado.getText());
}

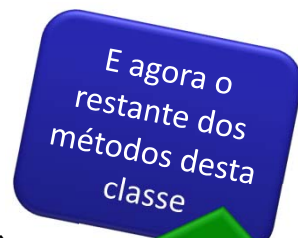
public void setResultado(int resultado) {
    this.resultado.setText(Integer.toString(resultado));
}

public int getPrimeiroNumero() {
    return Integer.parseInt(primeiroNumero.getText());
}



public int getSegundoNumero() {
    return Integer.parseInt(segundoNumero.getText());
}

public void addCalculaListener(ActionListener listenerDoBotaoCalcula){
    botaoCalcula.addActionListener(listenerDoBotaoCalcula);
}

public void exibeMensagemDeErro(String mensagemDeErro){
    JOptionPane.showMessageDialog(this, mensagemDeErro,
        "Mensagem de erro:", JOptionPane.ERROR_MESSAGE);
}
```



8

  INFORMA. FORMA. TRANSFORMA.


Classe CalculaController



Nesta classe ocorrem as ações e as interligações com as classes dos pacotes model e viewer.

Parte 1 de 2

```
private CalculaView calculaView = new CalculaView();
private CalculaModel calculaModel = new CalculaModel();

public CalculaController(){
    this.calculaView.addCalculaListener(new CalculaListener());
    this.calculaView.setVisible(true);
}
```

 9

  INFORMA. FORMA. TRANSFORMA.

Classe CalculaController

Parte 2 de 2


```
class CalculaListener implements ActionListener{


    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            calculaModel.setPrimeiroNumero(calculaView.getPrimeiroNumero());
            calculaModel.setSegundoNumero(calculaView.getSegundoNumero());

            calculaView.setResultado(calculaModel.somaDoisNumeros());

        } catch (NumberFormatException nfe) {
            calculaView.exibeMensagemDeErro("Digite 2 números inteiros válidos.");
        }
    }
}

public static void main(String[] args) {
    new CalculaController();
}
```

 10





Exercício de Fixação:

Copie o projeto GUI06_MVC e cole renomeando-o para GUI061_ExercicioMVC.


Implementar as operações matemáticas de subtração, multiplicação e divisão utilizando uma das seguintes opções:

- ComboBox para as operações;
- RadioButton para as operações;
- Um botão para cada operação.



11




Dúvidas?




12

INFORMA, FORMA, TRANSFORMA.


Bibliografia



Java Como Programar 8ª Edição
Paul Deitel e Harvey Deitel
Ed. Pearson



Java 8 Programação de Computadores
José Augusto N. G. Manzano & Roberto Affonso da Costa Junior
Ed. Érica | Saraiva



Fundamentos de Computação e Orientação a Objetos Usando Java
Francisco A. C. Pinheiro
Ed. LTC

13