



UNIVERSIDADE ESTADUAL DE SANTA CRUZ - UESC

PRÓ-REITORIA DE GRADUAÇÃO – PROGRAD

DEPARTAMENTO DE CIÊNCIAS EXATAS - DCET

COLEGIADO DE CIÊNCIA DA COMPUTAÇÃO - COLCIC

Lista de Exercícios

Curso	Ciência da Computação
Disciplina	Linguagem de Programação III
Professor(es)	Otacílio José Pereira
Assunto	- Capítulo 5: Interface gráfica

Parte 1 - Sobre o exemplo que utiliza o JOptionPane

- a) O que ocorrerá com o programa quando este comando for executado?

```
JOptionPane.showMessageDialog(null, "Objeto criado com sucesso");
```

- b) Qual a principal utilidade do comando a seguir?

```
JOptionPane.showInputDialog("...")
```

- c) O uso do JOptionPane envolve a criação de objetos para as janelas ou se utiliza atributos/metodos de?

Parte 2 - Sobre o exemplo que cria uma primeira janela vazia sem componentes

- a) Como o conceito de herança foi aplicado ao criarmos nossa primeira janela?
- b) Está faltando algo no código a seguir para conseguirmos ter uma classe janela minimamente instanciável?

```
5 package InterfaceGrafica.Ex02_JanelaVazia;
6
7 import javax.swing.JFrame;
8 public class FrmConta extends JFrame{
9
10 }
```

- c) Em qual linha a seguir é possível perceber que uma janela nada mais é do que um objeto em memória? O objeto janela ao ser criado (primeira linha) já é visualizada pelo usuário?

```
public static void main(String[] args)
{
    FrmConta f = new FrmConta();
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.setBounds(300, 300, 400, 200);
    f.setVisible(true);
}
```

- d) Comentamos ao utilizar orientação a objetos que é importante encapsular detalhes das implementações e fazemos isso pelos métodos getters e setters. No exemplo de código a seguir que informações encapsuladas estão sendo acessadas por métodos setters? Qual a função de cada um?

```
public static void main(String[] args)
{
    FrmConta f = new FrmConta();
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.setBounds(300, 300, 400, 200);
    f.setVisible(true);
}
```

- e) No exemplo de código anterior, qual a classe representa a janela? E qual a superclasse dela?

Parte 3 - Sobre o exemplo que trabalhava com os controles visuais

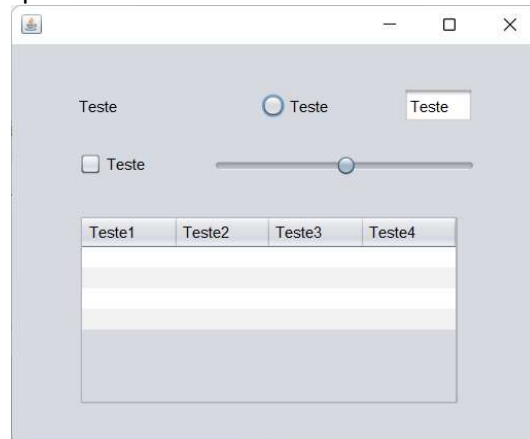
- Explique o que são os controles visuais e qual o papel deles em uma janela?
- Escolha dois exemplos de controles visuais que usamos em sala e escreva uma linha de código que cria dois destes controles e respectivamente com os textos "Controle visual 1" e "Controle visual 2"
- Pesquise por outros dois ou três tipos de controles visuais, explique para que servem e um exemplo de situação em que podem ser usados.
Exemplo: O JComboBox é um controle que permite a seleção dentre uma lista de opções. Em uma tela de compra de passagens ele pode ser usado para selecionar a cidade de origem ou de destino de uma viagem dentre uma lista de cidades possíveis.

d) Observe o exemplo a seguir. Apresente como ficará a interface.

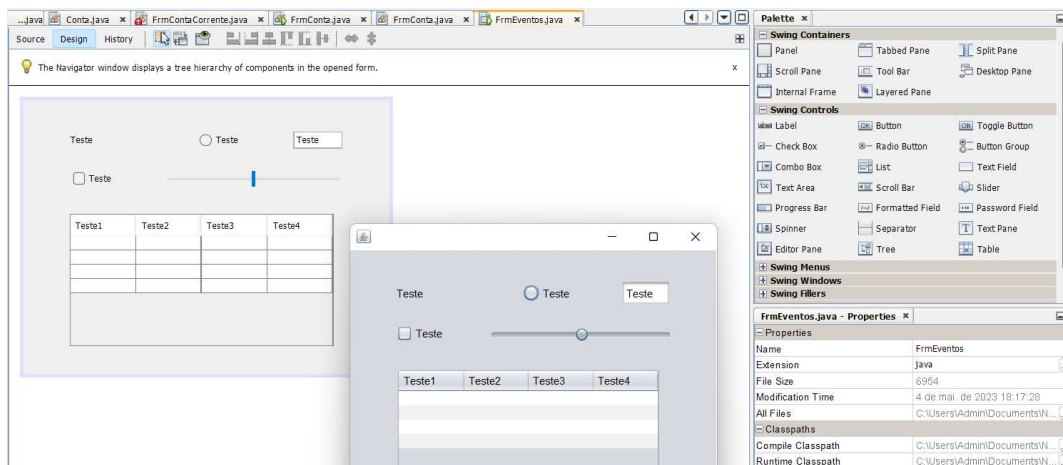
```
36 public FrmContaCorrente ()
37 {
38     super("Interface do Formulário");
39     //setLayout(new FlowLayout());
40     setLayout(new GridLayout(2, 2));
41
42     lblNumero = new JLabel();
43     lblNumero.setText("Número");
44     txtNumero = new JTextField();
45
46     lblAgencia = new JLabel();
47     lblAgencia.setText("Agência");
48     txtAgencia = new JTextField();
49     add(lblNumero);
50     add(lblAgencia);
51     add(txtNumero);
52     add(txtAgencia);
53 }
```

e) Qual o papel de um gerenciador de layout ao se trabalhar com interface do usuário?

f) Perceba que os controles são objetos, claro, são objetos de uma classe do Swing que dá a eles a Funcionalidade esperada em tela. Um JLabel é apenas um rótulo para algo, um JTextField serve para entrada de texto e tudo o mais. Na tela a seguir indique quais são os elementos de cada uma destas classes.



- g) Em nossos exemplos, criamos nossa interface “manualmente”, comando a comando. As ferramentas em geral oferecem um suporte para criação de interface na forma WYSWYG (what you see is what you get, o que você vê é o que você tem, algo assim!). Na figura a seguir está um exemplo disso no Netbeans.
- Aponte nesta tela as seguintes regiões:
- A região em que se desenha o layout da tela em si
 - A região em que se escolhe e “pluga” na tela os tipos de controles visuais
 - A região em que se pode mudar as propriedades dos controles (nome, texto que apresenta, cor de fundo e outros)
 - A região que permite acionar a edição dos eventos dos controles em tela



- h) Apresente pelo menos duas vantagens ao se criar uma interface no estilo que trabalhamos, “manualmente”. E apresente pelo menos 2 vantagens ao se trabalhar em um estilo exemplificado com o Netbeans.

Parte 4 - Sobre o exemplo que manipulava eventos

- a) Em uma interface, as interações com os controles produzem eventos mas que rodam em "modo silencioso", ou seja não há algum objeto que os tratem, que execute as ações relacionadas aos eventos. Para habilitar o tratamento dos eventos, precisamos de lidar com conceitos como Handler, Evento, Listener e outros. Nos exemplos que criamos.
- O que é a classe Handler?
 - O que é um listener?
 - Em que momento um handler é criado e associado a um controle?
 - Uma vez com os listeners “cadastrados” no controle visual, o que ocorre quando um evento é disparado no controle visual.
- b) A classe Handler herda de qual outra “classe” ou elemento da orientação a objetos?
- c) Em que momento um handler ou listener é catalogado para passar a ouvir um evento do botão?
- d) Qual método na classe handler foi sobrecarregado e como ele identificava a fonte do evento (btnNovo ou btnLimpar). Como a sobrecarga de método foi codificada?
- e) O que o comando a seguir faz e em qual parte do código ele deve estar?

```
// rotinas de tratamento de evento
EventoHandler handler = new EventoHandler();
btnLimpar.addActionListener(handler);
btnNovo.addActionListener(handler);
```

- f) Imagine que você precisa de adicionar um botão para limpar os campos de uma tela. Apresente quais modificações você precisaria fazer no código para adicionar esta funcionalidade:

Passo 1) Criar uma classe Handler que herda/implementa um Listener

Passo 2) ...

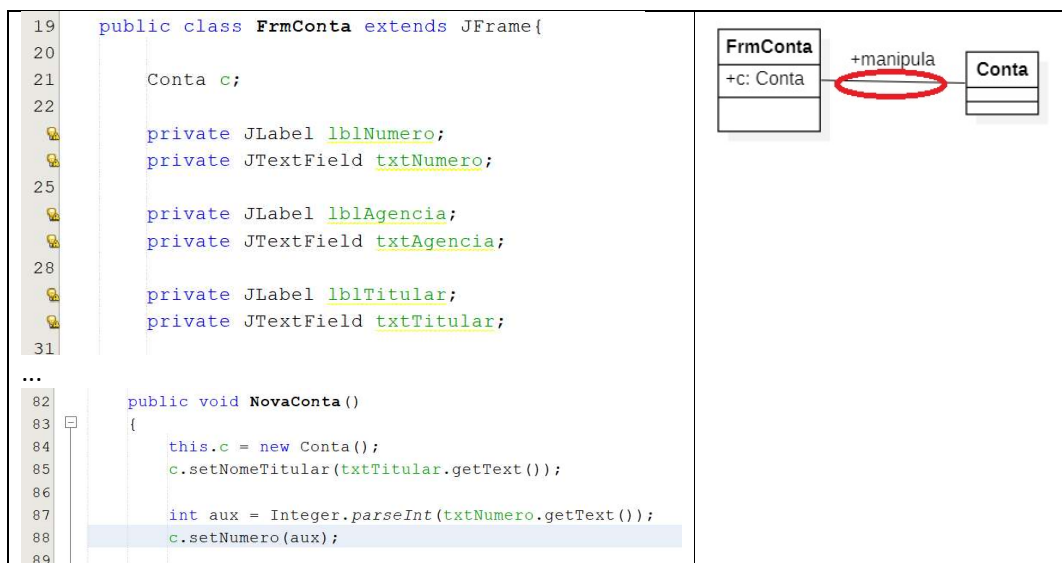
Passo 3) “Catalogar” o handler para que ele passe a escutar os eventos do botão btnLimpar via comando addActionListener

Passo 4) ...

...

Parte 5 - Sobre a vinculação da classe Janela com os objetos da classe conta (objeto e vetor)

- a) Quando estamos trabalhando com OO , é comum usarmos um diagrama de classe. Analise o diagrama de classes a seguir e aponte em que parte da codificação está ocorrendo o que está marcado com a elipse vermelha.



- b) Para um novo objeto ser criado, é preciso o seguinte código encapsulado no método NovaConta().

```

87 public void NovaConta ()
88 {
89     // Criando um objeto auxiliar
90     Conta c = new Conta();
91     // Recuperando os valores da interface no
92     // objeto
93     c.setNomeTitular(txtTitular.getText());
94     int aux = Integer.parseInt(txtNumero.getText());
95     c.setNumero(aux);
96     aux = Integer.parseInt(txtAgencia.getText());
97     c.setAgencia(aux);
98     double saldo = Double.parseDouble(txtSaldo.getText());
99     c.setValor(saldo);
100
101     // Adicionando o objeto no vetor
102     ultimaConta++;
103     contas[ultimaConta] = c;
104
105     JOptionPane.showMessageDialog(null,
106     "Objeto criado com sucesso! Numero : " + c.
107 }

```

- Qual diferença deste código para o código da questão anterior?
- Indique para que serve a linha 90.
- Indique para que servem as linhas 94 e 95.
- Indique para que servem as linhas 102 e 103.
- Onde o vetor contas deve ter sido declarado?
- Qual o papel da variável ultimaConta?

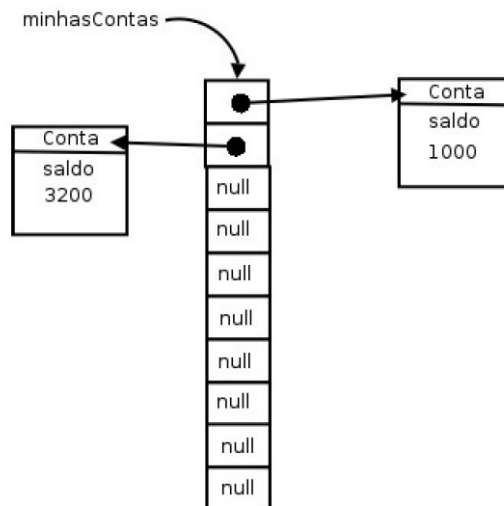
c) Qual o problema no código a seguir.

```

47 Conta[] contas = new Conta[10];
48
49 Conta c = contas[1];
50 c.setNumero(100);

```

Compare com esta figura para entender o erro.



Parte 6 – Exemplo de Container

- 1) O que é um container? Qual exemplo de container usamos em nossas interfaces?
- 2) Um gerenciador de layout é associado a um container ou a uma janela (na verdade seu contentpane)?
- 3) Quais as vantagens de se usar containers ao criar interfaces gráficas?
- 4) Fazendo uma analogia com interface web, quantos containers devem existir nesta interface?

