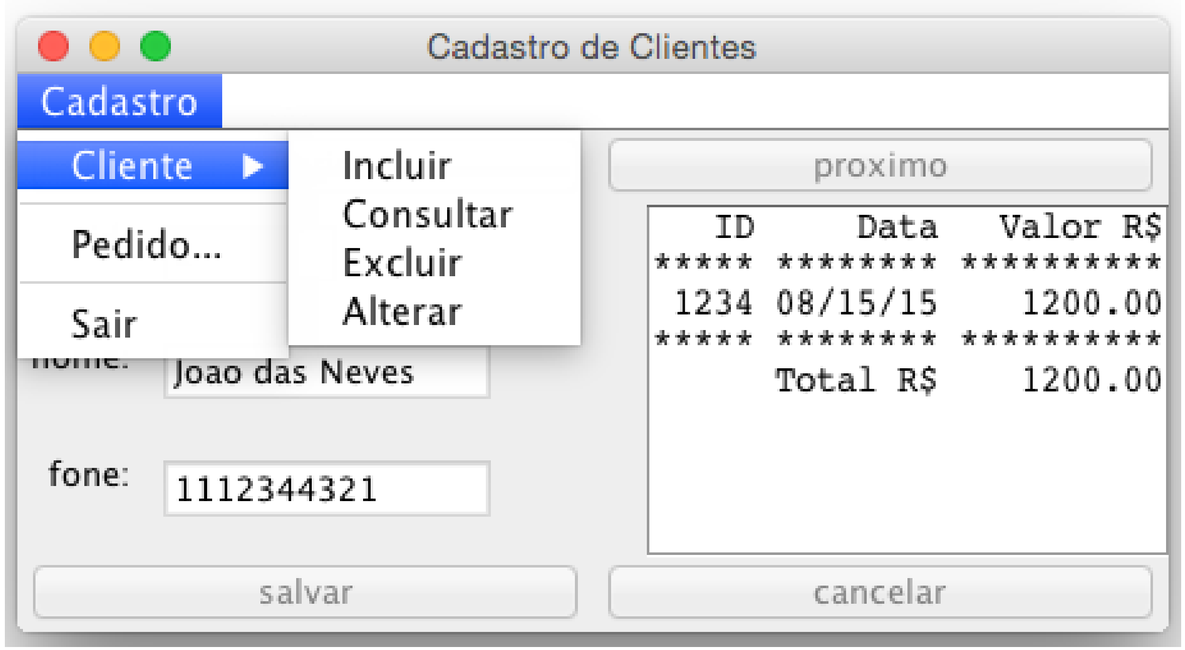
Interface Gráfica

Programação Orientada a Objetos - Aula 10

Professor: Hamilton Machiti da Costa


Cadastro de Clientes
Cadastro
Cliente
Pedido...
Sair
Incluir
Consultar
Excluir
Alterar
proximo
ID Data Valor R$
***** ******** **********
1234 08/15/15 1200.00
***** ******** **********
Total R$ 1200.00
Joao das Neves
fone:
1112344321
salvar
cancelar

Imports

- Imports: em geral, usam-se 3 packages diferentes
  - **javax.swing.***, que tem os componentes mais novos que sempre começam com jota.
  - **java.awt.***, que tem os componentes mais antigos, sem o jota.
  - **java.awt.event.***, que contém as interfaces de eventos.

```
1 import javax.swing.JFrame;
2 import javax.swing.JButton;
3 import javax.swing.JTextField;
4 import javax.swing.JLabel;
5 import java.awt.Container;
6 import java.awt.FlowLayout;
7 import java.awt.event.ActionEvent;
8 import java.awt.event.ActionListener;
```

Classe

- **JFrame:** uma tela com moldura, que pode ser maximizada e minimizada e que é vista pela aplicação, gerando ícone na barra de tarefas, é um JFrame.

- Para sua tela ser um JFrame é fácil. Basta usar a herança e estender JFrame.

```
public class TelaCalculadora extends JFrame{


}
```

Controle de Eventos

- O mecanismo de controle de eventos do Java funciona assim:
  - Os componentes de tela geram eventos quando o usuário faz alguma coisa.
  - Quando ele clica o mouse, quando ele digita em um campo, quando ele aperta um botão, um evento é gerado e passado para a JVM.
  - Você pode "pegar" estes eventos e mandar o sistema fazer alguma coisa quando eles acontecerem inscrevendo a classe que irá fazer alguma coisa como ouvinte (listener) do evento. Você verá isso mais adiante.
  - Por enquanto, é importante dizer que para poder ser ouvinte de eventos gerados por um botão a classe precisa implementar a interface ActionListener e, como consequência, implantar o método definido por ela, que é o public void actionPerformed(ActionListener), também mais adiante.

```
10 public class TelaCalculadora extends JFrame implements ActionListener{
11
12     public void actionPerformed(ActionEvent e){
13
14     }
15 }
```

Elementos Gráficos

- Crie suas variáveis de instância, que, neste caso, serão os elementos gráficos da tela, ou *widgets*:

```
12      private JButton botao; // botao
13      private JTextField texto; //campo texto de uma linha
14      private JLabel etiqueta; //etiqueta de nome do campo
```

Construtor

- No construtor você irá efetivamente montar a tela

- A primeira coisa a fazer é chamar o **super()** passando como parâmetro o título da tela que irá ficar na barra superior da janela.

```
16      public TelaCalculadora(){
17      //chamar construtor da superclasse e configurar o titulo
18          super("Calculadora");
19      
20      }
```

Instanciação dos Elementos de Tela

- Instancie os botões, campos de texto e etiquetas criados; veja que cada construtor é de um jeito.
  - O do botão (JButton) recebe o texto do botão, assim como o da etiqueta (JLabel) recebe o texto da etiqueta.
  - O campo texto (JTextField), por sua vez, recebe o tamanho do campo.
  - Veja a documentação da API do Java para outras opções de construtor.

```
19          //instanciar os widgets
20          botao = new JButton("Soma");
21          texto = new JTextField("0", 10);
22          etiqueta = new JLabel("Valor: ");
```

Gerenciador de Lauyout

- A tela de um JFrame, além da moldura, tem um painel, que é onde serão colocados os elementos de interface.
- Todo painel tem que ter o seu **gerenciador de layout**.
- Se não tiver, todo elemento de tela colocado neste painel irá assumir o tamanho do painel e, como consequência, somente o último elemento adicionado irá aparecer.
- No caso do JFrame, o painel é o Container, que você pega usando o método getContentPane().

FlowLayout

- Este é o gerenciador de layout usado neste exemplo.
- Há outros, que mencionarei mais tarde.
- O comportamento do FlowLayout é o seguinte: ele vai organizando os elemento de tela que vão sendo adicionados, da esquerda para direita, um depois do outro.
- Se acabar o espaço horizontal da tela, então ele pula a linha.

```
23          //pega o container (ou painel)
24          Container caixa = getContentPane();
25          //configura o gerenciador de layout
26          caixa.setLayout(new FlowLayout());
```

Adicionar os elementos de tela

- Use o comando add(objeto) para adicionar os elementos no painel (neste caso, container).

- O gerenciador de layout é quem irá organizá-los.

```
27          //adiciona na tela na ordem em que quer que apareca
28          caixa.add(etiqueta);
29          caixa.add(texto);
30          caixa.add(botao);
```

Registrar o listener

- Lembre-se, o listener é quem ouve os eventos e reage a eles;
- Esta tela que estamos fazendo, por implementar a interface ActionListener, pode ser registrada como ouvinte de eventos de botão.

```
31          //registra este objeto como listener
32          botao.addActionListener(this);
```

Ajustes finais da tela

- configure o tamanho (largura x e altura y);
- defina o que fazer quando o usuário clicar no X para fechar - neste caso, encerrar o sistema (mas nem sempre!);
- deixe a tela visível (sempre deve ser a última)

```
33          //configura ajustes finais
34          //configura o tamanho inicial da tela
35          setSize(200,100);
36          //encerra a aplicacao quando clica o xis
37          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
38          //torna a janela visivel
39          setVisible(true);
```

Implementar o método que responderá aos cliques no botão.

- É o actionPerformed.
- No exemplo a seguir, ele verifica quem gerou o evento usando o getSource() e então faz alguma coisa;
- Neste caso, soma 10 ao valor que está o campo texto.
- Note que o método String getText() pega o texto que está digitado em um JTextField, e o setText(String) muda o texto.

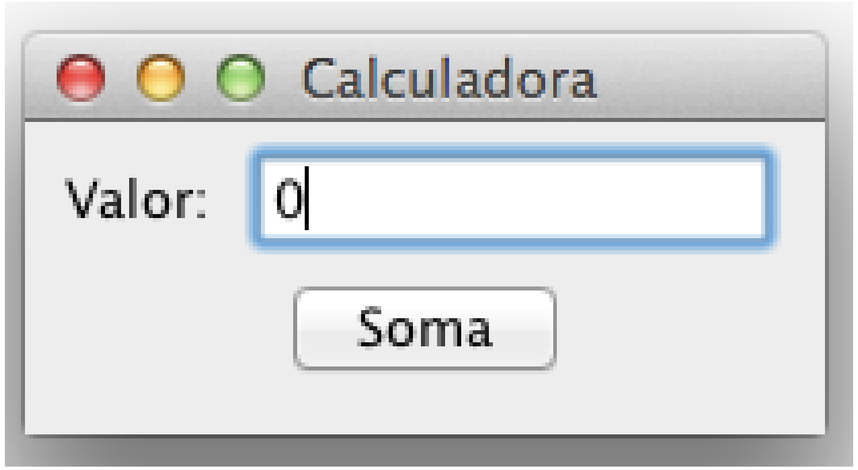
```
42      public void actionPerformed(ActionEvent e){
43          if(e.getSource()==botao){
44              int valor = Integer.parseInt(texto.getText());
45              valor+=10;
46              texto.setText(""+valor);
47          }
48      }
```

Código completo

```java
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.JTextField;
import javax.swing.JLabel;
import java.awt.Container;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class TelaCalculadora extends JFrame implements ActionListener{

   private JButton botao; // botao
   private JTextField texto; //campo texto de uma linha
   private JLabel etiqueta; //etiqueta de nome do campo

   public TelaCalculadora(){
   //chamar construtor da superclasse e configurar o titulo
      super("Calculadora");
      //instanciar os widgets
      botao = new JButton("Soma");
      texto = new JTextField("0", 10);
      etiqueta = new JLabel("Valor: ");
      //pega o container (ou painel)
      Container caixa = getContentPane();
      //configura o gerenciador de layout
      caixa.setLayout(new FlowLayout());
      //adiciona na tela na ordem em que quer que apareca
      caixa.add(etiqueta);
      caixa.add(texto);
      caixa.add(botao);
      //registra este objeto como listener
      botao.addActionListener(this);
      //configura ajustes finais
      //configura o tamanho inicial da tela
      setSize(200,100);
      //encerra a aplicacao quando clica o xis
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      //torna a janela visivel
      setVisible(true);
   }

   public void actionPerformed(ActionEvent e){
      if(e.getSource()==botao){
         int valor = Integer.parseInt(texto.getText());
         valor+=10;
         texto.setText(""+valor);
      }
   }
}
```

```java
public class Calculadora{

   public static void main(String[] args){
      //instancia o JFrame
      TelaCalculadora tela = new TelaCalculadora();
   }
}
```

Resultado



Outros gerenciadores de layout

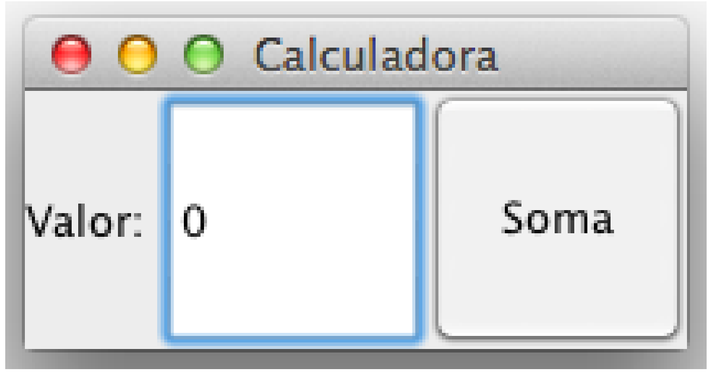
BorderLayout

- O Border divide a tela em 5 espaços: central, norte (topo), sul (rodapé), leste (direita) e oeste (esquerda).
- Para adicionar coisas no Border você usa o mesmo comando add, só que desta vez com mais um parâmetro, a posição em que quer colocar.
- container.add(objeto, BorderLayout.SOUTH);
- container.add(outroObjeto, BorderLayout.EAST);

O código fica assim:

```
25          //configura o gerenciador de layout
26          caixa.setLayout(new BorderLayout());
27          caixa.add(etiqueta, BorderLayout.WEST);
28          caixa.add(texto, BorderLayout.CENTER);
29          caixa.add(botao, BorderLayout.EAST);
```

A tela fica assim:



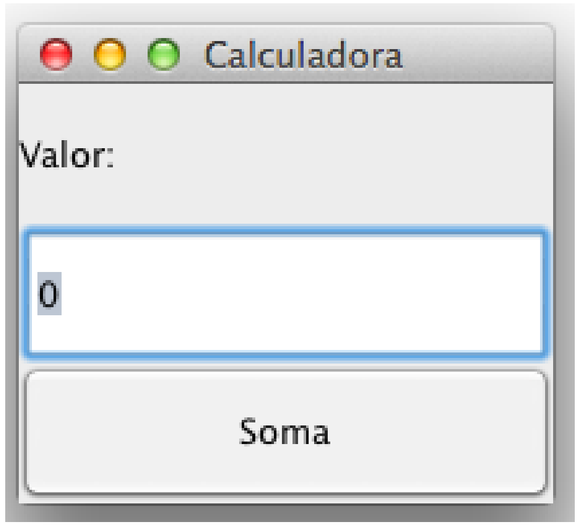
GridLayout

- Divide a tela em uma matriz de linhas e colunas.
- Por exemplo, ao invocar o construtor new GridLayout (3,1) você está criando uma matriz de 3 linhas e 1 coluna.
- Use o add para colocar os elementos na tela.
- A cada add, o gerenciador irá adicionar o elemento de tela em uma célula da matriz, começando do topo, à esquerda, e preenchendo a linha até o final. Aí pula linha e volta para a esquerda.

O código fica assim:

```
26          caixa.setLayout(new GridLayout(3,1));
27          //adiciona na tela na ordem em que quer que apareca
28          caixa.add(etiqueta);
29          caixa.add(texto);
30          caixa.add(botao);
```

A tela fica assim:



Combinação de Layouts

- Para usar mas de um gerenciador em um mesmo JFrame e construir layouts mais elaborados, faça o seguinte:
  - Crie um JPane e dê a ele o gerenciador de layout desejado.
  - Adicione os elementos gráficos neste JPane.
  - Adicione o JPane ao container do JFrame.

O código fica assim:

```
29          caixa.setLayout(new BorderLayout());
30          //cria os paineis secundarios
31          JPanel painelSul = new JPanel(new FlowLayout());
32          JPanel painelCentro = new JPanel(new GridLayout(1,2));
33          //adiciona os widgets nos paineis secundarios
34          painelSul.add(botao);
35          painelCentro.add(etiqueta);
36          painelCentro.add(texto);
37          //adiciona os paineis secundarios no principal
38          caixa.add(painelSul, BorderLayout.SOUTH);
39          caixa.add(painelCentro, BorderLayout.CENTER);
```

A tela fica assim: