

Programação com interfaces gráficas

Sérgio Soares
scbs@cin.ufpe.br

Até aqui...

- Criamos a infra-estrutura para a aplicação bancária
 - classes que representam os tipos de conta
 - classe que implementa regras de negócio
 - interface e classe que armazena contas
- Mas ainda não temos uma interface **decente** com o usuário

Programação Orientada a Objetos - Programação com interfaces gráficas

2

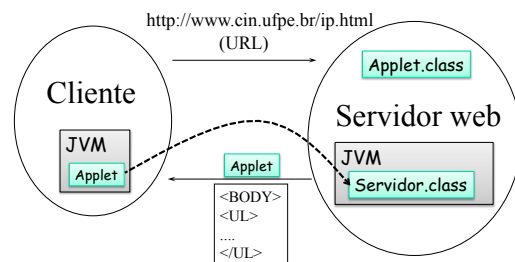
Interfaces gráficas em Java

- Por ser em Java as interfaces gráficas criadas estão aptas a serem executadas pela internet
- Um **applet** é um programa Java que executa na internet
 - Subclasse de `java.applet.Applet`

Programação Orientada a Objetos - Programação com interfaces gráficas

3

Clientes e Servidores WWW



Programação Orientada a Objetos - Programação com interfaces gráficas

4

Applets versus HTML+CGI+Javascript

- Applets eliminam gargalos:
 - cliente fala direto com o servidor da aplicação (usando objetos ao invés de strings)
 - qualquer tipo de processamento pode ser feito no cliente
 - carga do servidor pode ser distribuída
- Java dá suporte a princípios de engenharia de software
- Problemas: eficiência e portabilidade

Programação Orientada a Objetos - Programação com interfaces gráficas

5

WWW e Java: Interação

- *Applets* são programas Java disponibilizados via WWW, através de uma página HTML

```
<HTML>
...
<applet codebase="http://www.cin.ufpe.br/ip"
        code="AppletBanco.class"
        width=10 height=90>
</applet>
</HTML>
```

Programação Orientada a Objetos - Programação com interfaces gráficas

6

AppletBanco:

Atributos (como qualquer outra classe Java)

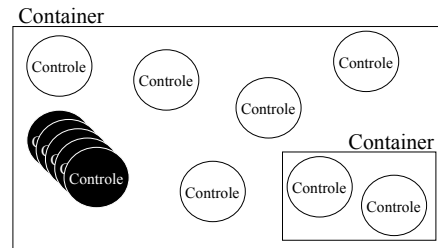
```
public class AppletBanco extends Applet {
    //Ligação com o banco
    private Banco fachada = null;

    //Controles
    Button buttonProcurar = new Button();
    Label labelValor = new Label();
    TextField textFieldVal = new TextField();
    Button buttonDebito = new Button();
    ...
}
```

Programação Orientada a Objetos - Programação com interfaces gráficas

7

Container e Controles



Programação Orientada a Objetos - Programação com interfaces gráficas

8

Componentes do AWT

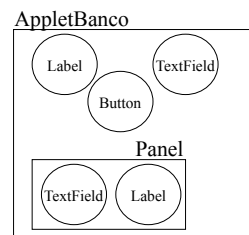
- | | |
|--|--|
| <ul style="list-style-type: none"> ■ Containers • Container • Panel • Window • Frame • Dialog • Applet • ... | <ul style="list-style-type: none"> ■ Controles • Button • Canvas • Label • TextField • Choice • List • ... |
|--|--|

Mesmos componentes do Swing acrescentando J no início do nome

Programação Orientada a Objetos - Programação com interfaces gráficas

9

Container e Controles



Programação Orientada a Objetos - Programação com interfaces gráficas

10

Container e Controles



Programação Orientada a Objetos - Programação com interfaces gráficas

11

Mas como os controles e os componentes se comunicam?

- Comunicação por eventos
 - editor vs assinante (*publisher vs subscriber*)
- Editora vs. Assinante do jornal
 - Botão é a editora
 - Avisa aos assinantes quando for clicado
 - Applet é o assinante
 - Pede ao botão para ser avisado

Programação Orientada a Objetos - Programação com interfaces gráficas

12

Em Java

- Objetos se comunicam através de métodos
- Eventos = Comunicação assíncrona
 - O applet chama um método do botão para assinar o serviço
 - O botão chama um método do applet para avisar quando for clicado

Programação Orientada a Objetos - Programação com interfaces gráficas

13

Vamos implementar um botão

```
public class Botao {  
    private Object[] assinantes;  
    private int indice;  
    ...  
    public void assinar(Object o) {  
        assinantes[indice] = o;  
        indice++;  
    }  
    private void avisa() {  
        for(int i=0; i < indice; i++) {  
            assinantes[i].botaoClicado();  
        }  
    }  
}
```

Tem esse método em Object? 14

Pre_

Mas espera...

- O botão tem de chamar sempre um mesmo método dos assinantes
 - Como obrigar que todo assinante tenha esse mesmo método?
- Lembram de interfaces?

Programação Orientada a Objetos - Programação com interfaces gráficas

15

Uma interface para assinante

```
public interface Assinante {  
    void botaoClicado();  
}
```

Vamos mudar o botão para aceitar apenas assinantes...

Programação Orientada a Objetos - Programação com interfaces gráficas

16

Vamos implementar um botão

```
public class Botao {  
    private Assinante[] assinantes;  
    private int indice;  
    ...  
    public void assinar(Assinante o) {  
        assinantes[indice] = o;  
        indice++;  
    }  
    private void avisa() {  
        for(int i=0; i < indice; i++) {  
            assinantes[i].botaoClicado();  
        }  
    }  
}
```

Agora sim! 17

Programaç

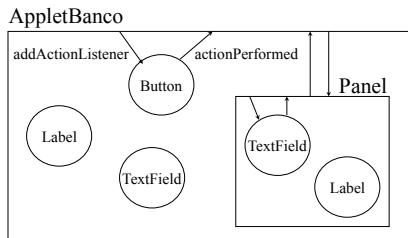
Mas em Java...

- Os nomes são ligeiramente diferentes
 - Assinante é ActionListener
 - botaoClicado é actionPerformed

Programação Orientada a Objetos - Programação com interfaces gráficas

18

Comunicação entre Containers e Controles: Eventos



Programação Orientada a Objetos - Programação com interfaces gráficas

19

AppletBanco: Inicialização

```

public void init() {
    ...
    javax.swing.JLabel lb_title = new JLabel();
    jContentPane = new javax.swing.JPanel();
    jContentPane.setLayout(null);
    lb_title.setBounds(63, 14, 178, 23);
    lb_title.setText("Applet Banco");
    jContentPane.add(lb_title, null);
    jContentPane.add(getJTextField(), null);
    jContentPane.add(getJTextField2(), null);
    jContentPane.add(getJButton(), null);
    jContentPane.add(getJButton2(), null);
    ...
}
    
```

Programação Orientada a Objetos - Programação com interfaces gráficas

20

AppletBanco: Tratamento de Eventos

```

private JButton getJButton2() {
    if (bt_creditar == null) {
        bt_creditar = new JButton();
        bt_creditar.setBounds(197, 79, 86, 25);
        bt_creditar.setText("Creditar");
        bt_creditar.addActionListener(assinante);
    }
    return bt_creditar;
}
    
```

Na verdade o código gerado é um pouco mais confuso

Programação Orientada a Objetos - Programação com interfaces gráficas

21

AppletBanco: Tratamento de Eventos

```

new java.awt.event.ActionListener() {
    ...
    bt_creditar.setBounds(197, 79, 86, 25);
    bt_creditar.setText("Creditar");
    bt_creditar.addActionListener(this);
    ...
    public void actionPerformed(ActionEvent e) {
        creditar();
    }
}
    
```

Programação Orientada a Objetos - Programação com interfaces gráficas

22

```

private void creditar() {
    String numero = this.tf_numero.getText();
    String v = tf_valor.getText();
    try {
        double valor = new Double(v).doubleValue();
        banco.creditar(numero, valor);
        JOptionPane.showMessageDialog(this,
            "Crédito executado com sucesso!");
    } catch (ContaNaoEncontradaException e) {
        JOptionPane.showMessageDialog(this,
            "Conta " + numero + " não existe!");
    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(this,
            "Digite um número!");
    } finally {
        ...
    }
}
    
```

P }

23

Applets: Aspectos de Segurança

- Applets devem satisfazer várias restrições, impostas pelos folheadores:
 - não ter acesso a arquivos do cliente
 - só se conectar com o servidor de origem
 - não usar métodos nativos
- Restrições podem ser eliminadas para applets assinados e transmitidos de forma segura!

Programação Orientada a Objetos - Programação com interfaces gráficas

24

Exercício

- Vamos executar o roteiro em
<http://www.cin.ufpe.br/~scbs/gui/>