

## Aula 22: Interface Gráfica

Professor(a): Virgínia Fernandes Mota

<http://www.dcc.ufmg.br/~virginiaferm>

TECNOLOGIAS DE PROGRAMAÇÃO - SETOR DE INFORMÁTICA



- Ao término desta aula, você será capaz de:
  - desenvolver aplicações em Java com interface gráfica.

- Vamos começar nossa interface gráfica pela classe chamada *PrimeiraUI* que estará no pacote *gui* terá um método *montaTela* que desenha a tela em Swing e um método *main* que apenas dispara a montagem da tela:

```
1 public class PrimeiraUI {  
2  
3     public static void main(String[] args) {  
4         new PrimeiraUI().montaTela();  
5     }  
6  
7     private void montaTela() {  
8         // TODO ...  
9  
10    }  
11 }
```

- É boa prática quando usamos Swing quebrar a declaração dos componentes em pequenos métodos que fazem tarefas simples.
- Vamos usar os componentes e conceitos vistos até aqui para criar nossa tela organizadamente.
- Para começar, vá ao método `montaTela()` e preencha-o com as seguintes chamadas aos métodos que formarão a tela.

```
1 public void montaTela() {  
2     preparaJanela();  
3     preparaPainelPrincipal();  
4     preparaBotaoCarregar();  
5     preparaBotaoSair();  
6     mostraJanela();  
7 }
```

- Preencha o método `preparaJanela` com a criação do `JFrame` e, como essa é a principal janela da aplicação, configure-a para terminar o programa Java quando a fecharem.
- Aproveite para preencher também o método `mostraJanela`, organizando os componentes (`pack`), configurando o tamanho (`setSize`) e mostrando-a (`setVisible`).
- Como é necessário utilizar o `JFrame` no método `mostraJanela`, será preciso que a janela seja um atributo em vez de uma variável local.

# Interface Gráfica

```
1 public class PrimeiraUI {
2     private JFrame janela;
3
4     // main e montaTela
5
6     private void preparaJanela() {
7         janela = new JFrame("Minha Interface");
8         janela.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
9     }
10
11     // outros metodos prepara...
12
13     private void mostraJanela() {
14         janela.pack();
15         janela.setSize(540, 540);
16         janela.setVisible(true);
17     }
18 }
```

- Agora, só falta preencher os métodos que preparam os componentes da tela: o painelPrincipal, que conterà os botões, e cada um dos botões com seus ActionListeners.
- Lembre-se, também, que em cada um desses métodos será necessário dar new no componente e adicioná-lo ao seu componente pai.
- Note que, como será necessário adicionar os botões ao painelPrincipal, este também deve ser criado como um atributo.

```
1 private void preparaPainelPrincipal() {  
2     painelPrincipal = new JPanel();  
3     janela.add(painelPrincipal);  
4 }
```



# Interface Gráfica

```
1 private void preparaBotaoCarregar() {
2     JButton botaoCarregar = new JButton("Carregar Arquivo");
3     botaoCarregar.addActionListener(new ActionListener() {
4         @Override
5         public void actionPerformed(ActionEvent e) {
6             new EscolhedorArquivo().escolhe();
7         }
8     });
9     painelPrincipal.add(botaoCarregar);
10 }
11
12 private void preparaBotaoSair() {
13     JButton botaoSair = new JButton("Sair");
14     botaoSair.addActionListener(new ActionListener() {
15         @Override
16         public void actionPerformed(ActionEvent e) {
17             System.exit(0);
18         }
19     });
20     painelPrincipal.add(botaoSair);
21 }
```

- Mas isso nós já vimos, será que é preciso sempre escrever tudo isso para criar cada interface, cada botão?

- Vamos criar uma nova aplicação Java.
- Novo > Form JFrame.
- Quando adicionamos o contêiner JFrame, o IDE abriu o form!
- Estamos trabalhando na GUI Builder.

- **Área de Design.** A janela principal do GUI Builder para criar e editar forms de GUI do Java. O botão Código-fonte da barra de ferramenta permite exibir o código-fonte de uma classe; o botão Design permite a exibição gráfica dos componentes da GUI; o botão Histórico permite o acesso ao histórico local de alterações do arquivo. Os botões adicionais da barra de ferramentas oferecem acesso conveniente aos comandos comuns, como escolher entre os modos Seleção e Conexão, alinhamento de componentes, definição do comportamento de dimensionamento automático e exibição de forms.

- **Navegador.** Fornece uma representação de todos os componentes, ambos visual e não-visuais na sua aplicação, como uma hierarquia de árvore. O Navegador também oferece apresentação visual sobre qual componente da árvore está sendo editado no GUI Builder, bem como permite organizar componentes nos painéis disponíveis.
- **Paleta.** Uma lista customizável de componentes disponíveis contendo guias para jfc/swing e JavaBeans componentes awt, bem como gerentes de layout. Além disso, é possível criar, remover e reorganizar as categorias exibidas na Paleta utilizando o personalizador.
- **Janela Propriedades.** Exibe as propriedades do componente atualmente selecionado no navegador GUI Builder, janela janela, projetos, ou arquivos janela.

- 1- Implemente uma janela de cadastro de clientes:
  - Campos: Nome, Rua, Número, Complemento, Bairro, Cep, Cidade, Estado.
  - Botões: Gravar e Cancelar.
  - Sempre que um campo de texto receber o foco, seu fundo deve ficar amarelo.
  - Sempre que um botão receber o foco, a cor da fonte deve mudar para vermelho.
  - Quando o botão Gravar for pressionado, o programa deve: exibir uma mensagem de confirmação, Limpar o texto de todos os campos e enviar o foco para o primeiro campo.
  - Quando o botão Cancelar for pressionado, a janela deve ser fechada.

- 2 - Implemente uma janela contendo 5 botões, 4 na parte superior e 1 na inferior (Fechar).
  - Quando aberta, o painel de conteúdo da janela deve apresentar o fundo preto.
  - Os 4 botões da parte superior devem permitir a mudança de cor do painel: Vermelho, verde, azul e preto.
  - Quando o botão Fechar for pressionado, um diálogo com botões Sim e Não deve ser exibido: Tem certeza que deseja fechar?

# Na próxima aula...

## Exercícios



- 3 - Implemente uma janela de cadastro de contatos:
  - Campos: Nome, Telefone Residencial, Telefone Celular, Email.
  - Botões: Gravar e Cancelar.
  - Sempre que um campo de texto receber o foco, seu fundo deve ficar azul ciano.
  - Sempre que um botão receber o foco, a cor da fonte deve mudar para azul escuro.
  - Sempre que o mouse passar sobre um campo, este componente deve requisitar o foco para si.
  - Quando o botão Gravar for pressionado, o programa deve: exibir uma mensagem de confirmação, Limpar o texto de todos os campos e enviar o foco para o primeiro campo.
  - Se algum campo estiver vazio, o botão Gravar deve: exibir uma mensagem de erro, enviar o foco para o primeiro campo vazio.
  - Quando o botão Cancelar for pressionado, a janela deve ser fechada.

- 4 - Implemente uma janela contendo um menu suspenso com opções de acesso aos exercícios anteriores.
  - O menu Sistema deve conter uma opção para sair.
  - O menu Eventos contém o acesso aos exercícios.
  - Defina atalhos para todas as opções de menu.

# Na próxima aula...

Exercícios e Prova