

1 Introdução a Interfaces gráficas

O pacote **javax.swing** e seus sub-pacotes possuem classes úteis para a criação de interfaces gráficas para o usuário (GUI: *Graphical User Interface*). Até então, fizemos uso de uma delas: `JOptionPane`. Neste material estudaremos a criação de interfaces gráficas utilizando diferentes recursos, indo além da simples exibição de caixas de diálogo.

2 Desenvolvimento

2.1 (A classe JFrame) A criação de uma interface gráfica (uma tela) envolve o uso da classe `JFrame`, que faz parte do pacote `javax.swing`. A palavra **frame** significa **moldura**. A ideia é construirmos uma moldura e então, adicionarmos conteúdo a ela, como botões, campos textuais, tabelas, menus etc. Um `JFrame` possui um **painel de conteúdo** (do inglês **content pane**). Trata-se de um objeto capaz de armazenar outros objetos. Não adicionamos conteúdo diretamente a um `JFrame`. Uma vez que tenhamos um `JFrame` em mãos, obtemos uma referência ao seu painel de conteúdo e adicionamos conteúdo a ele.

2.2 (Hello, Swing) Vamos criar um primeiro exemplo para ilustrar o uso de duas classes do pacote Swing: `JFrame` e `JLabel`. Criaremos uma aplicação que exibe uma tela com o texto **Hello, Swing**.

- Crie um novo projeto e dê a ele um nome que esteja de acordo com o contexto estudado.

- Crie a classe `HelloSwing` e adicione a ela um método auxiliar, como na Listagem 2.2.1.

Listagem 2.2.1

```
public class HelloSwingTela{  
    public static void criarTela() {  
  
    }  
}
```

- Para instanciar um `JFrame`, podemos utilizar um construtor que recebe uma `String`. Ela será utilizada como título da tela. Na Listagem 2.2.2, além de fazermos isso, também utilizamos um método para especificar qual o comportamento esperado quando a tela for fechada pelo usuário. Essa operação é necessária pois, por padrão, caso o usuário feche a aplicação, a tela somente ficará invisível, mas programa continuará em execução.

Listagem 2.2.2

```
public static void criarTela() {  
    JFrame tela = new JFrame ("Hello, Swing!!!");  
    tela.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
}
```

- A finalidade da classe **`JLabel`** é exibir conteúdo textual que o usuário não pode editar. Para fazer isso, é preciso construir um objeto desse tipo, informar o texto a ser exibido no construtor e adicionar o objeto ao painel de conteúdo do `JFrame`. Veja a Listagem 2.2.3.

Listagem 2.2.3

```
//constroi um JLabel  
JLabel helloSwingLabel = new JLabel ("Hello, Swing!!!!!!!!");  
//obtem o painel de conteúdo  
Container painelDeConteudo = tela.getContentPane();  
//adiciona o JLabel ao painel de conteúdo  
painelDeConteudo.add(helloSwingLabel);
```

- Por padrão, objetos `JFrame` são invisíveis. Precisamos tornar o objeto que criamos visível. Isso pode ser feito com o método **`setVisible`**. Além disso, usamos o método **`pack`** para fazer com que sua largura e altura se ajustem de acordo com as medidas de seu conteúdo. Veja a Listagem 2.2.4.

Listagem 2.2.4

```
//ajusta largura e altura da tela conforme seu conteúdo  
tela.pack();  
//torna a tela visível  
tela.setVisible(true);
```

- A fim de colocar o programa em execução, precisamos criar o método `main`. Ocorre que o método `main` é executado pelo fluxo principal (que chamamos de thread principal). Códigos que envolvem a atualização de componentes visuais devem ser

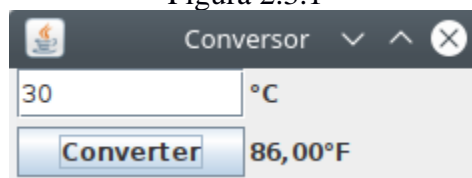
executados em um fluxo de execução à parte, que executa de maneira independente do principal e é reservado somente para isso. A programação concorrente foge do escopo da disciplina, porém é importante ter essa noção básica. A forma correta de se colocar em execução o nosso programa é exibida na Listagem 2.2.5. Não se preocupe com alguns detalhes. Isso se aprende com o tempo. Por hora, lembre-se de iniciar seu programa com interfaces gráficas sempre dessa forma.

Listagem 2.2.5

```
public static void main(String[] args) {  
    SwingUtilities.invokeLater() -> {  
        criarTela();  
    });  
}
```

2.3 (Conversão de Celsius para Fahrenheit) No Brasil, a temperatura é, em geral, medida em graus Celsius. Em outros países, é comum o uso de graus Fahrenheit. Vamos escrever um programa que permite ao usuário informar um valor em graus Celsius e fazer a conversão para Fahrenheit. Sua tela será parecida com a exibida pela Figura 2.3.1.

Figura 2.3.1



- Comece criando um novo projeto/programa com um nome adequado para o contexto.
- Crie a classe da Listagem 2.3.1. Seu método criarTela, como o nome sugere, será o responsável por criar a tela da aplicação. Ele já instancia um JFrame.

Listagem 2.3.1

```
public class ConversorDeTemperatura {  
  
    public static void criarTela(){  
        JFrame tela = new JFrame ("Conversor");  
  
    }  
}
```

- Utilizaremos as classes da Tabela 2.3.1 para montar a tela.

Tabela 2.3.1

Classe	Finalidade
JButton	Botões
JLabel	Componentes textuais não editáveis pelo usuário
JTextField	Componentes textuais editáveis pelo usuário

A construção de cada um deles é exibida na Listagem 2.3.2.

Listagem 2.3.2

```
public static void criarTela(){
    JFrame tela = new JFrame ("Conversor");
    JTextField celsiusTextField = new JTextField (10);
    JLabel celsiusLabel = new JLabel ("\u00B0C");
    JButton convertButton = new JButton ("Converter");
    JLabel valorConvertidoLabel = new JLabel ("");
}
```

- Para que apareçam na tela, os componentes precisam ser adicionados ao painel de conteúdo do JFrame. Isso é feito na Listagem 2.3.3.

Listagem 2.3.3

```
public static void criarTela(){
    JFrame tela = new JFrame ("Conversor");
    JTextField celsiusTextField = new JTextField (10);
    JLabel celsiusLabel = new JLabel ("\u00B0C");
    JButton convertButton = new JButton ("Converter");
    JLabel valorConvertidoLabel = new JLabel ("");
    Container painelDeConteudo = tela.getContentPane();
    painelDeConteudo.setLayout (new GridLayout (2, 4, 2, 4));
    painelDeConteudo.add(celsiusTextField);
    painelDeConteudo.add(celsiusLabel);
    painelDeConteudo.add(convertButton);
    painelDeConteudo.add(valorConvertidoLabel);
}
```

- O botão é um componente que pode sofrer um **evento** de interesse. Ele pode ser **clicado**. Desejamos especificar um método que será executado quando esse evento acontecer, pois é nesse momento que a conversão deve acontecer. O tratamento de eventos com Swing é feito por meio da aplicação do padrão de projetos **Observer**. Veja uma possível implementação na Listagem 2.3.4.

Listagem 2.3.4

```
public static void criarTela(){
    JFrame tela = new JFrame ("Conversor");
    JTextField celsiusTextField = new JTextField (10);
    JLabel celsiusLabel = new JLabel ("\u00B0C");
    JButton convertButton = new JButton ("Converter");
    JLabel valorConvertidoLabel = new JLabel ("");
    Container painelDeConteudo = tela.getContentPane();
    painelDeConteudo.setLayout (new GridLayout (2, 4, 2, 4));
    painelDeConteudo.add(celsiusTextField);
    painelDeConteudo.add(celsiusLabel);
    painelDeConteudo.add(convertButton);
    painelDeConteudo.add(valorConvertidoLabel);

    convertButton.addActionListener((e) -> {
        double celsius = Double.parseDouble(
            celsiusTextField.getText()
        );
        double fahrenheit = celsius / 5 * 9 + 32;
        valorConvertidoLabel.setText(
            String.format("%.2f\u00B0F", fahrenheit)
        );
    });
}
```

- Finalmente, fazemos os ajustes para que o JFrame seja exibido adequadamente, como na Listagem 2.3.5.

Listagem 2.3.5

```
public static void criarTela(){
    JFrame tela = new JFrame ("Conversor");
    JTextField celsiusTextField = new JTextField (10);
    JLabel celsiusLabel = new JLabel ("\u00B0C");
    JButton convertButton = new JButton ("Converter");
    JLabel valorConvertidoLabel = new JLabel ("");
    Container painelDeConteudo = tela.getContentPane();
    painelDeConteudo.setLayout (new GridLayout (2, 4, 2, 4));
    painelDeConteudo.add(celsiusTextField);
    painelDeConteudo.add(celsiusLabel);
    painelDeConteudo.add(convertButton);
    painelDeConteudo.add(valorConvertidoLabel);

    convertButton.addActionListener((e) -> {
        double celsius = Double.parseDouble(
            celsiusTextField.getText()
        );
        double fahrenheit = celsius / 5 * 9 + 32;
        valorConvertidoLabel.setText(
            String.format("%.2f\u00B0F", fahrenheit)
        );
    });

    //ajusta largura e altura de acordo com o conteúdo
    tela.pack();
    //centraliza
    tela.setLocationRelativeTo(null);
    //altera comportamento padrão do botão fechar
    tela.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //torna a tela visível
    tela.setVisible(true);
}
```

- Para executar a aplicação, criamos um método main e utilizamos o método `invokeLater` de `SwingUtilities` para chamar o método `criarTela` apropriadamente. Veja a Listagem 2.3.6.

Listagem 2.3.6

```
public static void main(String[] args) {
    SwingUtilities.invokeLater () -> {
        criarTela();
    });
}
```

Exercícios

1. Implemente um programa com interface gráfica como o conversor de temperatura. Ele deve permitir que o usuário digite um único valor numérico. Esse valor, que será digitado em centímetros, pode ser convertido para **metros**, **milímetros** e **quilômetros**. Deve haver três botões na aplicação, um para cada possível conversão. Deve haver somente um campo de resultado.
2. Ajuste a aplicação de login e senha para que o usuário seja validado em uma base de dados que tenha uma tabela de usuários.

Referências

DEITEL, P. e DEITEL, H. **Java Como Programar**. 8ª Edição. São Paulo, SP: Pearson, 2010.

LOPES, A. e GARCIA, G. **Introdução à Programação – 500 Algoritmos Resolvidos**. 1ª Edição. São Paulo, SP: Elsevier, 2002.