

Centro Universitário UNA

Programação de soluções computacionais – 2024/2

Professor: Wesley

Atividades: Prática 11 – Java Swing / FX

Grupo: 2 - Reciclo

Programação de Soluções Computacionais – JavaFX

Introdução

JavaFX é uma biblioteca gráfica que possibilita a criação de interfaces de usuário (GUI) de forma visualmente atraente e moderna para aplicações Java, especialmente no ambiente desktop. Desenvolvida inicialmente pela Sun Microsystems e depois mantida pela Oracle, o JavaFX surgiu em 2008 como uma evolução para criar interfaces mais ricas e dinâmicas do que as oferecidas pelo Swing, uma das primeiras bibliotecas gráficas de Java.

Em 2011, JavaFX foi reescrito inteiramente em Java, o que facilitou a integração com o restante da plataforma. Graças a isso, ele se tornou uma opção poderosa para desenvolvedores que buscavam uma experiência de desenvolvimento mais consistente e moderna em Java. Apesar de não ser tão popular quanto frameworks para desenvolvimento web, o JavaFX ainda tem um papel importante no desenvolvimento de aplicações corporativas, ferramentas de visualização e simuladores educacionais que exigem interfaces gráficas complexas e interativas.

Exemplos de Uso

Ferramentas Corporativas e de Visualização de Dados

JavaFX é amplamente utilizado em ferramentas de visualização de dados e dashboards empresariais. Com ele, é possível criar gráficos interativos que ajudam no acompanhamento e análise de dados.

Exemplo: Criando um gráfico de barras simples.

```

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.chart.BarChart;
import javafx.scene.chart.CategoryAxis;
import javafx.scene.chart.NumberAxis;
import javafx.scene.chart.XYChart;
import javafx.stage.Stage;

// Classe principal da aplicação JavaFX
public class ExemploGraficoBarras extends Application {
    @Override
    public void start(Stage palco) {
        // Define o título da janela
        palco.setTitle("Exemplo de Gráfico de Barras");

        // Define o eixo X como categórico e o eixo Y como numérico
        final CategoryAxis eixoX = new CategoryAxis();
        final NumberAxis eixoY = new NumberAxis();
        eixoX.setLabel("Categorias");
        eixoY.setLabel("Valores");

        // Cria o gráfico de barras com os eixos definidos
        final BarChart<String, Number> graficoBarras = new BarChart<>(eixoX, eixoY);
        graficoBarras.setTitle("Vendas Anuais");

        // Define a série de dados para o gráfico
        XYChart.Series<String, Number> serieDados = new XYChart.Series<>();
        serieDados.setName("Ano 2024"); // Nome da série de dados
        serieDados.getData().add(new XYChart.Data<>("Janeiro", 200)); // Valor de janeiro
        serieDados.getData().add(new XYChart.Data<>("Fevereiro", 150)); // Valor de fevereiro
        serieDados.getData().add(new XYChart.Data<>("Março", 250)); // Valor de março

        // Adiciona a série de dados ao gráfico
        graficoBarras.getData().add(serieDados);

        // Define a cena e adiciona o gráfico de barras nela
        Scene cena = new Scene(graficoBarras, 800, 600);
        palco.setScene(cena);

        // Exibe o palco (janela)
        palco.show();
    }

    public static void main(String[] args) {
        launch(args); // Inicia a aplicação JavaFX
    }
}

```

codesnap.dev

Aplicações Educacionais

JavaFX é também muito útil em ambientes educacionais, sendo usado para criar simuladores e interfaces que facilitam o aprendizado. Suas capacidades de animação permitem a criação de visuais dinâmicos e interativos.

Exemplo: Criando uma animação de uma bolinha se movendo na tela.

```

import javafx.animation.TranslateTransition;
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.stage.Stage;
import javafx.util.Duration;
import javafx.scene.Group;

// Classe principal da aplicação JavaFX
public class ExemploAnimacao extends Application {
    @Override
    public void start(Stage palco) {
        // Cria um círculo com raio 50 e cor azul
        Circle bolinha = new Circle(50, Color.BLUE);
        bolinha.setCenterX(100); // Posição inicial X da bolinha
        bolinha.setCenterY(100); // Posição inicial Y da bolinha

        // Configura a transição de movimento da bolinha
        TranslateTransition transicao = new TranslateTransition();
        transicao.setDuration(Duration.seconds(2)); // Duração de cada ciclo da animação
        transicao.setNode(bolinha); // Define a bolinha como o nó que será animado
        transicao.setByX(300); // Movimento de 300 unidades no eixo X
        transicao.setCycleCount(TranslateTransition.INDEFINITE); // Animação em loop
        transicao.setAutoReverse(true); // Faz a bolinha voltar ao ponto inicial
        transicao.play(); // Inicia a animação

        // Adiciona a bolinha ao grupo e cria a cena
        Group raiz = new Group(bolinha);
        Scene cena = new Scene(raiz, 400, 200);
        palco.setTitle("Exemplo de Animação JavaFX");
        palco.setScene(cena);
        palco.show(); // Exibe o palco
    }

    public static void main(String[] args) {
        launch(args); // Inicia a aplicação JavaFX
    }
}

```

codesnap.dev

Aplicações de Prototipagem de Interface

JavaFX também é popular para criar protótipos de interfaces desktop, pois oferece componentes prontos como botões, caixas de texto e layouts, permitindo desenvolver interfaces rapidamente.

Exemplo: Criando uma interface de login simples.

```

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;

// Classe principal da aplicação JavaFX
public class ExemploTelaLogin extends Application {
    @Override
    public void start(Stage palco) {
        // Cria rótulos e campos de entrada para usuário e senha
        Label rotuloUsuario = new Label("Usuário:");
        TextField campoUsuario = new TextField();

        Label rotuloSenha = new Label("Senha:");
        PasswordField campoSenha = new PasswordField();

        // Cria o botão de login
        Button botaoLogin = new Button("Login");

        // Cria o layout e organiza os elementos em uma grade
        GridPane painelGrade = new GridPane();
        painelGrade.add(rotuloUsuario, 0, 0); // Adiciona rótulo de usuário
        painelGrade.add(campoUsuario, 1, 0); // Adiciona campo de entrada de usuário
        painelGrade.add(rotuloSenha, 0, 1); // Adiciona rótulo de senha
        painelGrade.add(campoSenha, 1, 1); // Adiciona campo de senha
        painelGrade.add(botaoLogin, 1, 2); // Adiciona botão de login

        // Define a cena e exibe o palco
        Scene cena = new Scene(painelGrade, 300, 200);
        palco.setTitle("Exemplo de Tela de Login");
        palco.setScene(cena);
        palco.show();
    }

    public static void main(String[] args) {
        launch(args); // Inicia a aplicação JavaFX
    }
}

```

codesnap.dev

Particularidades do JavaFX

- **CSS para Estilização:** JavaFX permite que o desenvolvedor estilize componentes com CSS, o que facilita a personalização e deixa a interface mais atraente.
- **FXML:** A linguagem FXML, baseada em XML, facilita a criação de interfaces, separando o layout da lógica de programação.
- **Integração com Swing:** JavaFX permite integrar componentes Swing, uma vantagem para quem trabalha com sistemas mais antigos.
- **Responsividade:** Suporte a redimensionamento dinâmico, permitindo que a interface se adapte a diferentes tamanhos de tela.

Conclusão

JavaFX é uma ferramenta essencial para o desenvolvimento de interfaces de usuário modernas e interativas no ambiente desktop, especialmente em aplicações que exigem um visual dinâmico e maior controle sobre a interface. Com uma API rica, suporte para animações e opções de personalização, JavaFX ainda é uma solução valiosa para o desenvolvimento de GUIs robustas.