Vamos crir mini projetos de JavaFx:

1 Hello World, Estilo JavaFX

A melhor maneira de ensinar como é criar e construir um aplicativo JavaFX é com um aplicativo "Hello World". Um benefício adicional deste tutorial é que ele permite que você teste que sua tecnologia JavaFX está instalada corretamente.

A ferramenta usada neste tutorial é o NetBeans IDE 7.3. Antes de começar, certifique-se de que a versão do NetBeans IDE que você está usando suporta JavaFX 2. Consulte os requisitos do sistema para obter detalhes.

Construa a Aplicação

1. No menu **Arquivo,** escolha **Novo Projeto**.
2. Na categoria de aplicação **JavaFX,** escolha o **Aplicativo JavaFX**. Clique **em Next**.
3. Nomeie o projeto **HelloWorld** e clique em **Concluir**.

   O NetBeans abre o arquivo e o preenche com o código para um aplicativo básico do Hello World, como mostrado no Exemplo 1-1.`HelloWorld.java`

   Exemplo 1-1 Hello World

```
package helloworld;

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.StackPane;
```

```
import javafx.stage.Stage;

public class HelloWorld extends Application {
    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Hello World!");
        Button btn = new Button();
        btn.setText("Say 'Hello World'");
        btn.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent event) {
                System.out.println("Hello World!");
            }
        });

        StackPane root = new StackPane();
        root.getChildren().add(btn);
        primaryStage.setScene(new Scene(root, 300, 250));
        primaryStage.show();
    }
}
```

Aqui estão as coisas importantes para saber sobre a estrutura básica de um aplicativo JavaFX:

- A classe principal para um aplicativo JavaFX estende a classe. O método é o principal ponto de entrada para todas as aplicações JavaFX.`javafx.application.Applicationstart()`
- Um aplicativo JavaFX define o contêiner de interface do usuário por meio de um estágio e uma cena. A classe JavaFX é o recipiente JavaFX de alto nível. A classe JavaFX é o contêiner para todo o conteúdo. O exemplo 1-1 cria o palco e a cena e torna a cena visível em um determinado tamanho de pixel.`StageScene`
- No JavaFX, o conteúdo da cena é representado como um gráfico hierárquico de nodes. Neste exemplo, o nó raiz é um objeto, que é um nó de layout resizável. Isso significa que o tamanho do nó raiz rastreia o tamanho da cena e muda quando o palco é redimensionado por um usuário.`StackPane`

- O nó raiz contém um nó infantil, um controle de botão com texto, além de um manipulador de eventos para imprimir uma mensagem quando o botão é pressionado.
- O método não é necessário para aplicativos JavaFX quando o arquivo JAR para o aplicativo é criado com a ferramenta Pacote JavaFX, que incorpora o JavaFX Launcher no arquivo JAR. No entanto, é útil incluir o método para que você possa executar arquivos JAR que foram criados sem o JavaFX Launcher, como ao usar um IDE no qual as ferramentas JavaFX não estão totalmente integradas. Além disso, os aplicativos Swing que incorporam o código JavaFX requerem o método.`main()main()main()`

Para onde ir a seguir

Isso conclui o tutorial básico do Hello World, mas continue lendo para mais lições sobre o desenvolvimento de aplicativos JavaFX:

- Criar um Formulário no JavaFX ensina o básico do layout da tela, como adicionar controles a um layout e como criar eventos de entrada.
- Fancy Forms com JavaFX CSS fornece truques de estilo simples para melhorar seu aplicativo, incluindo a adição de uma imagem de fundo e botões de estilo e texto.
- O uso do FXML para criar uma interface de usuário mostra um método alternativo para criar a interface do usuário de login. FXML é uma linguagem baseada em XML que fornece a estrutura para construir uma interface de usuário separada da lógica de aplicação do seu código.
- Animação e Efeitos Visuais no JavaFX mostra como dar vida a um aplicativo adicionando animação de linha do tempo e efeitos de mistura.

- A implantação do primeiro aplicativo JavaFX descreve como executar seu aplicativo fora do NetBeans IDE.

2 Criando um Formulário em JavaFX

Criar um formulário é uma atividade comum ao desenvolver uma aplicação. Este tutorial ensina o básico do layout da tela, como adicionar controles a um painel de layout e como criar eventos de entrada.

Neste tutorial, você usará o JavaFX para construir o formulário de login mostrado na Figura 2-1.

Formulário de login da figura 2-1

Descrição de "Figura 2-1 Formulário de Login"

A ferramenta usada neste tutorial Getting Started é o NetBeans IDE. Antes de começar, certifique-se de que a versão do NetBeans IDE que você está usando suporta JavaFX 2. Consulte os requisitos do sistema para obter detalhes.

Criar o Projeto

Sua primeira tarefa é criar um projeto JavaFX no NetBeans IDE e nomeá-lo Login:

1. No menu **Arquivo,** escolha **Novo Projeto**.
2. Na categoria de aplicação **JavaFX,** escolha o **Aplicativo JavaFX**. Clique **em Next**.
3. Nomeie o **login** do projeto e clique em **Concluir**.

   Quando você cria um projeto JavaFX, o NetBeans IDE fornece um aplicativo Hello World como ponto de partida, que você já viu se seguiu o tutorial hello world.

4. Remova o método que o NetBeans IDE gerou e substitua-o pelo código no Exemplo 2-1.`start()`

Exemplo 2-1 Estágio de aplicação

```
@Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("JavaFX Welcome");

        primaryStage.show();
    }
```

Dica: Depois de adicionar código de amostra em um projeto NetBeans, pressione Ctrl (ou Cmd) + Shift + I para importar as embalagens necessárias. Quando houver uma escolha de declarações de importação, escolha a que começa com .`javafx`

Crie um layout GridPane

Para o formulário de login, use um layout porque permite que você crie uma grade flexível de linhas e colunas para definir controles. Você pode colocar controles em qualquer célula da rede, e você pode fazer controles de células de extensão conforme necessário.`GridPane`

O código para criar o layout está no Exemplo 2-2. Adicione o código antes da linha `GridPaneprimaryStage.show();`

Exemplo 2-2 GridPane com propriedades de gap e preenchimento

```
GridPane grid = new GridPane();
grid.setAlignment(Pos.CENTER);
grid.setHgap(10);
grid.setVgap(10);
grid.setPadding(new Insets(25, 25, 25, 25));

Scene scene = new Scene(grid, 300, 275);
primaryStage.setScene(scene);
```

O exemplo 2-2 cria um objeto e o atribui à variável nomeada . A propriedade de alinhamento muda a posição padrão da grade do canto superior esquerdo da cena para o centro. As propriedades de lacuna gerenciam o espaçamento entre as linhas e colunas, enquanto a propriedade de preenchimento gerencia o espaço ao redor das bordas do painel de grade. Os insets estão na ordem de cima, direita, inferior e esquerda. Neste exemplo, há pixels de preenchimento em cada lado.`GridPanegrid25`

A cena é criada com o painel de grade como o nó raiz, que é uma prática comum ao trabalhar com recipientes de layout. Assim, à medida que a janela é redimensionada, os nós dentro do painel de grade são redimensionados de acordo com suas restrições de layout. Neste exemplo, o painel de grade permanece no centro quando você cresce ou encolhe a janela. As propriedades de preenchimento garantem que haja um preenchimento ao redor do painel de grade quando você fizer a janela menor.

Este código define a largura e altura da cena para 300 por 275. Se você não definir as dimensões da cena, a cena padrão para o tamanho mínimo necessário para exibir seu conteúdo.

Adicionar texto, rótulos e campos de texto

Olhando para a Figura 2-1,você pode ver que o formulário requer o título "Bem-vindo" e campos de texto e senha para coletar informações do usuário. O código para a criação desses controles está no exemplo 2-3. Adicione este código após a linha que define a propriedade de preenchimento da grade.

Exemplos 2-3 Controles

```
Text scenetitle = new Text("Welcome");
scenetitle.setFont(Font.font("Tahoma", FontWeight.NORMAL, 20));
grid.add(scenetitle, 0, 0, 2, 1);

Label userName = new Label("User Name:");
grid.add(userName, 0, 1);

TextField userTextField = new TextField();
grid.add(userTextField, 1, 1);

Label pw = new Label("Password:");
grid.add(pw, 0, 2);

PasswordField pwBox = new PasswordField();
grid.add(pwBox, 1, 2);
```

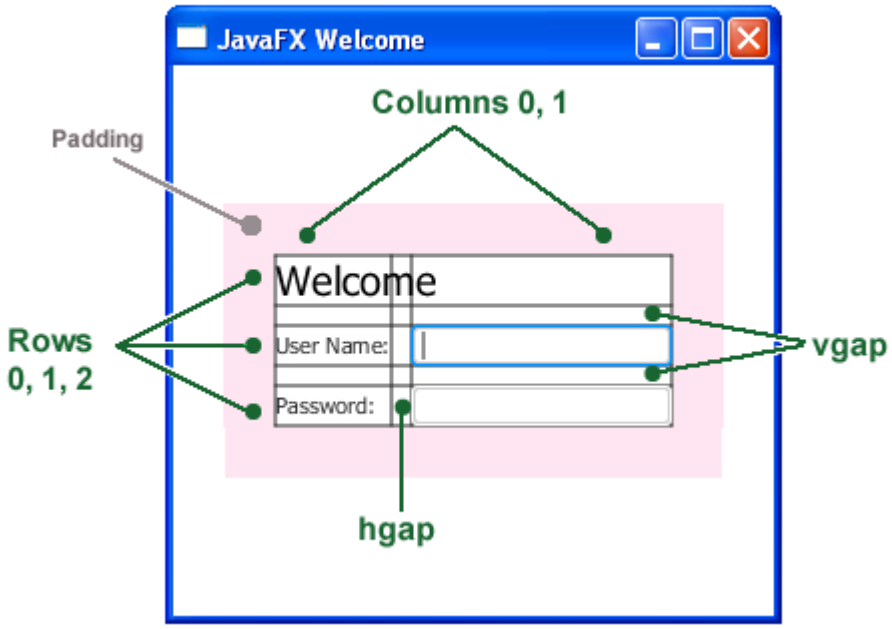
A primeira linha cria um objeto que não pode ser editado, define o texto e atribui-o a uma variável chamada . A próxima linha usa o método para definir a família da fonte, o peso e o tamanho da variável. Usar um estilo inline é apropriado onde o estilo está vinculado a uma variável, mas uma técnica melhor para estilizar os elementos da sua interface de usuário é usando uma folha de estilo. No próximo tutorial, Fancy Forms com JavaFX CSS, você substituirá o estilo inline por uma folha de estilo.`TextWelcomescenetitlesetFont()scenetitle`

O método adiciona a variável ao layout . A numeração para colunas e linhas na grade começa em zero, e é adicionada na coluna 0, linha 0. Os dois últimos argumentos do método definem o vão da coluna para 2 e o vão da linha para 1.`grid.add()scenetitlegridscenetitlegrid.add()`

As próximas linhas criam um objeto com texto na coluna 0, linha 1 e um objeto que pode ser editado. O campo de texto é adicionado ao painel de grade na coluna 1, linha 1. Um campo de senha e rótulo são criados e adicionados ao painel de grade de forma semelhante.`LabelUser NameText Field`

Ao trabalhar com um painel de grade, você pode exibir as linhas de grade, o que é útil para fins de depuração. Neste caso, você pode adicionar após a linha que adiciona o campo de senha. Em seguida, quando você executa o aplicativo, você vê as linhas para as colunas e linhas de grade, bem como as propriedades de lacuna, como mostrado na Figura 2-2.`grid.setGridLinesVisible(true)`

Formulário de login da Figura 2-2 com linhas de grade



Descrição de "Figura 2-2 Formulário de Login com Linhas de Grade"

Adicione um botão e texto

Os dois últimos controles necessários para o aplicativo são um controle para enviar os dados e um controle para exibir uma mensagem quando o usuário pressiona o botão.`ButtonText`

Primeiro, crie o botão e posicione-o no canto inferior direito, que é uma colocação comum para botões que executam uma ação que afeta toda a forma. O código está no exemplo 2-4. Adicione este código antes do código para a cena.

Exemplo 2-4 Botão

```
Button btn = new Button("Sign in");
HBox hbBtn = new HBox(10);
hbBtn.setAlignment(Pos.BOTTOM_RIGHT);
hbBtn.getChildren().add(btn);
grid.add(hbBtn, 1, 4);
```

A primeira linha cria um botão nomeado com a etiqueta e a segunda linha cria um painel de layout nomeado com espaçamento de pixels. O painel define um alinhamento para o botão diferente do alinhamento aplicado aos outros controles no painel de grade. A propriedade tem um valor de , que posiciona um nó na parte inferior do espaço verticalmente e na borda direita do espaço horizontalmente. O botão é adicionado quando criança do painel, e o painel é adicionado à grade na coluna 1, linha 4.`btnSign in,HBoxhbBtn10HBoxalignmentPos.BOTTOM_RIGHTHBoxHBox`

Agora, adicione um controle para exibir a mensagem, como mostrado no exemplo 2-5. Adicione este código antes do código para a cena.`Text`

Exemplo 2-5 Texto

```
final Text actiontarget = new Text();
```

```
        grid.add(actiontarget, 1, 6);
```

A figura 2-3 mostra a forma agora. Você não verá a mensagem de texto até trabalhar na próxima seção do tutorial, Adicionar código para lidar com um evento.

Formulário de login da figura 2-3 com botão

Descrição de "Figura 2-3 Formulário de Login com botão"

Adicionar código para lidar com um evento

Finalmente, faça com que o botão exiba a mensagem de texto quando o usuário a pressionar. Adicione o código no exemplo 2-6 antes do código para a cena.

Exemplo 2-6 Evento de botão

```
btn.setOnAction(new EventHandler<ActionEvent>() {

    @Override
    public void handle(ActionEvent e) {
        actiontarget.setFill(Color.FIREBRICK);
        actiontarget.setText("Sign in button pressed");
    }
});
```

O método é usado para registrar um manipulador de eventos que define o objeto quando o usuário pressiona o botão. A cor do objeto é definida como vermelho de tijolo de fogo.`setOnAction()actiontargetSign in button pressedactiontarget`

Executar o aplicativo

Clique com o botão direito do mouse no nó do projeto **Login** na janela Projetos, escolha **Executar**e clique no botão Assinar. A Figura 2-4 mostra os resultados. Se você encontrar problemas, então dê uma olhada no código no arquivo `Login.java`.

Dia 28 – Inerface grafica JAVAFX

Vídeo

https://www.youtube.com/watch?v=FLkOX4Eez6o

Exercício

1)O que é Scene Graph?

a) O Scene Graph é uma estrutura em árvore que representa a interface gráfica das aplicações Java.
b) O Scene Graph é uma estrutura em estrutura de dados que representa a interface gráfica das aplicações JavaFEX
c) O Scene Graph é uma estrutura em árvore que representa a interface gráfica das aplicações Python
d) Nem uma das alternativas

2) Como é a arquitetura JavaFX?

a) A arquitetura do JavaFX é composta por vários componentes que fornecem uma ampla variedade de funcionalidades, como variáveis , animações e recursos multimídia.
b) A arquitetura do JavaFX é composta por vários componentes que fornecem uma ampla variedade de funcionalidades, como gráficos, animações e recursos programatico.
c) A arquitetura do JavaFX é composta por vários componentes que fornecem uma ampla variedade de funcionalidades, como gráficos, animações e recursos multimídia.
d) Nem uma das alternativas

Dia 28 – Inerface grafica JAVAFX

3) Qual foi a primeira interface gráfica

a) JavaFx

b) Swing

c) AWT

d) Nem uma das

Prova

1) Como é a arquitetura JavaFX?

a) A arquitetura do JavaFX é composta por vários componentes que fornecem uma ampla variedade de funcionalidades, como variáveis , animações e recursos multimídia.
b) A arquitetura do JavaFX é composta por vários componentes que fornecem uma ampla variedade de funcionalidades, como gráficos, animações e recursos programatico.
c) A arquitetura do JavaFX é composta por vários componentes que fornecem uma ampla variedade de funcionalidades, como gráficos, animações e recursos multimídia.
d) Nem uma das alternativas

2) O gridpane precisa estar com qual propriedade

a) *GridPane com propriedades de gap e preenchimento*

b) *GridPane com propriedades de post e preenchimento*

c) *GridPane com propriedades de gap e get*

d) *GridPane com propriedades de nap e preenchimento*

3) Qual foi a primeira interface gráfica

a) JavaFx

b) Swing

c) AWT

d) Nem uma das