

Programação Orientada a Objetos

Java FX

Prof. Gabriel M. Alves

April 17, 2023

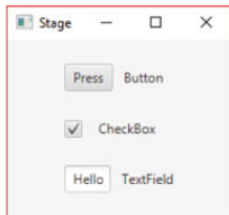
versão: 70d70cc

- *Graphical User Interface* (GUI) é uma interface que permite aos usuários interagir com dispositivos eletrônicos por meio de ícones e componentes visuais.
- Uma interface gráfica responde a eventos, que representam a interação do usuário com um determinado componente. Exemplos: mouse, teclado, *touchscreen*.
- Existem diversas bibliotecas para desenvolvimento de aplicações gráficas.

- Nas primeiras versões de Java utilizava-se a Abstract Window Toolkit (AWT), cujo propósito era fornecer um sistema gráfico em que os componentes criados eram associados aos componentes gráficos nativos do sistema operacional.
- Utilizando AWT a interface gráfica variava dependendo do sistema operacional em uso.
- Por utilizar muitos recursos do sistema operacional os componentes AWT era considerados “pesados” (*heavyweight*)
- A biblioteca *Java Swing* veio para substituir a AWT e a maioria das classes foram escritas em Java. Os componentes foram considerados “leves” (*lightweight*).
- Em aplicações utilizando *Java Swing*, a mesma interface gráfica não se alterava significativamente entre sistemas operacionais.

Introdução

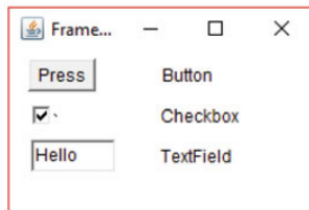
- Entre 2000 a 2014, *Java Swing* foi amplamente utilizada.
- Em 2014 foi anunciada a nova biblioteca **JavaFX**, embora *Java Swing* continua sendo disponibilizada junto com o Java.



(a) JavaFX



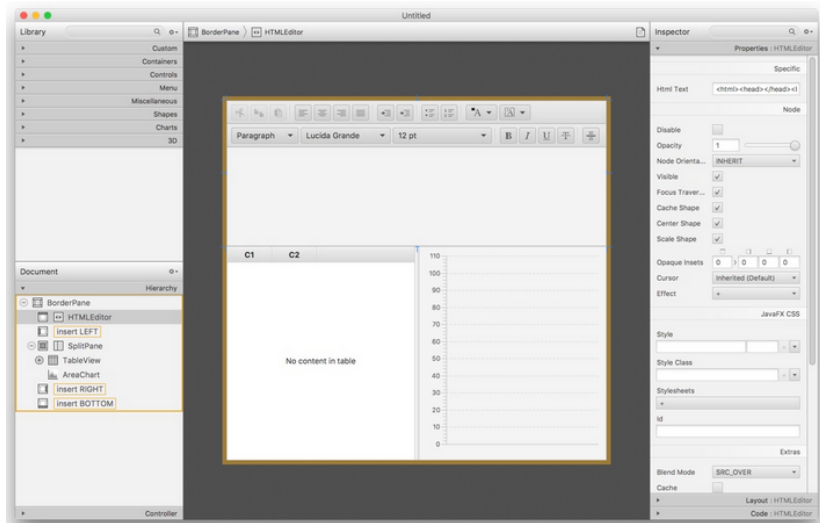
(b) Swing



(c) AWT (Windows)

- A biblioteca JavaFX consistem em um conjunto de pacotes que permitem o programador criar aplicações gráficas mais sofisticadas.
- Além de de interfaces gráficas (GUI), JavaFX tem potencial para jogos 2D e 3D, animações, efeitos especiais, aplicações que utilizam recursos de *touchscreen*.
- JavaFX tem vantagens sobre as bibliotecas predecessoras, como aceleração gráfica em hardware.
- JavaFX foi disponibilizado junto com o Java SDK até a versão 9. Agora a biblioteca é disponibilizada separadamente.

- Atualmente há duas maneiras de desenvolver uma aplicação utilizando JavaFX:
 - Utilizar o **JavaFX SDK**
 - Utilizar uma ferramenta de gerenciamento de software como **Apache Maven** ou **Gradle**
- Além disso, é possível utilizar a ferramenta **JavaFX Scene Builder** que permite criar interfaces gráficas, arrastando e soltando (*drag-and-drop*) componentes em uma área de *design* e gerando a maioria do código.



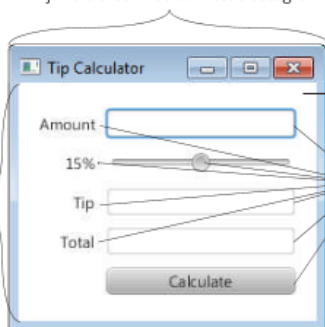
- Um programa JavaFX é chamado de **application**, pois a classe deve estender `Application`.

```
1  import javafx.application.Application;
2
3  public class MinhaTela extends Application{
4      // código utilizando JavaFX
5  }
```

- Outros termos associados a JavaFX:
 - stage** janela na qual estão todos os elementos da aplicação.
 - scene** uma cena em que estão os conteúdos da *stage*, em forma de grafo.
 - nodes** itens presentes em uma cena.
 - controls** componentes comuns que permitem interação com usuário (botões, campos, rótulos, etc).
 - containers** componentes que reúnem outros *nodes*. Permite organizar uma hierarquia de componentes.
- Normalmente colocamos um *node* principal e o chamamos de **root** (raiz). Utilizamos os termos **parent** e **children** para referenciar os objetos dentro de uma hierarquia.

JavaFX: visão geral

A janela é conhecida nesse estágio

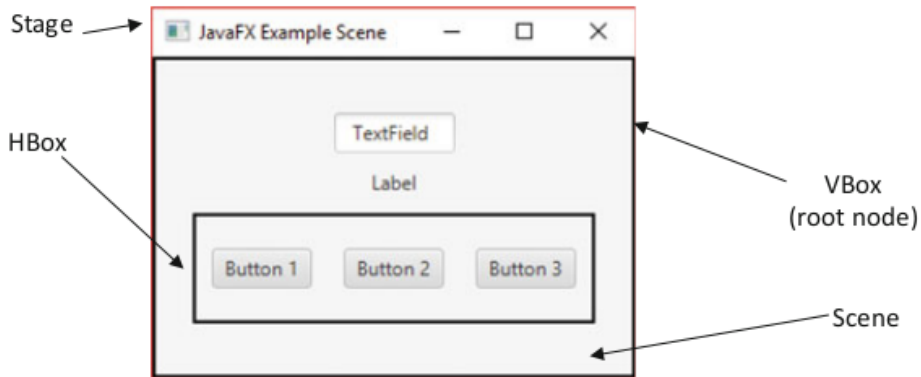


A cena desse estágio contém um grafo de cena de nós

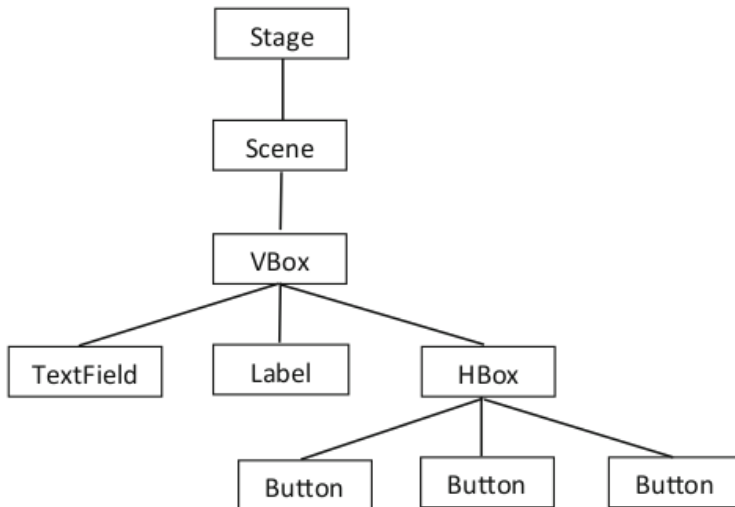
O nó raiz do gráfico de cena desta cena é um contêiner de layout que organiza os outros nós

Cada um dos componentes JavaFX nesta GUI é um nó no grafo de cena

JavaFX: visão geral



- Hierarquia dos componentes da interface gráfica apresentada.



- Quando uma aplicação JavaFX inicia, três métodos são chamados na seguinte ordem:
 - `void init()` : método em que são programadas rotinas necessárias e que devem ser executadas antes da aplicação iniciar.
 - `abstract void start(Stage stage)`: método no qual a aplicação é programada e iniciada. É importante observar que, se necessário, métodos auxiliares podem ser desenvolvidos e invocados dentro deste método.
 - `void stop()` : método em que são executados códigos após o encerramento da aplicação.
- A classe `Application` possui o método estático `launch()` a qual deve ser invocada a partir do `main()`:

```
1 Application.launch(args); // Opção 1
2 Application.launch(MinhaTela.class, args); // Opção 2
```

- Harvey M. Deitel, Paul J. Deitel. Java como programar (Capítulo 25), Pearson Brasil. 10ª edição. 2017.
- Quentin Charatan, Aaron Kans. Java in Two Semesters: Featuring JavaFX, Springer International Publishing. 2019.
- StackOverflow. Learning JavaFX. Disponível em: <http://riptutorial.com/ebook/javafx>.
- JavaFX: sítio oficial. Disponível em: <https://openjfx.io/>.

- Dúvidas?
- Comentários?

Contato

Gabriel Marcelino Alves
gabriel.marcelino@ifsp.edu.br



This work is licensed under Attribution-NonCommercial-ShareAlike 4.0 International. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/>

