Lập trình GUI với JavaFX

GV. Trần Trung Hiếu

Nội dung chính

1. **Giới thiệu lập trình GUI trong Java**
2. **Cài đặt JavaFX**
3. **Ứng dụng minh họa**
4. **Kiến trúc ứng dụng JavaFX**
5. **Các thành phần UI control cơ bản**
6. **Đối tượng khung chứa**
7. **Bộ quản lý trình bày**
8. **Xử lý sự kiện**
9. **Thiết lập style với CSS**
10. **FXML và Công cụ Scene Builder**

1. Giới thiệu về lập trình GUI trong Java

- **JavaAWT**
  - Thư viện Java AWT (Abstract Window Toolkit) bao gồm các thành phần có giao diện phụ thuộc vào nền tảng của hệ điều hành.
  - Cung cấp các thành phần UI như TextField, Label, TextArea, RadioButton, CheckBox, Choice, List...
- **Java Swing**
  - Không giống AWT, Java Swing cung cấp các thành phần gọn nhẹ và độc lập nền tảng.
  - Swing cung cấp nhiều thành phần mạnh mẽ hơn AWT như Tables, Lists, ScrollPanes, ColorChooser, TabbedPane...
  - Swing hỗ trợ plugin và tuân theo mô hình MVC (Model – View – Controller) trong khi AWT không có các đặc tính này.
- **JavaFX**
  - JavaFX là một giải pháp công nghệ cho GUI trên nền tảng Java nhằm tạo giao diện đồ họa người dùng dựa trên Swing và Java2D nhưng phong phú, mới mẻ và dễ sử dụng hơn rất nhiều.

1. Giới thiệu về lập trình GUI trong Java

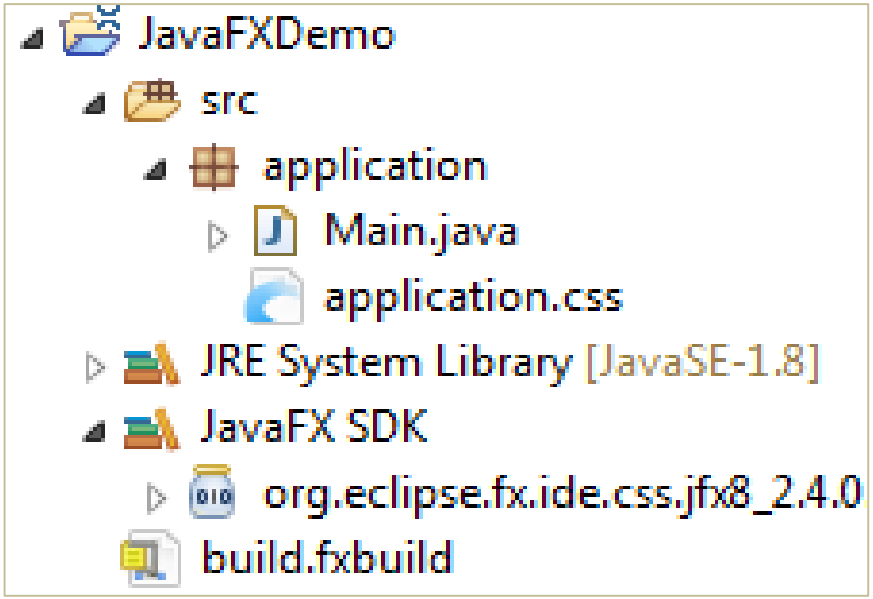
- **JavaFX**
  - JavaFX là một nền tảng phần mềm phát triển các ứng dụng Internet phong phú (Rich Internet Applications – RIAs) có thể chạy trên nhiều loại thiết bị, nhiều hệ điều hành khác nhau.
  - Người lập trình có thể lựa chọn xây dựng giao diện dựa trên FXML với sự hỗ trợ của công cụ trực quan Scene Builder
  - JavaFX cũng cho phép tích hợp CSS vào để làm ứng dụng nổi bật, phong phú và hoàn hảo hơn
  - JavaFX & HTML5: JavaFX giàu tính năng hơn HTML5, HTML 5 phù hợp với các ứng dụng Web công cộng, vì yêu cầu cài plugin cho trình duyệt khách hàng là không nên. JavaFX là lựa chọn tốt hơn cho các ứng dụng doanh nghiệp khi các plugin cần thiết và các tính năng khác đều được cài đặt sẵn trên các máy nội bộ. (JavaFX cần plugin để chạy trên trình duyệt)

2. Cài đặt JavaFX

- **Cài đặt trên Eclipse IDE**
  - Download: https://www.eclipse.org/downloads/packages/
  - Người sử dụng nên cài đặt bản Eclipse IDE for Enterprise Java Developers để có khả năng mở rộng cho lập trình web về sau
- **Cài đặt plugin e(fx)clipse để tạo ứng dụng JavaFX**
  - Từ cửa sổ Eclipse -> Help -> Eclipse Marketplace -> Tìm kiếm e(fx)clipse và cài đặt
- **Cài đặt công cụ thiết kế giao diện Scene Builder và tích hợp vào IDE**
  - http://www.oracle.com/technetwork/java/javase/downloads/javafxscenebuilder-1x-archive-2199384.html
  - Cấu hình Eclipse trỏ tới Scene Builder theo thứ tự các bước: Từ cửa sổ Eclipse chọn Window -> Preferences -> JavaFX -> Browser -> Chọn đường dẫn tới file chương trình chạy (exe) của Scene Builder

3. Ứng dụng minh họa

- **Tạo project ứng dụng JavaFX trên Eclipse**
  - Trên cửa sổ Eclipse chọn File -> New -> Project -> JavaFX -> JavaFX Project -> Project name (Nhập tên project) -> Finish
  - Cấu trúc ứng dụng JavaFX mặc định được tạo ra:



3. Ứng dụng minh họa

- **Cấu trúc lớp Main.java**

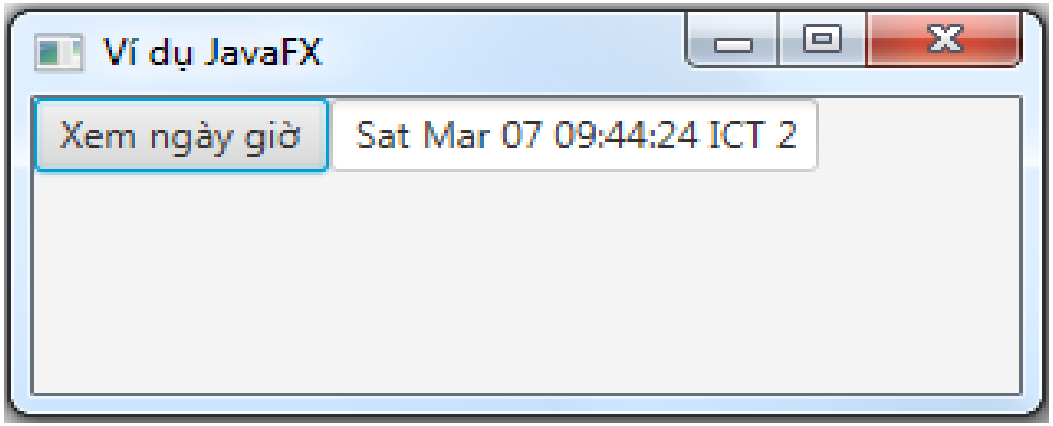
```
package application;
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.TextField;
import javafx.scene.layout.HBox;
import javafx.stage.Stage;

public class Main extends Application {
    @Override
    public void start(Stage primaryStage) {
        try {
            // Tạo thành phần UI nút bấm và trường text
            Button btn = new Button("Xem ngày giờ");
            TextField tf = new TextField();

            // Xử lý sự kiện bấm nút
            btn.setOnAction(new EventHandler<ActionEvent>() {
                @Override
                public void handle(ActionEvent arg0) {
                    // Thêm ngày giờ hiện tại vào ô text
                    tf.setText(new java.util.Date().toString());
                }
            });
```

3. Ứng dụng minh họa

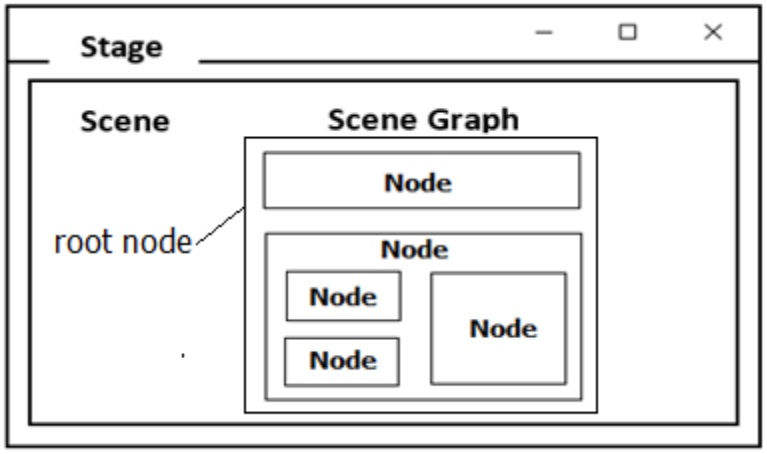
```
                        // Thêm nút bấm, trường text vào layout Hbox
                        HBox root = new HBox();
                        root.getChildren().addAll(btn, tf);

                        // Thêm layout vào khung chứa Scene
                        Scene scene = new Scene(root,300,100);

                        // Thêm Scene vào khung chứa Stage
                        primaryStage.setScene(scene);

                        // Đặt tiêu đề cho khung chứa Stage và hiển thị
                        primaryStage.setTitle("Ví dụ JavaFX");
                        primaryStage.show();
                } catch(Exception e) {
                        e.printStackTrace();
                }
        }
        public static void main(String[] args) {
                launch(args);
        }
}
```

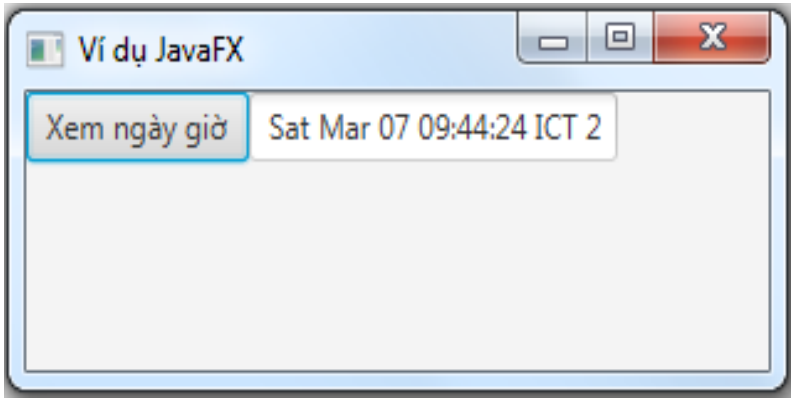
3. Ứng dụng minh họa

- **Chạy chương trình**
  - Để chạy chương trình, ta chuột phải vào lớp Main.java chọn Run As -> Java Application.
  - Cửa sổ chương trình hiển thị, người dùng bấm nút "Xem ngày giờ" để xem ngày giờ được hiển thị ở ô text bên cạnh.



4. Kiến trúc ứng dụng JavaFX






```
public class Main extends Application {
        @Override
        public void start(Stage primaryStage) {
                try {
                        // Tạo thành phần UI control

                        // Xử lý sự kiện với UI control nếu có

                        // Thêm UI control vào vùng chứa layout

                        // Thêm layout vào khung chứa Scene

                        // Thêm Scene vào khung chứa Stage

                        // Đặt tiêu đề cho khung chứa Stage và
                        hiển thị
                } catch(Exception e) {
                }
        }
        public static void main(String[] args) {
                launch(args);
        }
}
```

4. Kiến trúc ứng dụng JavaFX

- Trong JavaFX các ứng dụng được đại diện, mô hình hóa bởi lớp Application thuộc gói thư viện javafx.application.
- Các cửa sổ được đại diện bởi lớp javafx.stage.Window, lớp này có hai lớp thừa kế là Stage và PopupWindow.
- Mỗi thể hiện của lớp Stage ứng với lớp vùng chứa ngoài cùng của cửa sổ, lớp vùng chứa thứ hai bên trong nó gọi là Scene.
- Vùng chứa Scene chứa đựng các thành phần giao diện người sử dụng (UI) như nút bấm, trường text, đường link, nhãn.v.v. Mỗi thành phần UI gọi là một Node (nút).
- Lớp PopupWindow là lớp cha đại diện cho các lớp tạo cửa sổ popup bao gồm Popup, Tooltip, ContextMenu.

4. Kiến trúc ứng dụng JavaFX

- **Lớp Application**
  - Để tạo ứng dụng JavaFX, ta cần có lớp chương trình chính thừa kế từ lớp Application. Để chạy ứng dụng JavaFX, ta sử dụng phương thức ***launch*** của lớp Application theo một trong hai mẫu:

    public static void launch(java.lang.String... args)

    public static void launch(java.lang.Class<? extends Application>

    appClass, java.lang.String... args)
  - Các tham số truyền vào phương thức launch có thể nhận lại được từ bên trong lớp con của lớp Application bằng cách gọi phương thức getParameters.

4. Kiến trúc ứng dụng JavaFX

- **Lớp Application**
  - Khi một ứng dụng JavaFX được thực thi, nội dung thực hiện của nó bao gồm 5 bước:
    - *B1: Tạo một thể hiện của lớp Application*
    - *B2: Gọi phương thức khởi tạo init. Phương thức này nên được ghi đè nếu cần khởi tạo dữ liệu cho ứng dụng.*
    - *B3: Gọi phương thức start để tạo thể hiện cho Stage, Scene và các thành phần UI*
    - *B4: Ứng dụng kết thúc khi người lập trình chủ động gọi phương thức Platform.exit() hoặc khi tất cả các cửa sổ đã được đóng*
    - *B5: Gọi phương thức stop để giải phóng tài nguyên. Phương thức này nên ghi đè nếu cần chủ động giải phóng tài nguyên*

4. Kiến trúc ứng dụng JavaFX

- **Lớp Stage**
  - Stage là lớp vùng chứa ngoài cùng của mọi thành phần UI.
  - Khi ứng dụng chạy, một thể hiện của lớp Stage được tạo ra, nó được dùng làm tham số truyền vào trong lời gọi phương thức start, đây là cửa sổ chính của ứng dụng. Ta cũng có thể tạo ra các Stage khác nếu cần thiết.
  - Stage bao gồm một số thuộc tính thừa kế từ lớp Window như chiều rộng (width), chiều cao (height), tọa độ (x,y), trạng thái hiển thị (showing), thành phần chứa bên trong Scene.v.v. các thuộc tính riêng tiêu đề (title), khả năng tùy chỉnh kích cỡ (resizable), xác lập vị trí (alwaysOnTop).v.v. cùng các phương thức truy xuất vào các thuộc tính đó. Thông thường, các phương thức setTitle, setScene, show cần được gọi để đặt tiêu đề, thiết lập lớp vùng chứa Scene, và đổi trạng thái hiển thị.

4. Kiến trúc ứng dụng JavaFX

- **Lớp Scene**
  - Scene là lớp vùng chứa thứ hai được đặt trong lớp vùng chứa Stage. Có nhiều constructor để có thể tạo ra một đối tượng Scene:

    public Scene(Parent root)

    public Scene(Parent root, double width, double height)

    public Scene(Parent root, javafx.scene.paint.Paint fill)

    public Scene(Parent root, double width, double height, javafx.scene.paint.Paint fill)
  - Một đối tượng Scene phải chứa trong nó một Node cha (Parent). Một Node cha được đại diện bởi lớp javafx.scene.Parent.

4. Kiến trúc ứng dụng JavaFX

- **Các thành phần UI**
  - Mỗi thành phần UI trong JavaFX gọi là một Node.
  - Lớp javafx.scene.Node là lớp cha, lớp nền tảng cho tất cả các Node khác, nó bao gồm 5 lớp con ứng với 5 loại thành phần UI khác nhau:
    - *Canvas: Là một vùng hình chữ nhật cho phép người dùng vẽ lên đó*
    - *Parent: Là một lớp cha nền tảng cho một số thành phần UI, bất kỳ thành phần UI nào cần được đặt trong một thành phần UI thừa kế từ lớp Parent, trước khi tạo lập đối tượng Scene. Trong ví dụ minh họa, Hbox là một lớp con của Parent đóng vai trò là bộ quản lý trình bày (layout), các thành phần UI nút bấm, trường text được thêm vào Hbox, được sắp xếp theo hàng ngang trước khi đối tượng Hbox được truyền vào làm tham số tạo đối tượng Scene.*
    - *Shape: Là lớp cha cho các lớp đại diện cho các loại hình học như hình chữ nhật, hình tròn, elip, đường thẳng...*
    - *ImageView: Là một vùng chứa riêng cho hiển thị hình ảnh*
    - *MediaView: Cung cấp chế độ xem media bởi trình MediaPlayer*

4. Kiến trúc ứng dụng JavaFX

- **Các thành phần UI**
  - Các lớp Canvas, ImageView, and MediaView không có lớp thừa kế. Lớp Shape có các lớp thừa kế như Arc, Circle, Ellipse, Line, Path, Rectangle. Lớp Parent có 3 lớp thừa kế:
    - *Region: Là một lớp cha nền tảng cho các lớp giao diện tương tác (Control), các lớp bộ quản lý trình bày (Pane), các lớp đồ thị (Chart) và các lớp trục đồ thị (Axis)*
    - *Group: Là một vùng chứa để nhóm các thành phần UI vào một nhóm từ đó có thể thiết lập các ảnh hưởng chung cho nhóm. Group thường được sử dụng để nhóm các đối tượng thuộc lớp Shape*
    - *WebView: Là một trình duyệt web mini hay một trình duyệt được nhúng trong cửa sổ ứng dụng JavaFX*

5. Một số thành phần UI Control cơ bản

- Các thành phần giao diện tương tác (UI Control) thừa kế từ lớp javafx.scene.control.Control.
- Các mô tả về một thành phần giao diện bao gồm các lớp cha, các thuộc tính, phương thức của chính lớp đó hay thừa kế từ các lớp cha người lập trình cần tra cứu trong JavaFX API.
- Thông thường để hiểu và sử dụng một lớp, ta trước hết cần biết các thuộc tính của chính lớp đó, các thuộc tính kế thừa từ các lớp cha, căn cứ vào các thuộc tính đó ta sẽ dễ dàng tra ra các phương thức tác động vào thông qua các tham chiếu phương thức đính kèm của thuộc tính.
- Chi tiết về một số UI control xem tại: một số UI Control

6. Đối tượng khung chứa

- Về mặt khái niệm khung chứa là các thành phần mà có thể chứa các thành phần khác.
- Về cách tổ chức giao diện:
  - Java AWT và Swing tách biệt khung chứa và bộ quản lý trình bày (layout). Một bộ quản lý trình bày trong Java AWT và Swing chỉ có ý nghĩa logic, các thành phần con được thêm trực tiếp vào một số bộ khung chứa như frame, dialog, panel.
  - JavaFX thì khác, nó tổ chức giao diện thành các nút (Node), mỗi nút ứng với một thành phần giao diện, nút này chứa trong nó nút kia, cứ như vậy cho đến thành phần con nhỏ nhất. Các bộ quản lý trình bày trong JavaFX cũng là các nút, vừa có tác dụng khung chứa, vừa có tác dụng quản lý trình bày các thành phần giao diện con.
- Giới thiệu một số đối tượng khung chứa có dạng một cửa sổ, hộp thoại: Xem chi tiết

7. Bộ quản lý trình bày

- Các bộ quản lý trình bày trong JavaFX là các vùng chứa được sử dụng để sắp xếp linh hoạt và tương thích động các thành phần giao diện đồ họa tương tác trong vùng chứa Scene của JavaFX. Khi một cửa sổ được thay đổi kích thước, khung bố trí sẽ tự động định vị lại và thay đổi kích thước các thành phần chứa bên trong nó.

  javafx.scene.Node

  javafx.scene.Parent

  javafx.scene.layout.Region

  javafx.scene.layout.Pane

- JavaFX cung cấp một số lớp có chức năng của một bộ quản lý trình bày như Hbox, Vbox, FlowPane, StackPane, BorderPane, GridPane, TilePane, TextFlow, đây là các lớp thừa kế từ lớp javafx.scene.layout.Pane, ngoài ra có thể kể thêm một số lớp khác như ScrollPane, Accordion...
- Trong bài học này, chúng tôi giới thiệu một số lớp thông dụng thừa kế từ lớp Pane. Các lớp này thừa kế các thuộc tính, phương thức chung từ các lớp cha Pane, Region, Parent, Node như border, padding, margin, spacing, alignment...
- Xem chi tiết

8. Xử lý sự kiện

- Mỗi khi người sử dụng tương tác với các thành phần giao diện một sự kiện phát sinh.
- Có nhiều loại sự kiện như click vào một nút nhấn, di chuột, nhập ký tự từ bàn phím, chọn một mục từ danh sách hay cuộn trang...
- Mỗi thành phần giao diện khác nhau có thể phát sinh những sự kiện khác nhau.
- Tất cả các lớp sự kiện trong JavaFX đều thừa kế từ lớp nền tảng javafx.event.Event.
- Xem chi tiết

9. Thiết lập style với CSS

- JavaFX cho phép ta sử dụng các đoạn mã CSS để chèn vào thành phần giao diện, tương tự như sử dụng CSS để thiết kế các trang web. Các đoạn mã CSS này giúp cho các thành phần giao diện trong JavaFX trở nên đẹp mắt và phong cách hơn.
- Trong JavaFX, gói javafx.css chứa các lớp được sử dụng để áp dụng CSS cho các ứng dụng. Style mặc định của các giao diện ứng dụng được đặt trong file modena.css, file này nằm trong gói thư viện jar của JavaFX.
- Cú pháp để tích hợp CSS vào JavaFX gồm 3 phần:

  selector {property: value}

  – Trong đó:

  + Selector: loại thẻ được áp dụng. Ví dụ: thẻ <h1>, thẻ <table>

  + Property: thuộc tính trong CSS. Ví dụ: border, color

  + Value: giá trị được gán cho thuộc tính. Ví dụ: giá trị gán cho màu như red, #F1F1F1

9. Thiết lập style với CSS

- Tài liệu hướng dẫn tham chiếu CSS (JavaFX CSS Reference Guide) bạn đọc có thể xem tại địa chỉ:

  https://docs.oracle.com/javase/8/javafx/api/javafx/scene/doc-files/cssref.html
- CSS có thể được tích hợp vào JavaFX theo 2 cách:
  - Chèn trực tiếp mã CSS vào thành phần UI. Ví dụ:

```
button1.setStyle("-fx-background-color:linear-gradient(greenyellow, limegreen );\n" +
                 "    -fx-text-fill: black;\n" +
                 "    -fx-font-size: 20px;\n" +
                 "    -fx-padding: 5 30 5 30;\n" +
                 "    -fx-background-radius: 30, 30, 29, 28;");
```

  - Khai báo mã CSS ở một file riêng có đuôi css.

9. Thiết lập style với CSS

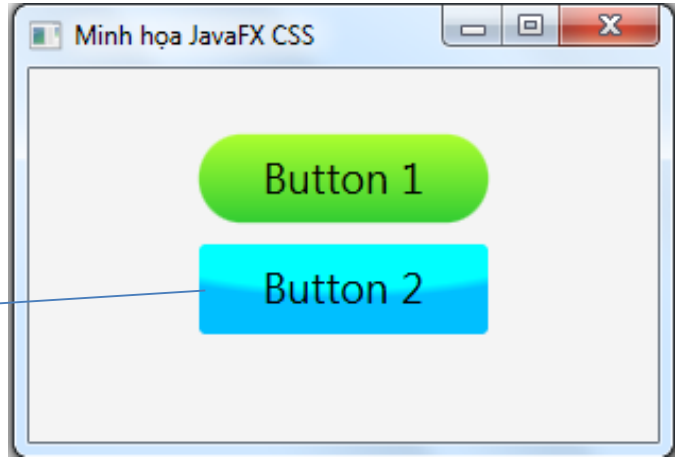
– Khai báo mã CSS ở một file riêng có đuôi css

- *Ví dụ:*

scene.getStylesheets().add(getClass().getResource("style.css").toExternalForm());

```
/* CSS chung cho các Button1, 2 có trong giao diện */
.button{
   -fx-background-color: radial-gradient(center 50% -40%,
radius 200%, #00FFFF 45%, #00BFFF 50%);
    -fx-text-fill: black;
    -fx-font-size: 20px;
    -fx-padding: 5 30 5 30;
}

.button:hover {
    -fx-background-color : skyblue ;
}
```

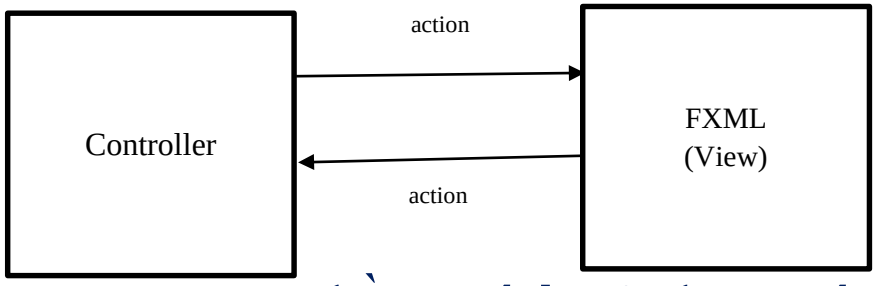
style.css



Chương trình chèn style trực tiếp cho Button1 và sử dụng style.css cấu hình chung. Style chèn trực tiếp được ưu tiên

10. FXML & Công cụ Scene Builder

- FXML là ngôn ngữ đánh dấu dựa trên XML để xây dựng giao diện người dùng (UI) trong ứng dụng JavaFX. Sử dụng FXML trong các ứng dụng JavaFX là một cách tuyệt vời để tách biệt tầng hiển thị và tầng logic.
- Theo mô hình này, tầng hiển thị hay giao diện người dùng được xây dựng hoàn toàn bằng các thẻ XML lưu trong một file có đuôi fxml



- Có thể tự xây dựng trang FXML bằng cách sử dụng các thẻ XML hoặc sử dụng công cụ hỗ trợ tạo giao diện Scene Builder
- Quy tắc sử dụng các thẻ XML trong FXML có thể tra cứu tại địa chỉ: *https://docs.oracle.com/javase/8/javafx/api/javafx/fxml/doc-files/introduction_to_fxml.html*

10. FXML & Công cụ Scene Builder

- **Các bước xây dựng ứng dụng JavaFX với FXML**
  - Bước 1: Tạo trang FXML với công cụ Scene Builder
    - *Từ một package trong thư mục project của Eclipse IDE, ta bấm chuột phải -> New -> Other -> JavaFX -> New FXML Document -> Đặt tên cho file FXML, ví dụ MyScene.fxml*
    - *Chuột phải vào file MyScene.fxml chọn "Open with SceneBuilder" để sử dụng công cụ Scene Builder tạo giao diện*
    - *Kết nối file FXML với lớp điều khiển: Sau khi tạo xong giao diện, ta thực hiện lưu file trên Scene Builder, quay trở lại cửa sổ Eclipse, ta mở file fxml, thêm thuộc tính fx:controller cho thẻ tương ứng với layout. Ví dụ:*

      <AnchorPane ... fx:controller="application.MyController">

      trong đó MyController là tên lớp điều khiển, application là tên gói chứa lớp.

10. FXML & Công cụ Scene Builder

- **Các bước xây dựng ứng dụng JavaFX với FXML**
  - Bước 2: Viết lớp điều khiển logic
    - *Lớp điều khiển logic cần quản lý các thành phần giao diện tương tác trên trang fxml tương ứng, các thuộc tính này cần có tên trùng với khai báo fx:id trên trang fxml và có thêm annotaion @FXML đi kèm mỗi khai báo thuộc tính.*
    - *Lớp điều khiển cũng cần có các phương thức trùng tên phương thức xử lý sự kiện đã khai báo trên trang fxml tương ứng.*

10. FXML & Công cụ Scene Builder

Ví dụ:

```
package application;
import javafx.fxml.FXML;
import javafx.scene.control.Button;
import javafx.scene.control.TextField;

public class MyController{
	@FXML
	private Button myButton;
	@FXML
	private TextField myTextField;

	public void showDateTime() {
		myTextField.setText(new
			java.util.Date().toString());
	}
}
```

10. FXML & Công cụ Scene Builder

- Bước 3: Kết hợp trang FXML và lớp điều khiển trong chương trình chính
  - *Tại lớp chương trình chính của ứng dụng JavaFX ta cần tải trang fxml và chuyển đổi nó sang đối tượng có kiểu javafx.scene.Parent. Đối tượng này sẽ được dùng để tạo Scene.*

- **Ví dụ:** (xem slide kế tiếp)

```
package application;
import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;
```

10. FXML & Công cụ Scene Builder

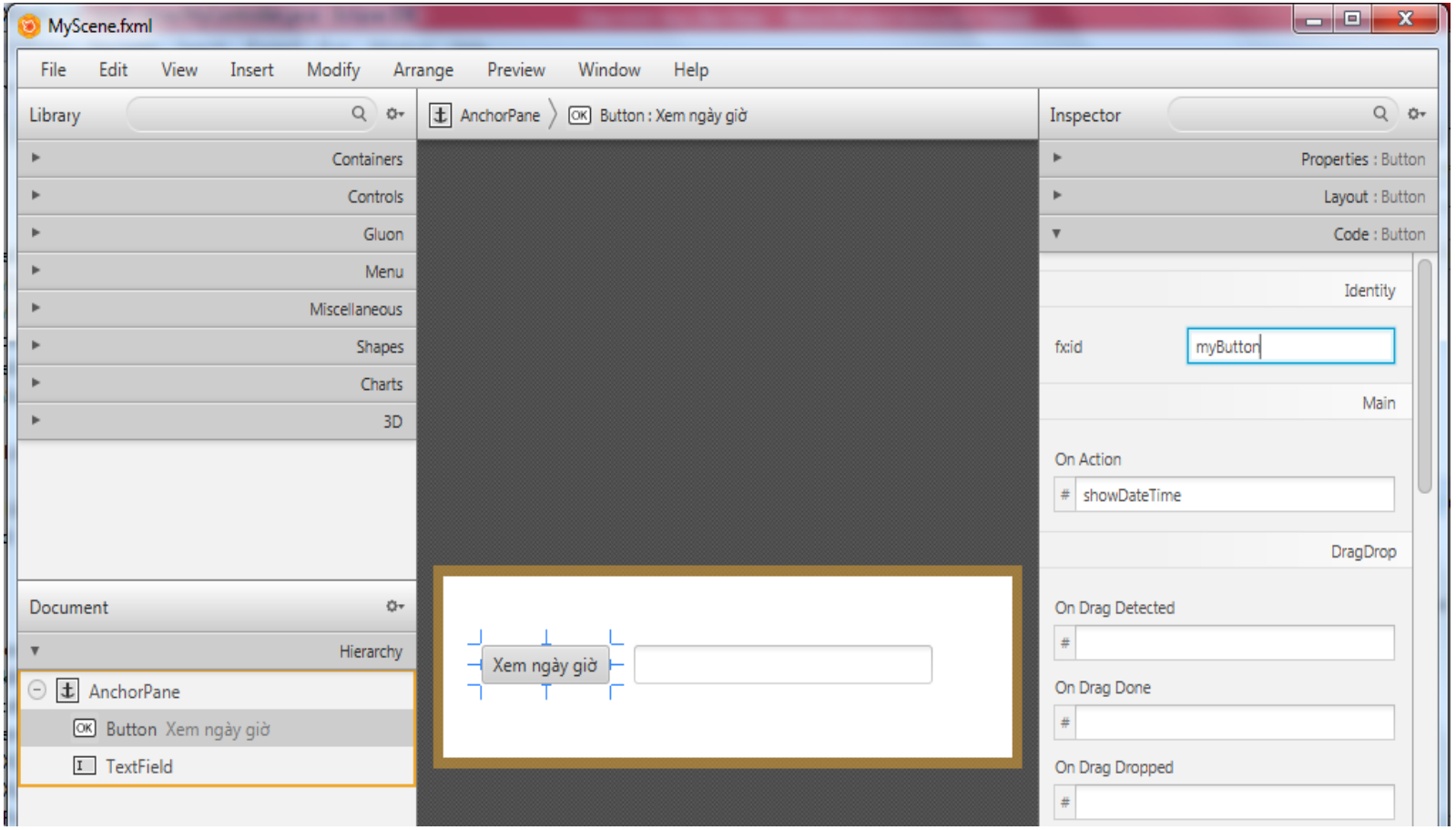
```
public class Main extends Application {
	@Override
	public void start(Stage primaryStage) {
		try {
			// Đọc file fxml và vẽ giao diện.
			Parent root =
		FXMLLoader.load(getClass().getResource("MyScene.fxml"));

			Scene scene = new Scene(root, 370, 100);
			primaryStage.setScene(scene);
			primaryStage.setTitle("Minh họa JavaFX với FXML");
			primaryStage.show();
		} catch (Exception e) {
			e.printStackTrace();
		}
	}

	public static void main(String[] args) {
		launch(args);
	}
}
```

10. FXML & Công cụ Scene Builder

- **Quy tắc tạo giao diện trên Scene Builder**
  - Tạo một layout đóng vài trò là vùng chứa bên ngoài của các thành phần giao diện tương tác, có thể có nhiều layout lồng nhau để phân vùng quản lý các thành phần chứa bên trong. Các layout này có thể lựa chọn tại nhóm Container trên Scene Builder.
    - *Ví dụ: Chọn layout AnchorPane*
  - Thêm các thành phần giao diện tương tác vào các vùng chứa layout
    - *Ví dụ: Thêm một Button, một TextField vào vùng AnchorPane*
  - Thiết lập thuộc tính, phương thức xử lý sự kiện cho mỗi thành phần giao diện tương tác. Với các thành phần giao diện cần quản lý trong lớp điều khiển, ta phải đặt id cho nó.
    - *Ví dụ: Tại mục "Text", đặt nhãn cho Button là "Xem ngày giờ", fx:id là myButton, sự kiện On Action là "showDateTime". Tương tự với thành phần TextField ta đặt id là myTextField.*

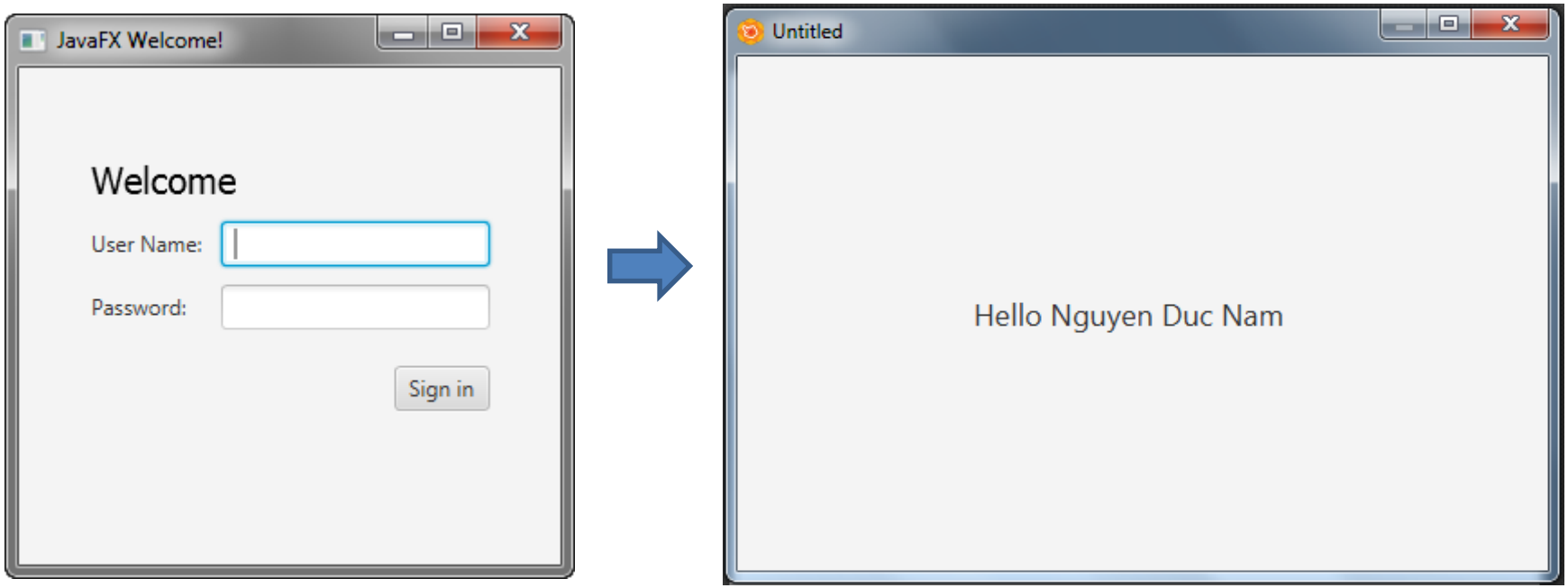
10. FXML & Công cụ Scene Builder

- **Quy tắc tạo giao diện trên Scene Builder**



10. FXML & Công cụ Scene Builder

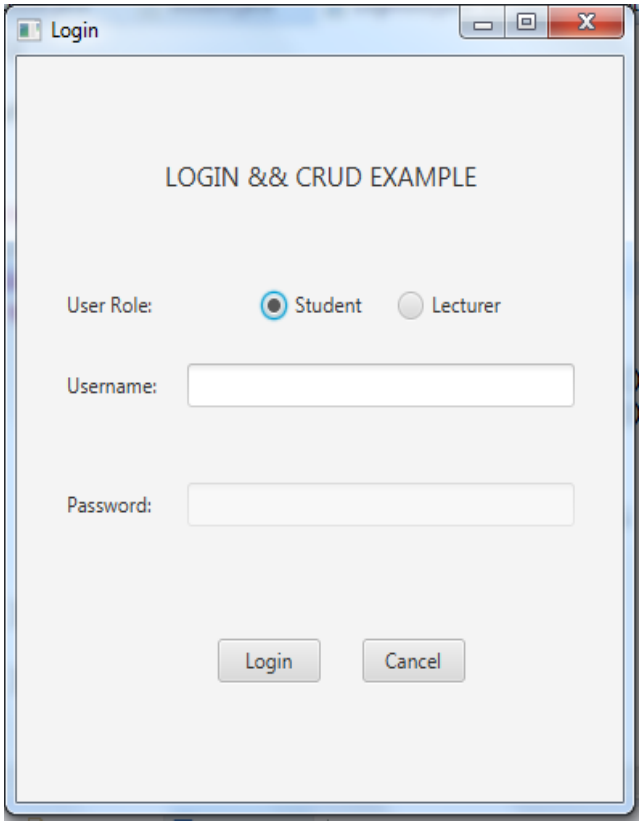
- **Bài tập thực hành 1**

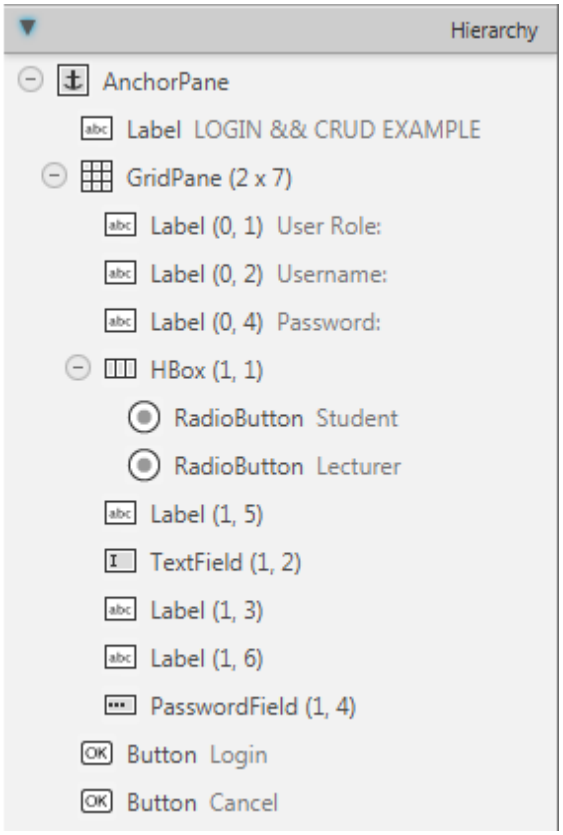


Sau khi đăng nhập thành công, chuyển tới màn hình chính với thông điệp: Xin chào + tên đầy đủ của người đăng nhập

10. FXML & Công cụ Scene Builder

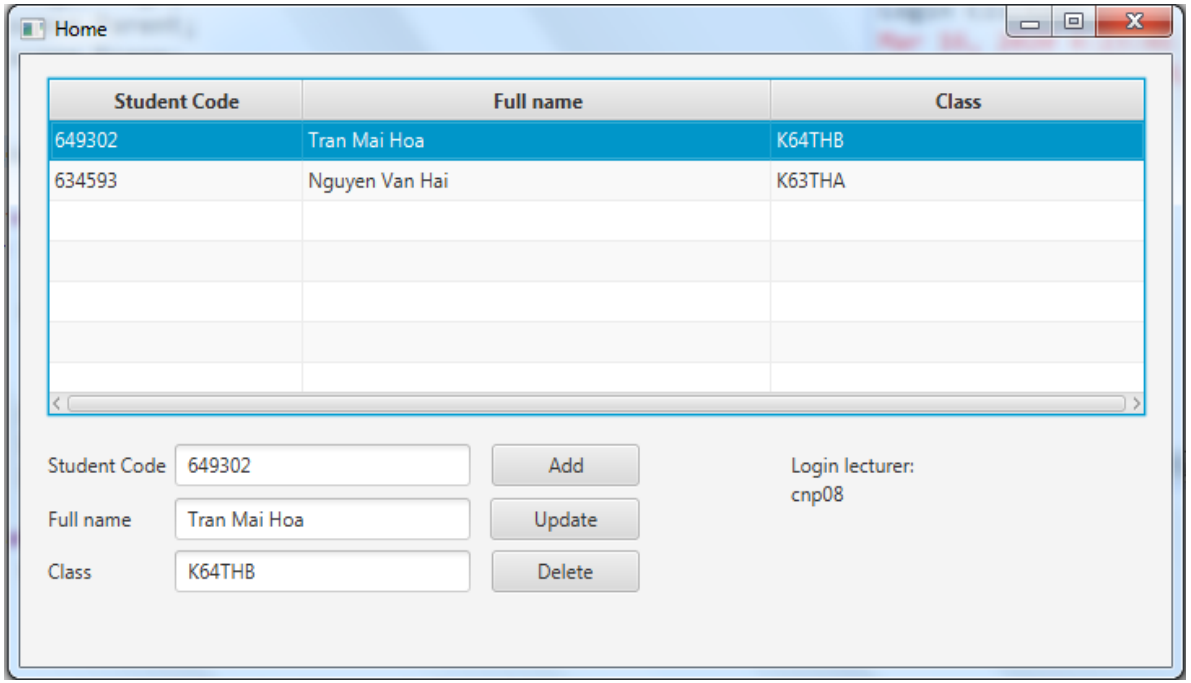
- **Bài tập thực hành 2**

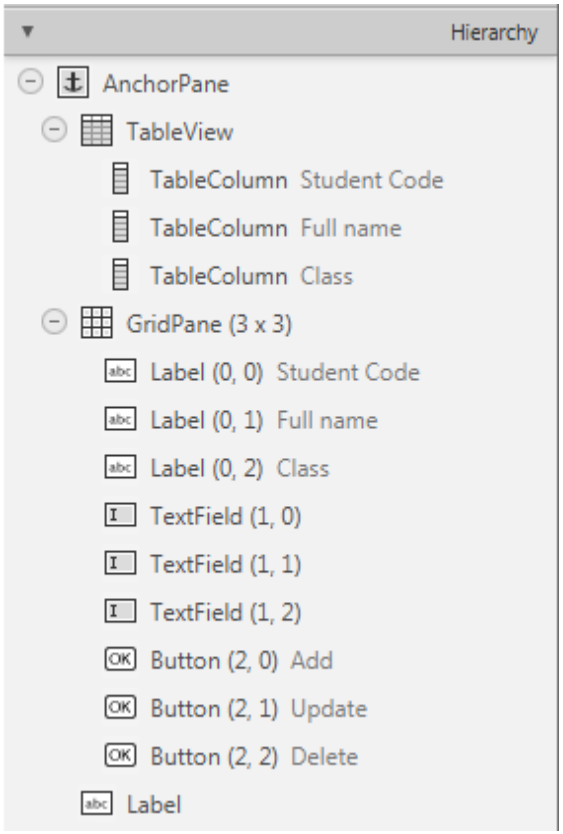




10. FXML & Công cụ Scene Builder

- **Bài tập thực hành 2**





10. FXML & Công cụ Scene Builder

- **Bài tập thực hành 2**
  - Viết chương trình xử lý
    - *Sinh viên:*
      - Chỉ cần đăng nhập bởi mã sinh viên
      - Sau khi đăng nhập, không được quyền sửa, xóa thông tin
    - *Giảng viên:*
      - Đăng nhập bằng mã giảng viên và mật khẩu
      - Sau khi đăng nhập được quyền thêm, sửa, xóa sinh viên
  - Gợi ý:
    - *Tạo các lớp:*
      - Main, LoginScene.fxml, LoginController, HomeScene.fxml, HomeController
      - Student, Lecturer, Manager
    - *Cần khởi tạo một danh sách sinh viên, danh sách giảng viên trong lớp quản lý Manager*

Tài liệu tham khảo

- Budi Kurniawan (2015). Java A Beginner's Tutorial, 4th edition. Brainy Software.
- Kishori Sharan (2015). Learn JavaFX 8: Building User Experience and Interfaces with Java 8 , 1th edition. Apress.
- O7planning.

  URL: https://o7planning.org/vi/11009/javafx