

LẬP TRÌNH JAVA

JAVAFX

ThS. Dương Hữu Thành
Khoa CNTT, Đại học Mở Tp.HCM
thanh.dh@ou.edu.vn



1



Nội dung chính

- 1. Giới thiệu JavaFX**
- 2. Lớp Color, Font, Image, ImageView**
- 3. Các loại pane**
- 4. Shape**
- 5. Xử lý sự kiện**
- 6. Hiệu ứng**
- 7. Vẽ biểu đồ**



Giới thiệu JavaFX

- Năm 2010 Java công bố các hướng tương lai phát triển Java bao gồm hoàn thành JavaFX là kỹ thuật trung tâm để xây dựng ứng dụng.
- Tháng 3/2014, Java phát hành **Java 8** là bản phát hành Java chính thức chứa JavaFX.
- JavaFX có thể dùng thay thế Swing + AWT trong phát triển các ứng dụng giao diện người dùng với Java.

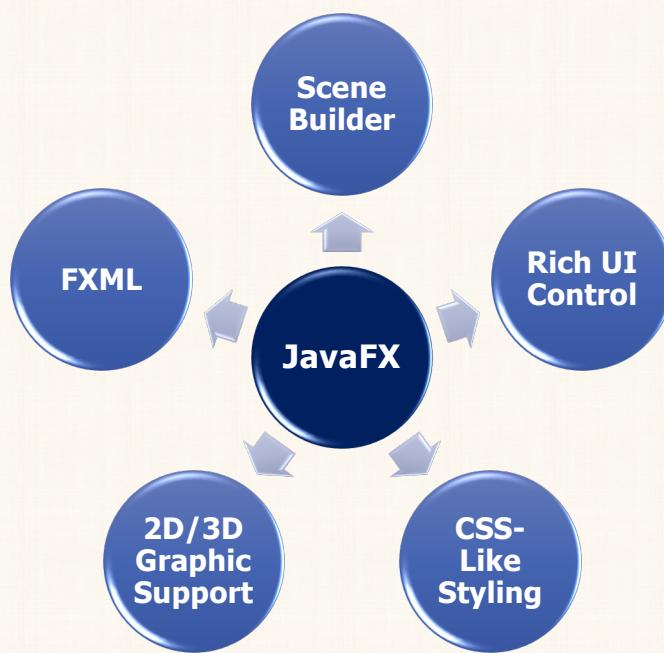
Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

3

3



Giới thiệu JavaFX



Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

4

4



Viết ứng dụng với JavaFX

javafx.application.Application

+ start(primaryStage: Stage): void
+ launch(args: String[]): void



Lớp ứng dụng

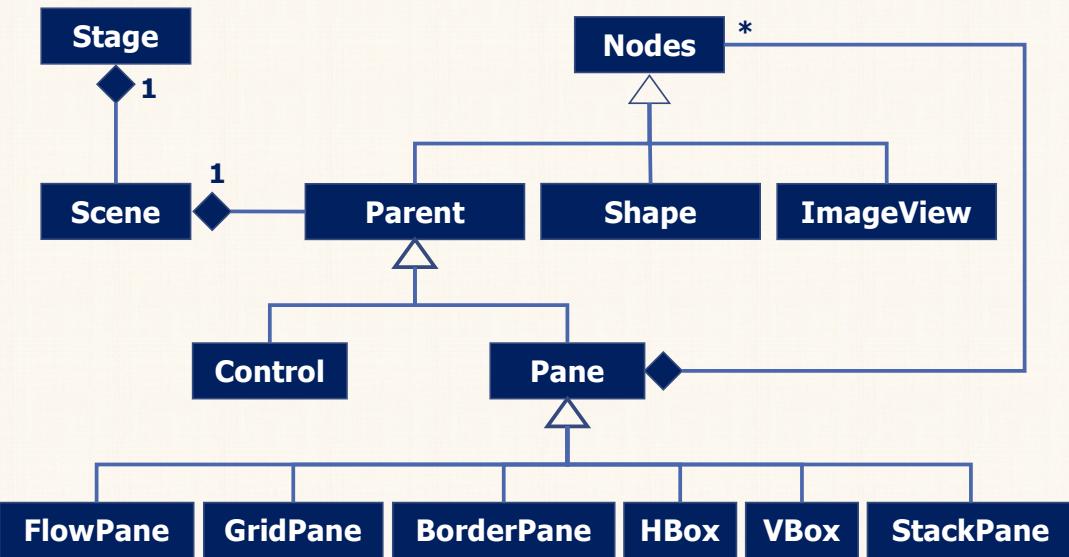


Viết ứng dụng với JavaFX

```
public class HelloApp extends Application {  
    @Override  
    public void start(Stage primaryStage) {  
        Button btn = new Button();  
        btn.setText("Hello, World");  
  
        Scene scene = new Scene(btn, 300, 250);  
  
        primaryStage.setTitle("Hello World!");  
        primaryStage.setScene(scene);  
        primaryStage.show();  
    }  
  
    public static void main(String[] args) {  
        Application.launch(args);  
    }  
}
```



Các thành phần ứng dụng JavaFX



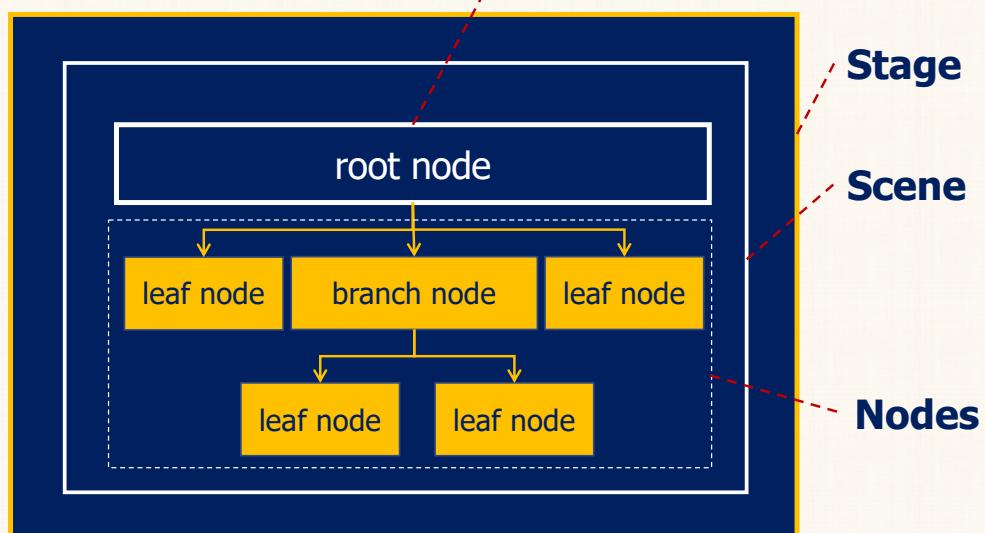
Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

7



Các thành phần ứng dụng JavaFX

Parent (Pane, Control)



Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

8



Các thành phần ứng dụng JavaFX

Stage là “*sân khấu*” để biểu diễn một “*cảnh diễn*” (**scene**) và **node** là các “*diễn viên*” diễn trong “*cảnh diễn*” (**scene**) đó

Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

9

9



Các thành phần ứng dụng JavaFX

```
@Override  
public void start(Stage primaryStage) {  
    Circle circle = new Circle(180, 125, 50);  
    circle.setStroke(Color.RED);  
    circle.setFill(Color.YELLOW);  
  
    Pane pane = new Pane();  
    pane.getChildren().add(circle);  
  
    Scene scene = new Scene(pane, 360, 250);  
  
    primaryStage.setTitle("Circle");  
    primaryStage.setScene(scene);  
    primaryStage.show();  
}
```

Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

10

10



Property Binding

- Property Binding cho phép một đối tượng đích (target object) gắn kết với đối tượng nguồn (source object), giá trị đối tượng nguồn thay đổi thì giá trị đối tượng đích cũng thay đổi theo tự động.

```
target.bind(source);
```

javafx.beans.property.Property

Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

11



Property Binding

```
@Override  
public void start(Stage primaryStage) {  
    Pane pane = new Pane();  
  
    Circle circle = new Circle();  
    circle.centerXProperty().bind(pane.widthProperty().divide(2));  
    circle.centerYProperty().bind(pane.heightProperty().divide(2));  
    circle.setRadius(50);  
    circle.setStroke(Color.RED);  
    circle.setFill(Color.YELLOW);  
  
    pane.getChildren().add(circle);  
    ...  
}
```

Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

12



Lớp Color

javafx.scene.paint.Color

- red: double
 - green: double
 - blue: double
 - opacity: double
- + Color(r: double, g: double, b: double, opacity: double)
- + brighter(): Color
- + darker(): Color
- + color(r: double, g: double, b: double): Color
- + color(r: double, g: double, b: double, opacity: double): Color
- + rgb(r: int, g: int, b: int): Color
- + rgb(r: int, g: int, b: int, opacity: double): Color



Lớp Font

javafx.scene.text.Font

- size: double
 - name: String
 - family: String
- + Font(size: double)
- + Font(name: String, size: double)
- + font(name: String, size: double)
- + font(name: String, w: FontWeight, size: double)
- + font(name: String, w: FontWeight, p: FontPosture, size: double)
- + getFamilies(): List<String>
- + getFontNames(): List<String>



Lớp Font

```
StackPane pane = new StackPane();
Circle circle = new Circle(150, 100, 60);

Label label = new Label("JavaFX");
Font font = Font.font("Arial", FontWeight.BOLD,
FontPosture.ITALIC, 25);
label.setFont(font);
label.setTextFill(Color.WHITE);
pane.getChildren().addAll(circle, label);

Scene scene = new Scene(pane, 300, 200);
primaryStage.setScene(scene);
primaryStage.show();
```



Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

15

15



Lớp Image và ImageView

javafx.scene.image.Image

- error: ReadOnlyBooleanProperty
- height: ReadOnlyBooleanProperty
- width: ReadOnlyBooleanProperty
- progress: ReadOnlyBooleanProperty
- + Image(filenameOrUrl: String)



javafx.scene.image.ImageView

- fitHeight: DoubleProperty
- fitWidth: DoubleProperty
- x: DoubleProperty
- y: DoubleProperty
- Image: ObjectProperty<Image>
- + ImageView()
- + ImageView(image: Image)
- + ImageView(filenameOrUrl: String)

Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

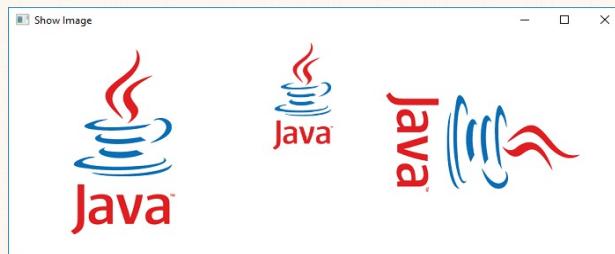
16

16



Lớp Image và ImageView

```
HBox pane = new HBox();
// Hình thứ nhất
Image image = new Image("images/java.png");
pane.getChildren().add(new ImageView(image));
// Hình thứ hai
ImageView imgView1 = new ImageView(image);
imgView1.setFitWidth(150);
imgView1.setFitHeight(150);
// Hình thứ ba
ImageView imgView2 = new ImageView(image);
imgView2.setRotate(90);
pane.getChildren().addAll(imgView1, imgView2);
```



Các loại Pane

Sắp nodes từng hàng theo chiều ngang hoặc từng cột theo chiều dọc

Sắp nodes theo dạng lưới hai chiều

Lớp cơ sở chứa phương thức getChildren() để lấy danh sách nodes trong pane

Sắp nodes nằm chồng lên nhau ở vị trí trung tâm

Pane

FlowPane

GridPane

BorderPane

HBox

VBox

StackPane

Sắp nodes ở các vùng trên, bên trái, dưới, bên phải hoặc trung tâm

Sắp nodes một dòng duy nhất

Sắp nodes một cột duy nhất



Các loại Pane

▪ Sử dụng FlowPane

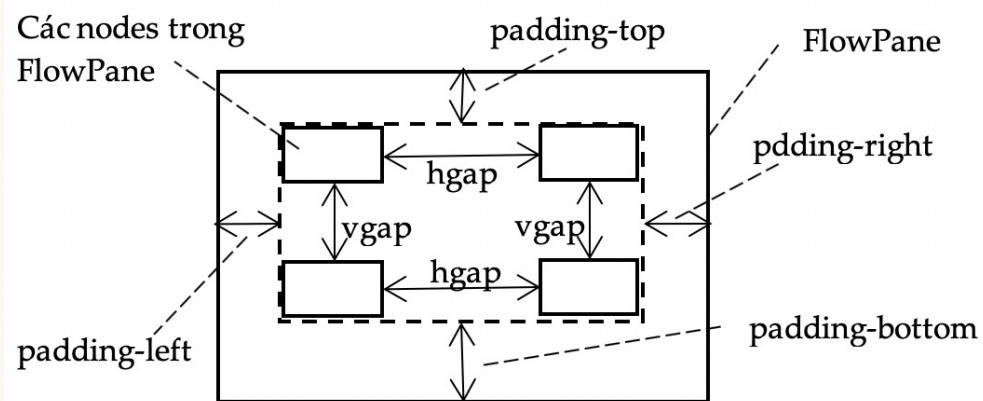
```
FlowPane pane = new FlowPane();
pane.setPadding(new Insets(15, 15, 15, 15));
pane.setAlignment(Pos.CENTER);
pane.setVgap(5);
pane.setHgap(5);

pane.getChildren().addAll(new Label("Nhập Số tiền:"),
                           new TextField(),
                           new Label("VNĐ"));
```



Các loại Pane

▪ Các thuộc tính FlowPane





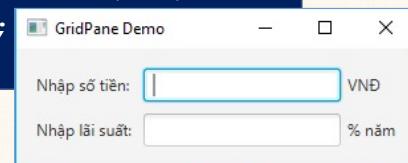
Các loại Pane

▪ Sử dụng GridPane

```
GridPane pane = new GridPane();
pane.setPadding(new Insets(15, 15, 15, 15));
pane.setAlignment(Pos.CENTER);
pane.setHgap(5);
pane.setVgap(5);

pane.add(new Label("Nhập số tiền: "), 0, 0);
pane.add(new TextField(), 1, 0);
pane.add(new Label("VNĐ"), 2, 0);

pane.add(new Label("Nhập lãi suất: "), 0, 1);
pane.add(new TextField(), 1, 1);
pane.add(new Label("% năm"), 2, 1);
```



Các loại Pane

▪ Sử dụng BorderPane, Hbox, VBox





Các loại Pane

▪ Phần đầu

```
HBox hpane = new HBox();
hpane.setAlignment(Pos.CENTER);
hpane.setStyle("-fx-background-color:gold;");
Image img = new Image("images/java.png");
ImageView logo = new ImageView(img);

Text subject = new Text("LẬP TRÌNH JAVAFX");
Font f = Font.font("Tahoma", FontWeight.BOLD, 48);
subject.setFont(f);
subject.setFill(Color.DARKBLUE);

hpane.getChildren().addAll(logo, subject);
```



Các loại Pane

▪ Phần bên trái

```
VBox vpane = new VBox();
String[] items = {
    "Sơ lược lịch sử JavaFX", "JavaFX căn bản", "..."
};
for (String item: items) {
    StackPane itemPane = new StackPane();
    Rectangle r = new Rectangle(200, 50);
    r.setFill(Color.BLUE);
    r.setStroke(Color.WHITE);
    Text t = new Text(item);
    t.setFill(Color.WHITE);
    t.setFont(Font.font("Tahoma", FontWeight.BOLD, 14));
    itemPane.getChildren().addAll(r, t);
    vpane.getChildren().add(itemPane);
}
```



Các loại Pane

- Phần nội dung

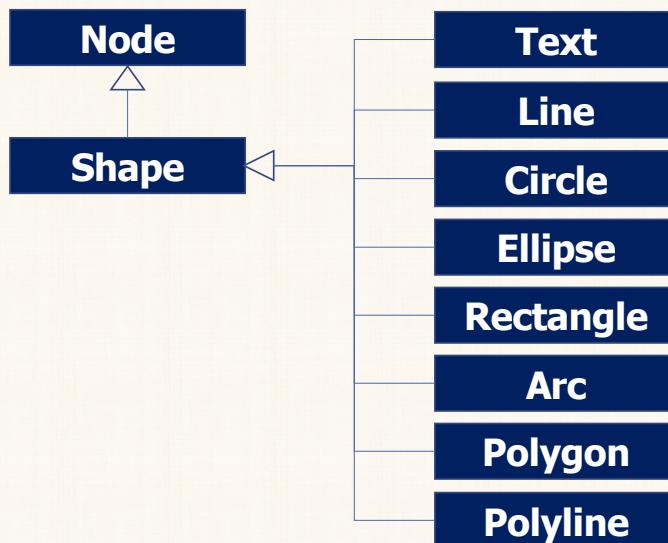
```
Text content = new Text("Chào mừng bạn đến với ứng  
dụng học JavaFX");  
content.setFont(Font.font("Tahoma",  
FontPosture.ITALIC, 20));  
content.setFill(Color.BLUE);
```

- Pane chính là BorderPane

```
BorderPane pane = new BorderPane();  
pane.setTop(hpane);  
pane.setLeft(vpane);  
pane.setCenter(content);
```



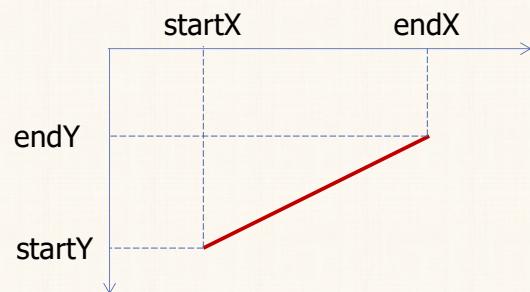
Shape



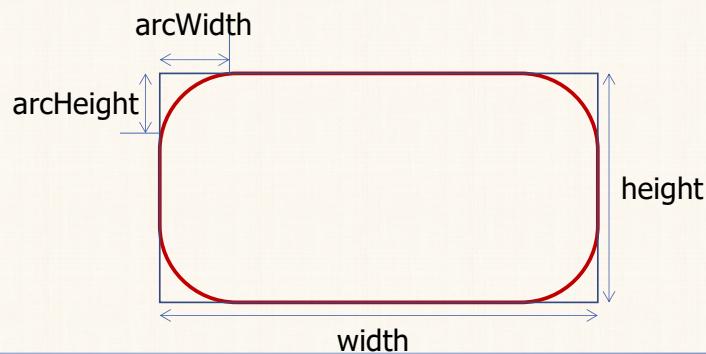


Shape

Line



Rectangle



Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

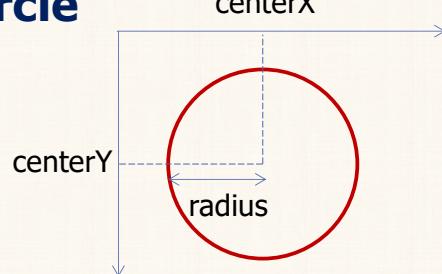
27

27

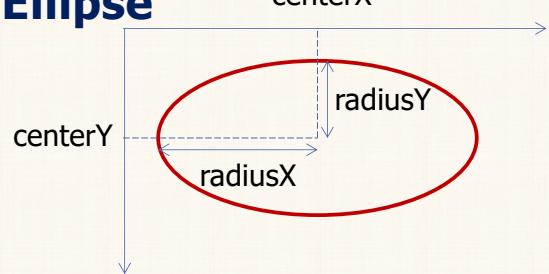


Shape

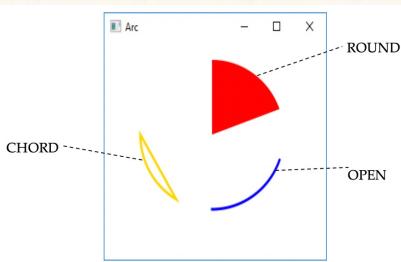
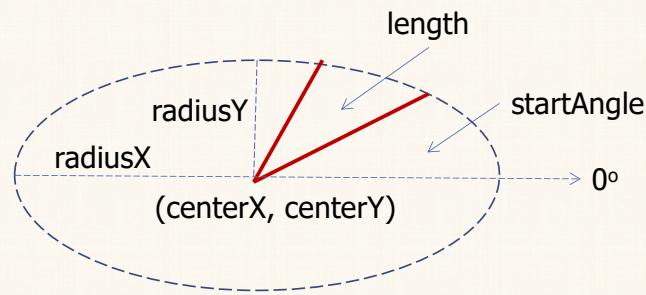
Circle



Ellipse



Arc



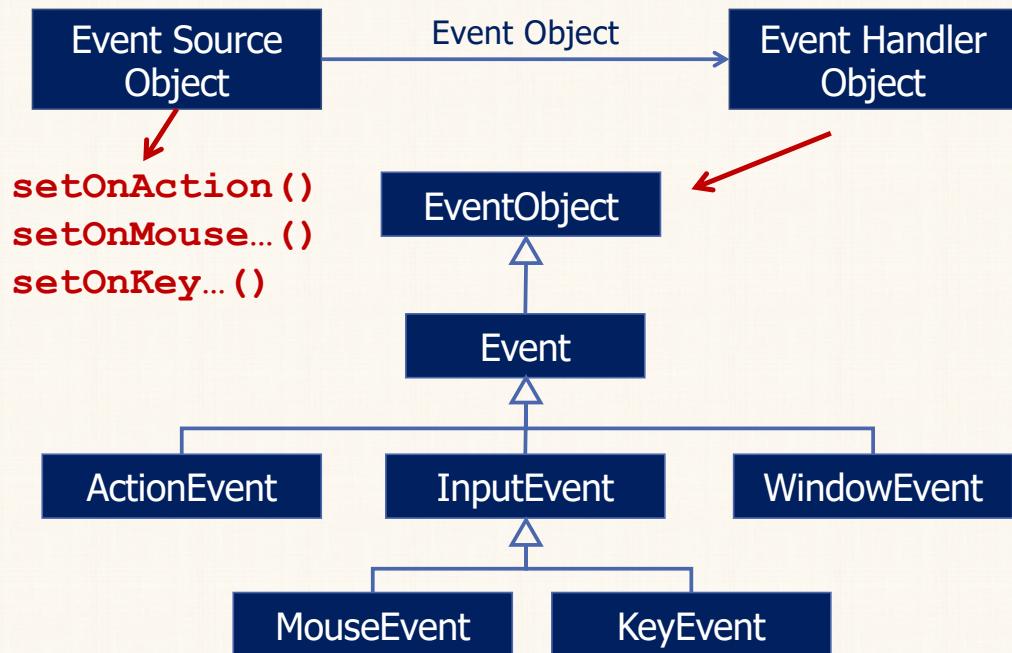
Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

28

28



Xử lý sự kiện



Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

29

29



Xử lý sự kiện

EventHandler<T extends Event>

+ handle(event: T): void



Lớp xử lý sự kiện

Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

30

30



Xử lý sự kiện

```
// Lớp ứng dụng
class MyApp extends Application {
    // ...
}

// Lớp xử lý sự kiện
class EventHandler implements EventHandler<T> {
    public void handle(T event) {
        // ...
    }
}
```



Xử lý sự kiện

```
// Lớp ứng dụng
class MyApp extends Application {
    // ...

    // Lớp xử lý sự kiện
    class EventHandler implements EventHandler<T> {
        public void handle(T event) {
            // ...
        }
    }
}
```

Inner Class



Xử lý sự kiện

setOnAction()
setOnMouse...()
setOnKey...()

```
sourceObject.setE(new EventHandler<T>() {  
    @Override  
    public void handle(ActionEvent event) {  
        // ...  
    }  
});
```

Anonymous
Inner Class



Xử lý sự kiện

setOnAction()
setOnMouse...()
setOnKey...()

```
sourceObject.setE(e -> {  
    // thân phương thức handle  
});
```

Biểu thức
Lambda



Sự kiện chuột

javafx.scene.input.MouseEvent

+ getButton(): MouseButton
+ getClickCount(): int
+ getX(): double
+ getY(): double
+ getSceneX(): double
+ getSceneY(): double
+ getScreenX(): double
+ getScreenY(): double
+ isAltDown(): boolean
+ isControlDown: boolean
+ isMetaDown(): boolean
+ isShiftDown(): boolean

Phím chuột đang click

Số lần click chuột

Phím Alt được bấm không

Phím Ctrl được bấm không

Phím Meta được bấm không

Phím Shift được bấm không



Sự kiện bàn phím

javafx.scene.input.KeyEvent

+ getCharacter(): String
+ getCode(): String
+ getText(): String
+ isAltDown(): boolean
+ isControlDown(): boolean
+ isMetaDown(): boolean
+ isShiftDown(): boolean

Trả về ký tự phím.

Trả về key code của phím là một giá trị trong enum KeyCode.

Trả về chuỗi mô tả key code.

Phím Alt được bấm không

Phím Ctrl được bấm không

Phím Meta được bấm không

Phím Shift được bấm không



Observable Listener

- Theo dõi thay đổi giá trị các thể hiện của giao diện Observable bằng phương thức `addListener()`.

javafx.beans.Observable
<<interface>>

+ `addListener(InvalidationListener invalidate)`

javafx.beans.InvalidationListener
<<interface>>

+ `invalidated(Observable observable)`



Hiệu ứng

- Lớp Animation

javafx.animation.Animation

- `autoReverse: BooleanProperty`
- `cycleCount: IntegerProperty`
- `rate: DoubleProperty`
- `status: ReadOnlyObjectProperty<Animation.status>`

- + `pause(): void`
- + `play(): void`
- + `stop(): void`

PAUSED, RUNNING, STOPPED

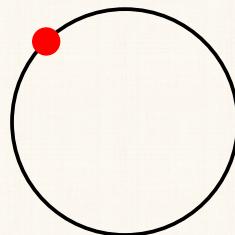


Hiệu ứng

▪ Lớp PathTransition

javafx.animation.PathTransition

- duration: ObjectProperty<Duration>
- node: ObjectProperty<Node>
- orientation: ObjectProperty<PathTransition.OrientationType>
- path: ObjectType<Shape>
- + PathTransition()
- + PathTransition(duration: Duration, path: Shape)
- + PathTransition(duration: Duration, path: Shape, node: Node)



Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

39

39

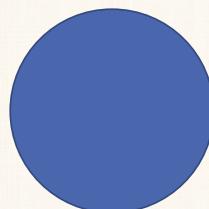


Hiệu ứng

▪ Lớp FadeTransition

javafx.animation.FadeTransition

- duration: ObjectProperty<Duration>
- node: ObjectProperty<Node>
- fromValue: DoubleProperty
- toValue: DoubleProperty
- byValue: DoubleProperty
- + FadeTransition()
- + FadeTransition(duration: Duration)
- + FadeTransition(duration: Duration, node: Node)



Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

40

40



Hiệu ứng

- Lớp Timeline dùng thực thi một KeyFrame sau mỗi khoảng thời gian (interval)
 - Phương thức khởi tạo Timeline

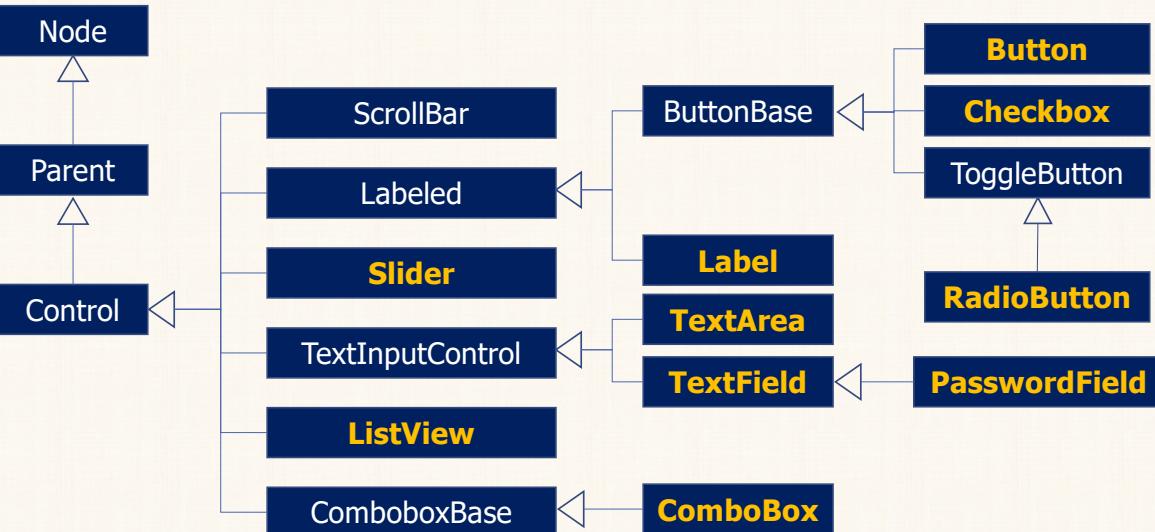
```
new Timeline(KeyFrame... keyframes)
```

- Tạo KeyFrame cho Timeline

```
new KeyFrame(Duration duration,  
EventHandler<ActionEvent> onFinished)
```



Các điều khiển thông dụng





Các điều khiển thông dụng

▪ Lớp Labeled

javafx.scene.control.Labeled

- alignment: ObjectProperty<Pos>
- contentDisplay: ObjectProperty<ContentDisplay>
- graphic: ObjectProperty<Node>
- graphicTextGap: DoubleProperty
- textFill: ObjectProperty<Paint>
- text: StringProperty
- underline: BooleanProperty
- wrapText: BooleanProperty



Các điều khiển thông dụng

Media

MediaPlayer

MediaView

```
File f = new File("src");
String filePath = "file:///\" + 
    f.getAbsolutePath().replaceAll("\\\\\", "/");
"/video/qua-tang-cuoc-song.mp4";
Media media = new Media(filePath);
MediaPlayer player = new MediaPlayer(media);
MediaView mediaView = new MediaView(player);
mediaView.setFitWidth(500);
mediaView.setFitHeight(400);
```



Dialog Box

Dialog

Alert

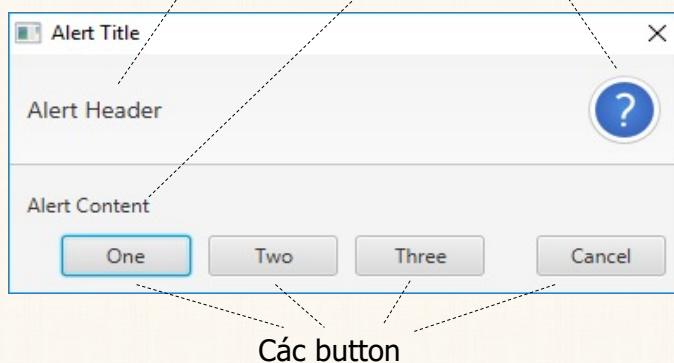
TextInputDialog

ChoiceDialog

header

content

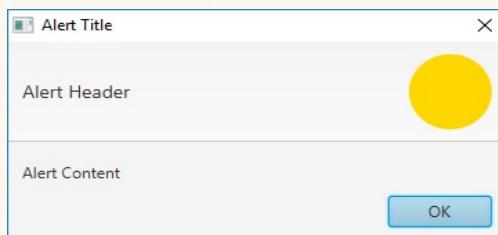
graphic



Dialog Box

▪ Ví dụ

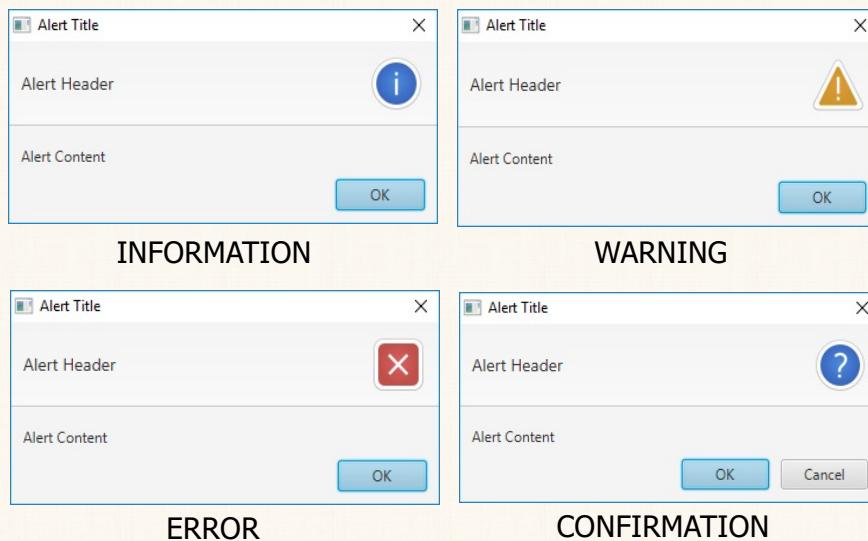
```
Alert a = new Alert(Alert.AlertType.INFORMATION);
a.setTitle("Alert title");
a.setHeaderText("Alert Header");
a.setContentText("Alert content");
a.setGraphic(new Circle(30, Color.GOLD));
a.showAndWait();
```





Dialog Box

▪ Các loại Dialog



Dialog Box

▪ Đối với loại CONFIRMATION, để biết người dùng đã chọn OK hay Cancel có thể dung một trong các cách sau:

```
Optional<ButtonType> re = a.showAndWait();
if (re.get() == ButtonType.OK)
    System.out.println("OK");
else if (re.get() == ButtonType.CANCEL)
    System.out.println("CANCEL");

a.showAndWait().ifPresent(res -> {
    if (res == ButtonType.OK)
        System.out.println("OK");
    else if (res == ButtonType.CANCEL)
        System.out.println("CANCEL");
});
```

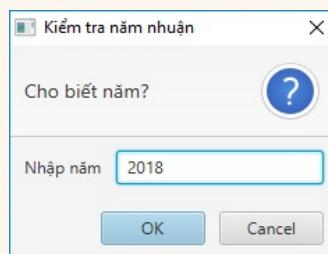


Dialog Box

▪ Sử dụng TextInputDialog

```
TextInputDialog d = new TextInputDialog();
d.setTitle("Kiểm tra năm nhuận");
d.setHeaderText("Cho biết năm?");
d.setContentText("Nhập năm");

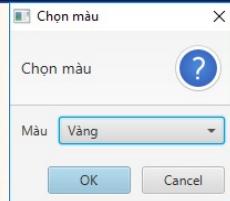
Optional<String> y = d.showAndWait();
if (y.isPresent()) // bấm OK trên dialog
    System.out.println(y.get());
```



Dialog Box

▪ Sử dụng ChoiceDialog

```
ArrayList<String> choices = new ArrayList<>();
choices.add("Xanh");
choices.add("Vàng");
ChoiceDialog c = new ChoiceDialog("Vàng", choices);
c.setTitle("Chọn màu");
c.setHeaderText("Chọn màu");
c.setContentText("Màu");
Optional<String> res = c.showAndWait();
if (res.isPresent()) // bấm OK trên dialog
    System.out.println("Màu " + res.get());
```





Vẽ biểu đồ

PieChart

Chart

XYChart

BarChart

LineChart

AreaChart



Vẽ biểu đồ

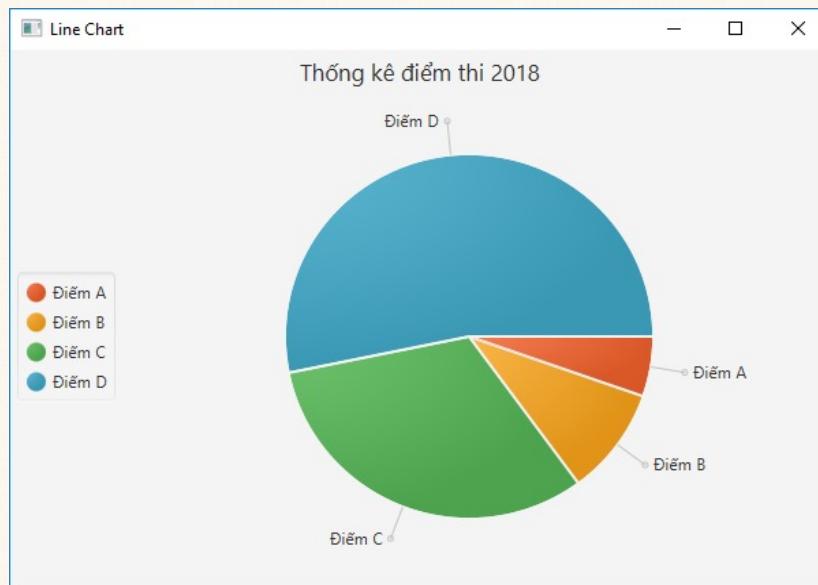
- Ví dụ vẽ biểu đồ tròn với số liệu như sau:

Điểm thi	Điểm A	Điểm B	Điểm C	Điểm D
Số lượng	50	90	300	500

```
StackPane pane = new StackPane();
PieChart c = new PieChart();
c.setTitle("Thông kê điểm thi 2018");
c.setLabelLineLength(25);
c.setLegendSide(Side.LEFT);
c.getData().add(new PieChart.Data("Điểm A", 50));
c.getData().add(new PieChart.Data("Điểm B", 90));
c.getData().add(new PieChart.Data("Điểm C", 300));
c.getData().add(new PieChart.Data("Điểm D", 500));
pane.getChildren().add(c);
```

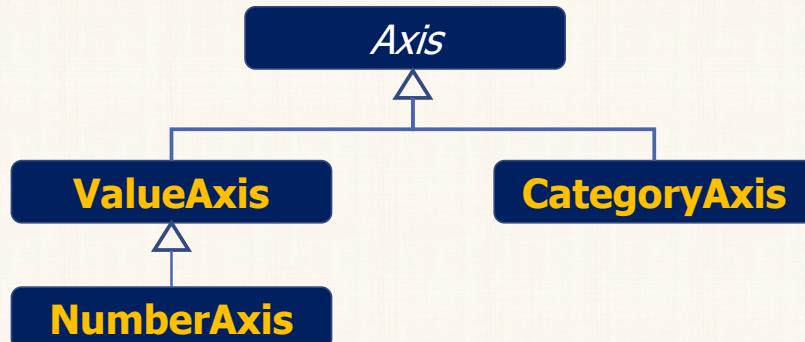


Vẽ biểu đồ



Vẽ biểu đồ

- Vẽ XYChart cần kết hợp với các lớp Axis để vẽ các trục ngang và trục dọc:
 - CategoryAxis: dùng khi các giá trị trục là tên danh mục.
 - NumberAxis: dùng khi các giá trị trục là số.





Vẽ biểu đồ

- Vẽ BarChart cho dữ liệu sau:

	A	B	C	D
2016	200	300	100	110
2017	300	100	50	150

- Tạo các trục

```
CategoryAxis xAxis = new CategoryAxis();
xAxis.setCategories(
    FXCollections.observableArrayList("2016", "2017")
);
xAxis.setLabel("Năm");
NumberAxis yAxis = new NumberAxis();
yAxis.setLabel("Số lượng");
```



Vẽ biểu đồ

```
BarChart chart = new BarChart(xAxis, yAxis);
XYChart.Series s1 = new XYChart.Series();
s1.setName("A");
s1.getData().add(new XYChart.Data("2016", 200));
s1.getData().add(new XYChart.Data("2017", 300));
XYChart.Series s2 = new XYChart.Series();
s2.setName("B");
s2.getData().add(new XYChart.Data("2016", 300));
s2.getData().add(new XYChart.Data("2017", 100));
XYChart.Series s3 = new XYChart.Series();
s3.setName("C");
s3.getData().add(new XYChart.Data("2016", 100));
s3.getData().add(new XYChart.Data("2017", 50));
XYChart.Series s4 = new XYChart.Series();
s4.setName("D");
s4.getData().add(new XYChart.Data("2016", 110));
s4.getData().add(new XYChart.Data("2017", 150));

chart.getData().addAll(s1, s2, s3, s4);
```

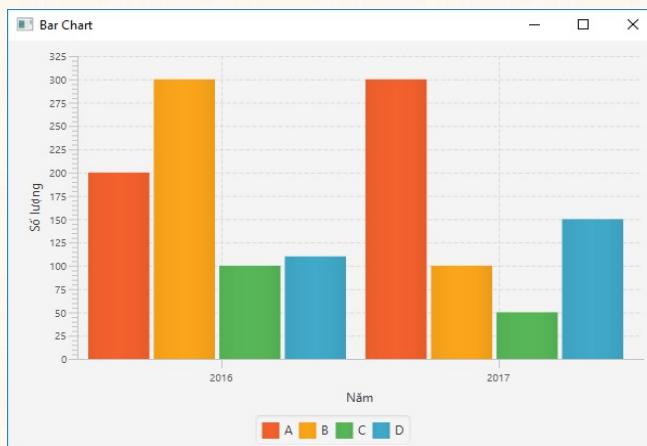


Vẽ biểu đồ

- Đặt chart lên StackPane

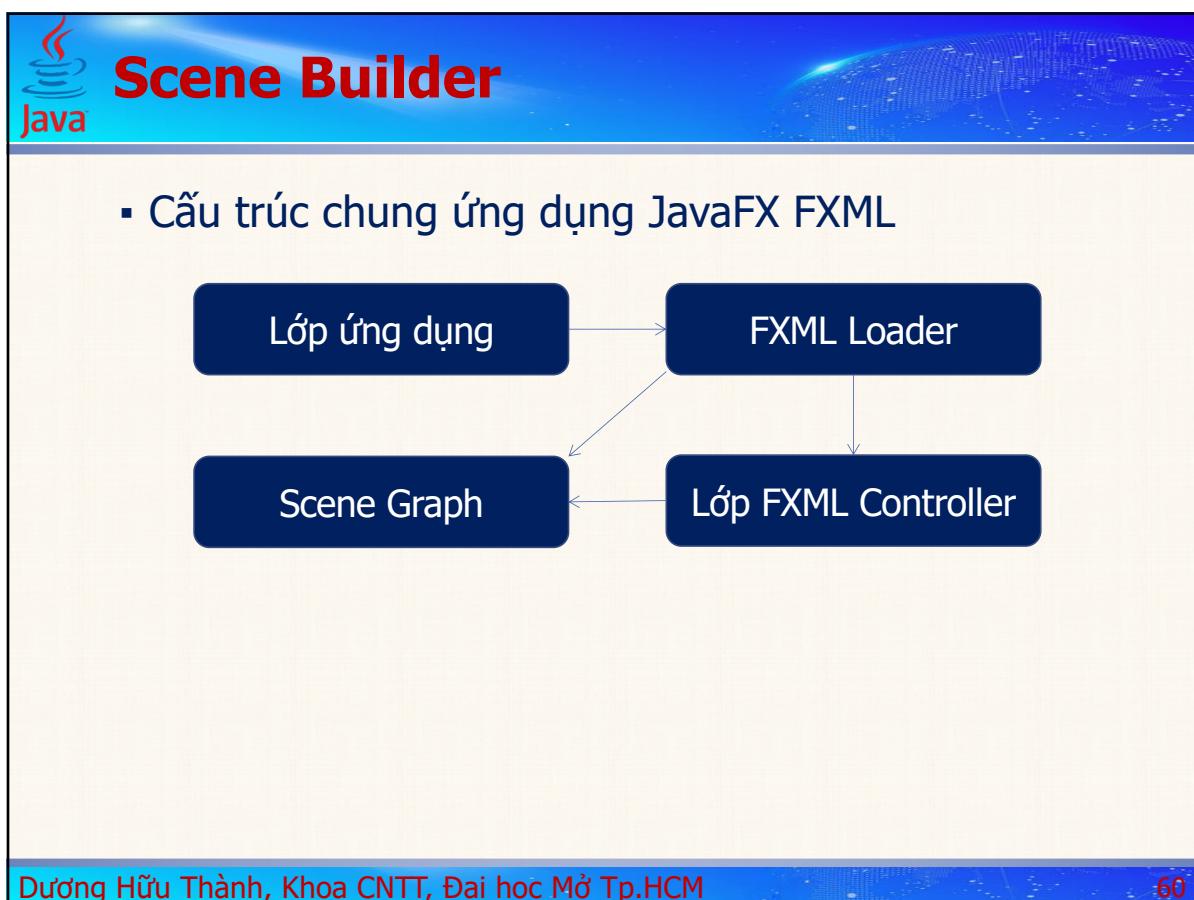
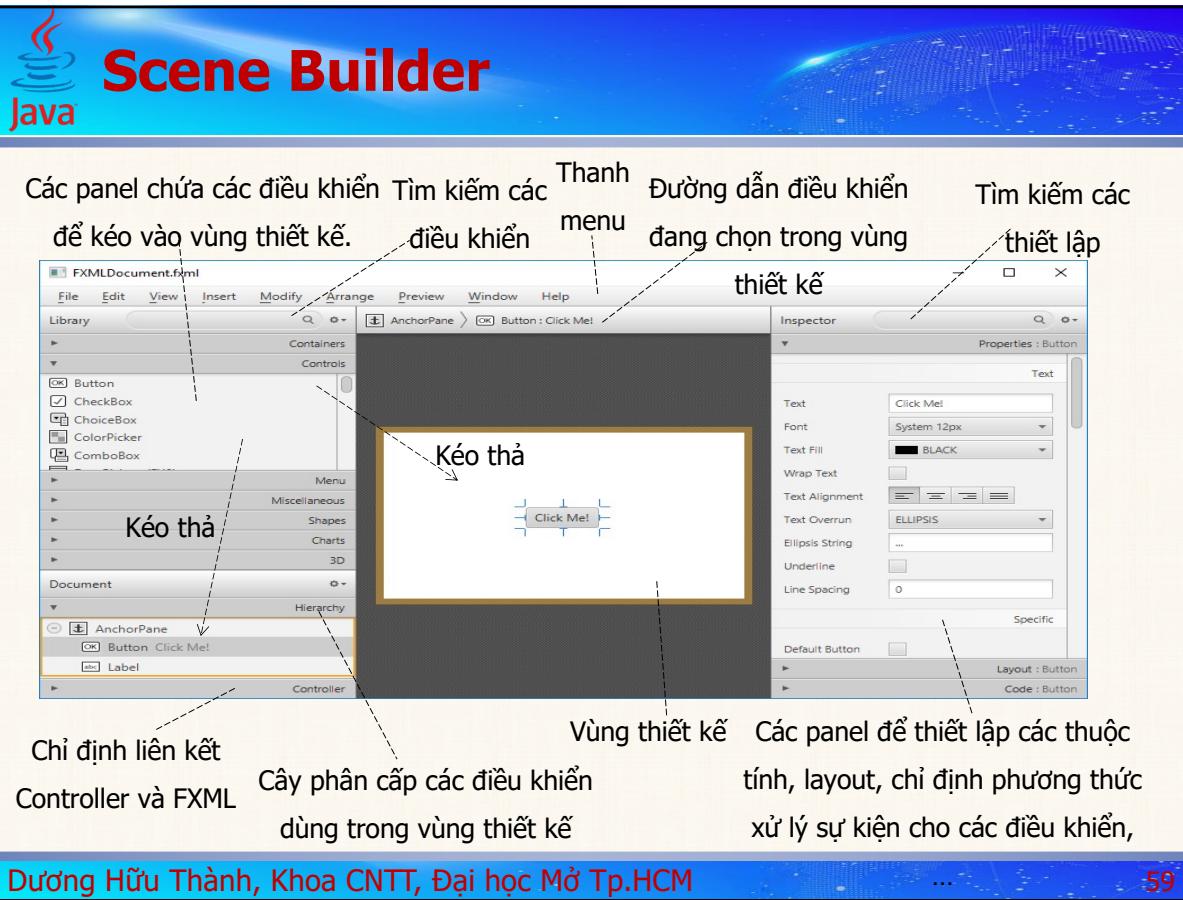
```
StackPane pane = new StackPane();  
pane.getChildren().add(chart);
```

- Kết quả



Scene Builder

- Scene Builder cung cấp môi trường thiết kế giao diện trực quan cho ứng dụng JavaFX.
- Scene Builder có thể được sử dụng độc lập hoặc kết hợp với các IDE như NetBeans hay Eclipse để phát triển ứng dụng với JavaFX.
- Khi thiết kế giao diện với Scene Builder thì mã nguồn FXML được sinh tự động trong IDE mà project được phát triển.
- Tải Scene Builder tại:
 - <https://gluonhq.com/products/scene-builder/>





FXML Loader

- FXML là ngôn ngữ có cú pháp tựa như XML dùng xây dựng giao diện người dùng cho ứng dụng JavaFX.
- Với FXML giúp cho việc xử lý logic và thiết kế giao diện có thể được phát triển độc lập, mỗi thành phần trong FXML là thể hiện các lớp, FXML Loader sẽ tạo các đối tượng đó.
- Cú pháp nạp tập tin FXML

```
URL url = getClass().getResource("<fxml-path>");  
FXMLLoader.load(url);
```



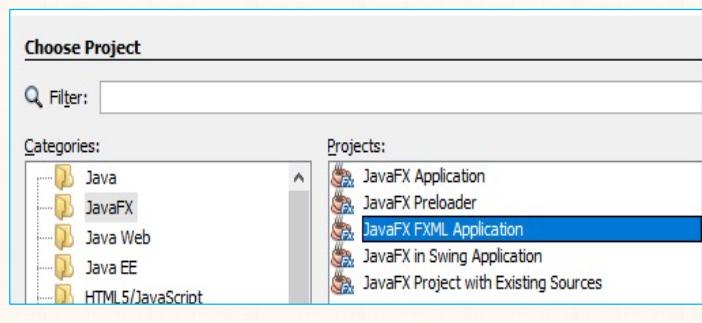
FXML Controller

- Các lớp controller cung cấp các logic xử lý cho các đối tượng thiết kế trong FXML.
- Mỗi thuộc tính trong lớp controller được gắn annotation `@FXML` để đánh dấu là đối tượng được định nghĩa trong FXML.
- Trong lớp controller, muốn truy cập các thành phần của tập tin FXML sử dụng thuộc tính `fx-id`.



Scene Builder

- Tạo project trong NetBeans



Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

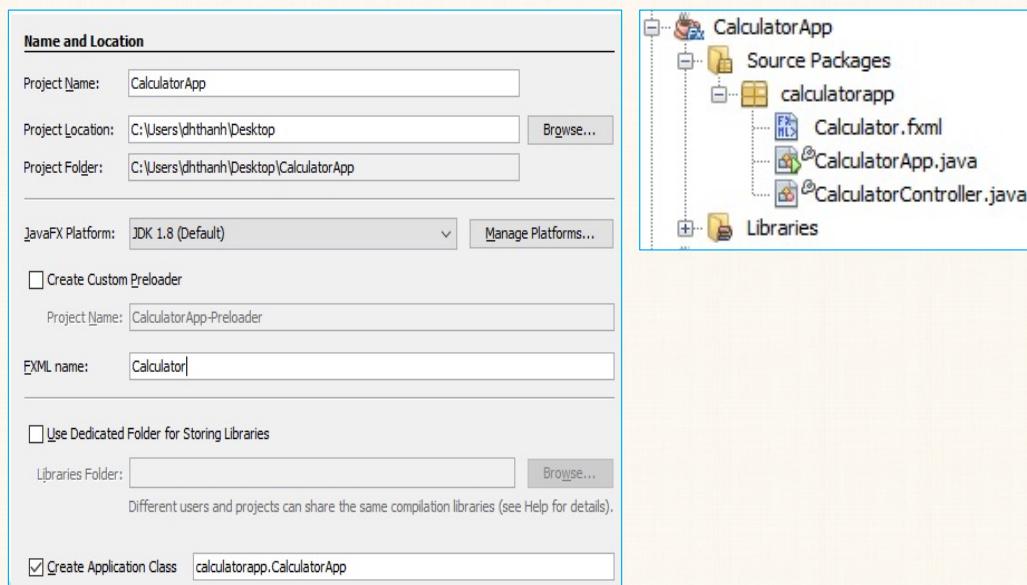
63

63



Scene Builder

- Đặt tên project



Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

64

64



Scene Builder

- Tạo tập tin style.css và liên kết ứng dụng

```
GridPane {  
    -fx-background-color: azure;  
    -fx-font-family: "Tahoma";  
    -fx-font-size: 14;  
    -fx-font-weight: bold;  
}  
  
HBox {  
    -fx-spacing:10;  
}  
  
Label, RadioButton, Button {  
    -fx-text-fill: brown;  
}  
  
Button {  
    -fx-background-color: gold;  
}
```

```
.mainTitle {  
    -fx-text-fill: brown;  
    -fx-font-size: 36;  
}
```



Scene Builder

- Double-click vào tập tin fxml để mở Scene Builder và kéo các điều khiển như sau:

Assigned fx:id	
fx:id	Component
btnTinh	Button
txtSo2	TextField
rdoCong	RadioButton
txtSo1	TextField
rdoTru	RadioButton
rdoNhan	RadioButton
Operation	ToggleGroup
txtKq	TextField
rdoChia	RadioButton



Q&A

Dương Hữu Thành, Khoa CNTT, Đại học Mở Tp.HCM

67