



جامعة طرابلس
كلية تقنية المعلومات
قسم هندسة البرمجيات



البرمجة المرئية Visual Programming ربيع 2025

المحاضرة السابعة



مواضيع المحاضرة

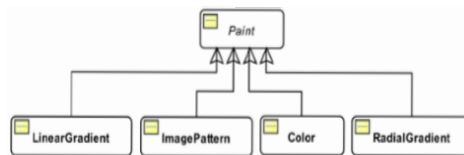
- ▶ كيف يتم تمثيل الألوان في JavaFX
- ▶ ما هي أنماط الألوان المختلفة
- ▶ كيفية استخدام نمط الصورة
- ▶ كيفية استخدام التدرج اللوني الخطي
- ▶ كيفية استخدام التدرج اللوني الشعاعي



الألوان Colors

- ▶ لتطبيق الألوان على أحد التطبيقات، توفر JavaFX فئات مختلفة في الحزمة `javafx.scene.paint`. تحتوي هذه الحزمة على فئة تسمى الرسام `paint` وهي الفئة الأساسية لجميع الفئات المستخدمة لتطبيق الألوان.
- ▶ باستخدام هذه الفئات ، يمكنك تطبيق الألوان في الأنماط التالية:
 - ▶ لون موحد
 - ▶ نمط الصورة
 - ▶ التدرج اللون.

- ▶ Uniform color.
- ▶ Image pattern.
- ▶ Color gradient.



▶ 3



الألوان Colors

- ▶ لون موحد **Uniform**: في هذا النمط ، يتم تطبيق اللون بشكل موحد في جميع أنحاء العقدة.
- ▶ نمط الصورة **Image Pattern**: يتيح لك هذا ملء منطقة العقدة بنمط صورة.
- ▶ التدرج اللوني **Gradient**: في هذا النمط ، يختلف اللون المطبق على العقدة من نقطة إلى أخرى.
- ▶ لديها نوعان من التدرجات وهما التدرج الخطي **Linear Gradient** والتدرج الشعاعي **Radial Gradient**.

▶ 4



الألوان Colors

- ▶ كل فئات العقد node classes التي يمكنك تطبيق اللون عليها مثل الشكل Text والنص Shape (بما في ذلك المشهد Scene) لها طرق باسم setFill() و setStroke().
- ▶ سيساعد ذلك في ضبط قيم الألوان للعقد وحدودها على التوالي.
- ▶ تقبل هذه الطرق كائنًا من النوع Paint.

▶ 5



استخدام فئة الألوان Colors

- ▶ تمثل فئة اللون لونًا solid من مساحة ألوان RGB.
- ▶ كل لون له قيمة محددة بين 0.0 إلى 1.0 أو من 0 إلى 255.
- ▶ قيمة اللون 0.0 أو 0 تعني أن اللون شفاف تمامًا، وقيمة اللون 1.0 أو 255 تشير إلى لون معتم تمامًا.
- ▶ بشكل افتراضي، يتم تعيين قيمة اللون على 1.0.
- ▶ يمكن الحصول على مثيل لفئة اللون بثلاث طرق:
 - ▶ باستخدام المنشئ constructor.
 - ▶ باستخدام إحدى طرق المصنع one of the factory methods.
 - ▶ استخدام أحد ثوابت اللون المعلنة في فئة اللون one of the color constants declared in the Color class

▶ 6



استخدام المنشئ constructor

- ▶ تحتوي فئة Color على مُنشئ واحد فقط يسمح لك بتحديد RGB والعتامة في النطاق بين 0.0 و 1.0:
- ▶ `public Color(double red, double green, double blue, double opacity)`
- ▶ يُنشئ الجزء البرمجي التالي لونًا أزرق معتمًا تمامًا:
- ▶ `Color blue = new Color(0.0, 0.0, 1.0, 1.0);`
- ▶ يمكنك استخدام الطرق الثابتة التالية في فئة Color لإنشاء كائنات ملونة.
- ▶ يجب أن تكون القيم بين 0.0 و 1.0 أو قيم صحيحة بين 0 و 255.

7



التعامل مع الألوان

- `Color color(double red, double green, double blue, double opacity)`
- `Color hsb(double hue, double saturation, double brightness)`
- `Color hsb(double hue, double saturation, double brightness, double opacity)` // السطوع التشبع التدرج
- `Color rgb(int red, int green, int blue)`
- `Color rgb(int red, int green, int blue, double opacity)`
- ▶ <https://htmlcolorcodes.com>

8



استخدام بعض الدوال

- ▶ باستخدام إحدى طرق المصنع
- ▶ تتيح لك أساليب المصنع `valueOf()` و `web()` إنشاء كائنات ملونة من سلاسل بتنسيقات قيم ألوان الويب. يُنشئ المقتطف التالي من التعليمات البرمجية كائنات زرقاء اللون باستخدام تنسيقات سلسلة مختلفة:

```
Color blue = Color.valueOf("blue");
Color blue = Color.web("blue");
Color blue = Color.web("#0000FF");
Color blue = Color.web("0X0000FF");
Color blue = Color.web("rgb(0, 0, 255)");
Color blue = Color.web("rgba(0, 0, 255, 0.5)"); // 50% transparent blue
```



استخدام أحد ثوابت اللون المعلنة في فئة اللون

- ▶ تحدد فئة اللون حوالي 140 من ثوابت اللون، على سبيل المثال: **RED**، **BLUE**، الألوان المحددة بواسطة هذه الثوابت غير شفافة تمامًا.

Color.BLUE



نمط الصورة Image pattern

```
public void start(Stage stage) {
    Image img = new Image("blueRect.png");

    ImagePattern p1 = new ImagePattern(img, 0, 0, 0.25, 0.25, true);
    Rectangle r1 = new Rectangle(100, 50);
    r1.setFill(p1);

    ImagePattern p2 = new ImagePattern(img, 0, 0, 0.5, 0.5, true);
    Rectangle r2 = new Rectangle(100, 50);
    r2.setFill(p2);

    ImagePattern p3 = new ImagePattern(img, 0, 0, 0.25, 0.25, true);
    Polygon triangle = new Polygon(50, 0, 0, 50, 100, 50);
    triangle.setFill(p3);

    ImagePattern p4 = new ImagePattern(img, 0, 0, 0.1, 0.1, true);
    Circle c = new Circle(50, 50, 25);
    c.setFill(p4);

    HBox root = new HBox();
    root.getChildren().addAll(r1, r2, triangle, c);
    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.setTitle("Using Image Patterns");
    stage.show();
}
```



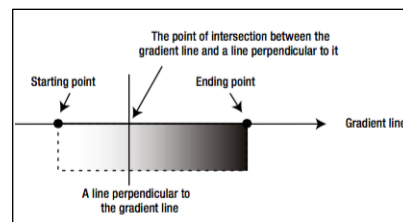
▶ 11



استخدام فئة LinearGradient

▶ في JavaFX، يمثل مثل فئة LinearGradient تدرج لوني خطي.

- `LinearGradient(double startX, double startY, double endX, double endY, boolean proportional, CycleMethod cycleMethod, List<Stop> stops)`
- `LinearGradient(double startX, double startY, double endX, double endY, boolean proportional, CycleMethod cycleMethod, Stop... stops)`



▶ 12



LinearGradient خصائص فئة

► يجب أن تكون إحداثيات x و y لنقطتي البداية والنهاية بين 0.0 و 1.0 (مزدوجة).

Property	Data Type / Description
startX	Double Set X coordinate of the gradient axis start point.
startY	Double Set Y coordinate of the gradient axis start point.
endX	Double Set X coordinate of the gradient axis end point.
endY	Double Set Y coordinate of the gradient axis end point.
proportional	Boolean Set whether the coordinates are proportional to the shape. true will use unit square coordinates, otherwise use screen coordinate system.
cycleMethod	CycleMethod Set cycle method applied to the gradient.
stops	List<Stop> Set stop list for gradient's color specification.

► 13



LinearGradient خصائص فئة

► قيمته هي أحد ثوابت التعداد المحددة في CycleMethod:

- CycleMethod.NO_CYCLE
- CycleMethod.REFLECT
- CycleMethod.REPEAT

تحدد وبسيطة التوقف نقاط توقف اللون على طول خط التدرج اللوني. يتم تمثيل نقطة توقف اللون بمثيل من فئة Stop، والذي يحتوي على مُنشئ واحد فقط: توقف (إزاحة مزدوجة، لون اللون)

```
Stop[] stops = new Stop[]{new Stop(0, Color.RED),
                           new Stop(0.25, Color.GREEN),
                           new Stop(0.50, Color.BLUE),
                           new Stop(0.75, Color.ORANGE),
                           new Stop(1, Color.YELLOW)};
LinearGradient lg = new LinearGradient(0, 0, 1, 0, true, NO_CYCLE, stops);
Rectangle r = new Rectangle(200, 100);
r.setFill(lg);
```



► 14



مثال

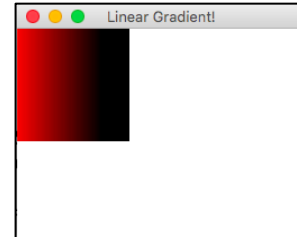
```
public void start(Stage primaryStage) {
    VBox box = new VBox();

    Stop[] stops = new Stop[] { new Stop(0, Color.RED),
                                new Stop(.75, Color.BLACK)};

    LinearGradient lg1 = new LinearGradient(
        0,
        0,
        1,
        0,
        true,
        CycleMethod.NO_CYCLE,
        stops);

    Rectangle r1 = new Rectangle(0, 0, 100, 100);
    r1.setFill(lg1);

    box.getChildren().add(r1);
}
```

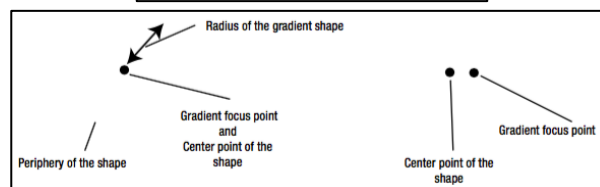
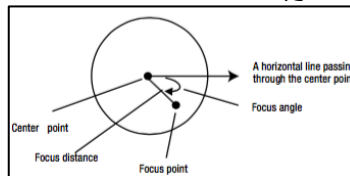


15



تدرج الألوان Radial Gradient

- فهم تدرج اللون الشعاعي في التدرج اللوني الشعاعي، تبدأ الألوان من نقطة واحدة، وتنتقل بسلسلة إلى الخارج في شكل دائري أو بيضاوي.
- يتم تعريف التدرج اللوني الشعاعي باستخدام ثلاثة مكونات: شكل متدرج (مركز ونصف قطر دائرة التدرج) نقطة تركيز لها أول لون من التدرج اللوني يتوقف اللون



16



تدرج الألوان Radial Gradient

Property	Data Type/Description
focusAngle	Double Set the angle in degrees from the center of the gradient to the focus point where the first color is mapped.
focusDistance	Double Set the distance from the center of the gradient to the focus point where the first color is mapped.
centerX	Double Set X coordinate of the center point of the gradient's circle.
centerY	Double Set Y coordinate of the center point of the gradient's circle.
radius	Double Set the radius of the circle of the color gradient.
proportional	boolean Set coordinates and sizes are proportional to the shape.
cycleMethod	CycleMethod Set the Cycle method applied to the gradient.
Stops	List<Stop> Set stop list for gradient's color

▶ 17



تدرج الألوان Radial Gradient

▶ يمثل مثيل فئة RadialGradient تدرج لوني شعاعي.

▶ انحدار شعاعي (زاوية تركيز مزدوج ، تركيز مزدوج المسافة ، مركز مزدوج X ، مركز مزدوج Y ، نصف قطر مزدوج ، نسبي منطقي ، طريقة دورة طريقة الدورة ، قائمة <إيقاف> توقف)

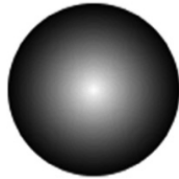
▶ 18



تدرج الألوان Radial Gradient

► RadialGradient زاوية تركيز مزدوجة ، تركيز مزدوج المسافة ، مركز مزدوج X ، مركز مزدوج Y ، نصف قطر مزدوج ، منطقي متناسب ، دورة طريقة دورة ، توقف ... توقف)

```
Stop[] stops = new Stop[]{new Stop(0, Color.WHITE), new Stop(1, Color.BLACK)};
RadialGradient rg = new RadialGradient(0, 0, 0.5, 0.5, 0.5, true, NO_CYCLE, stops);
Circle c = new Circle(50, 50, 50);
c.setFill(rg);
```



19

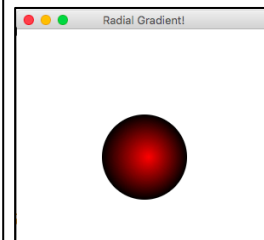


مثال

```
// Radial gradient color
RadialGradient gradient1 = new RadialGradient(0,
    .1,
    .5,
    .5,
    .5,
    true,
    CycleMethod.NO_CYCLE,
    new Stop(0, Color.RED),
    new Stop(1, Color.BLACK));

Circle c1 = new Circle(150, 150, 50);
c1.setFill(gradient1);

G.getChildren().add(c1);
```



20

نهاية المحاضرة

