



جامعة طرابلس  
كلية تقنية المعلومات  
قسم هندسة البرمجيات



## البرمجة المرئية Visual Programming ربيع 2025

المحاضرة الثالثة



### مواضيع المحاضرة

- ▶ الأشكال ثنائية الأبعاد 2D
- ▶ مقدمة
- ▶ Line – 2D Shapes
- ▶ Rectangle – 2D Shapes
- ▶ Circle – 2D Shapes
- ▶ Ellipse – 2D Shapes
- ▶ Polygon – 2D Shapes
- ▶ Arc – 2D Shapes
- ▶ تمارين



## الأشكال ثنائية الأبعاد 2D

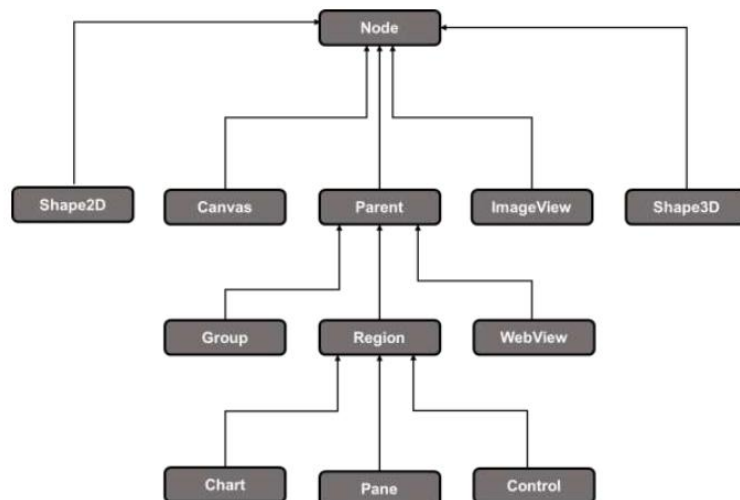
► أي شكل يمكن رسمه في بعدين two-dimensional يسمى الشكل ثنائي الأبعاد.

► تقدم JavaFX مجموعة متنوعة من nodes لرسم أنواع مختلفة من الأشكال (خطوط ، دوائر ، مستطيلات ، إلخ) باستخدام الحزمة `javafx.scene.shape.Shape` حيث الفئة `Shape` هي الفئة الأساس لكل الاشكال ثنائية البعد في JavaFX.

3



## الأشكال ثنائية الأبعاد 2D



4



## الأشكال ثنائية الأبعاد 2D

- ▶ لرسم الشكل الثنائي البعد نحتاج الى:
- ▶ تعريف الفئة المناسبة للشكل المطلوب
- ▶ تحديد خصائص الشكل مثل نقطة البداية ونقطة النهاية للخط او العرض والطول في المستطيل او طول نصف الدائرة في الدائرة.
- ▶ إضافة الشكل الى group.

▶ 5



## Line - 2D Shapes

- ▶ ضرورة تضمين الفئة `javafx.scene.shape.Line`
- ▶ لابد من تحديد نقطة البداية  $(x,y)$  ونقطه النهاية  $(x,y)$ .
- ▶ الاحداثي  $x$  يزداد كلما اتجهنا من الأعلى للأسفل
- ▶ والاحداثي  $y$  يزداد كلما اتجهنا من اليسار لليمين.
- ▶ استخدام قيم سالبة يمكن ان يدفع الشكل لخارج الشاشة.
- ▶ يمكن استخدام احدى الطريقتين التاليتين لرسم الخط:
- ▶ `Line(double startX, double startY, double endX, double endY)`
- ▶ `Line line = new Line();`  
`line.setStartX(value);`  
`line.setStartY(value);`  
`line.setEndX(value);`  
`line.setEndY(value);`

▶ 6



## Line - 2D Shapes

- ▶ حيث ان قيم جميع المعاملات بالطريقة الأولى من نوع double.
- ▶ قيمة العرض الافتراضي للخط `strokeWidth` هو 1.0 ويمكن تعديله باستخدام الدالة `setStrokeWidth()`
- ▶ اللون الافتراضي لاطار الشكل هو اللون الأسود `Color.BLACK` ويمكن تغييره باستخدام الدالة `setStroke()`
- ▶ اللون الافتراضي لملء الشكل هو `null` للخط أي بدون لون. بينما هو `Color.BLACK` لباقي الاشكال في هذه المحاضرة. ويمكن تغيير القيمة الافتراضية باستخدام الدالة `setFill()`
- ▶ يتم استخدام `built-in JavaFX colors` الموجودة بحزمة `javafx.scene.paint.Color` لاضافة قيم اللون للاطار او الحشوة باستخدام الدوال `setStroke()` و `setFill()` على التوالي
- ▶ يمكن أيضا تحديد شكل نهاية الخط باستخدام الخاصية `setStrokeLineCap()` التي لها احدى القيم التالية `StrokeLineCap.BUTT` أو `StrokeLineCap.ROUND` أو `StrokeLineCap.SQUARE`

7

```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.shape.Line;
import javafx.stage.Stage;
public class Lect3Ex5 extends Application{

    public void start(Stage stage) {

        Line line = new Line();

        line.setStartX(100.0);
        line.setStartY(150.0);
        line.setEndX(500.0);
        line.setEndY(150.0);

        Group root = new Group(line);

        Scene scene = new Scene(root, 600, 300);

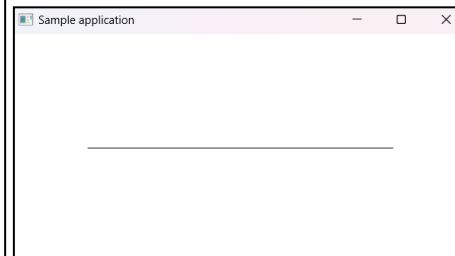
        stage.setTitle("Sample application");

        stage.setScene(scene);

        stage.show();
    }

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

مثال



مثال

```

public class Lect3Ex4 extends Application {

    public void start(Stage primaryStage) {
        primaryStage.setTitle("Drawing Lines");
        Group root = new Group();
        Scene scene = new Scene(root, 300, 150, Color.GRAY); //Width & Height

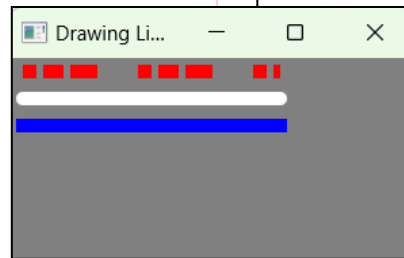
        // Red line
        Line redLine = new Line(10, 10, 200, 10);
        redLine.setStroke(Color.RED);
        redLine.setStrokeWidth(10);
        redLine.setStrokeLineCap(StrokeLineCap.BUTT); // SQUARE or ROUND
        redLine.getStrokeDashArray().addAll(10d, 5d, 15d, 5d, 20d, 30d); //dash gap pattern
        root.getChildren().add(redLine);

        // White line
        Line whiteLine = new Line(10, 30, 200, 30);
        whiteLine.setStroke(Color.WHITE);
        whiteLine.setStrokeWidth(10);
        whiteLine.setStrokeLineCap(StrokeLineCap.ROUND);
        root.getChildren().add(whiteLine);

        // Blue line
        Line blueLine = new Line(10, 50, 200, 50);
        blueLine.setStroke(Color.BLUE);
        blueLine.setStrokeWidth(10);
        root.getChildren().add(blueLine);

        primaryStage.setScene(scene);
        primaryStage.show();
    }
}

```



## Rectangle - 2D Shapes

- ▶ يتم تمثيل rectangle node من Rectangle class بحزمة `javafx.scene.shape.Rectangle`
- ▶ لهذه الفئة أربعة خصائص بقيم من نوع `double` والخصائص هي: نقطة البداية بأحداثيات `x` و `y` والتي تكون في الزاوية العلوية اليسرى للمستطيل مع تحديد قيمة العرض والطول.
- ▶ يمكن تحديد قيم هذه الخصائص بإحدى الطريقتين:
  - ▶ `Rectangle rectangle = new Rectangle(x, y, width, height);`
  - ▶ `Rectangle rectangle = new Rectangle();`  
`rectangle.setX(value);`  
`rectangle.setY(value);`  
`rectangle.setWidth(value);`  
`rectangle.setHeight(value);`

## مثال

```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.stage.Stage;
import javafx.scene.shape.Rectangle;

public class Lect3Ex6 extends Application {
    public void start(Stage stage) {
        Rectangle rectangle = new Rectangle();
        rectangle.setX(150.0);
        rectangle.setY(75.0);
        rectangle.setWidth(300.0);
        rectangle.setHeight(150.0);

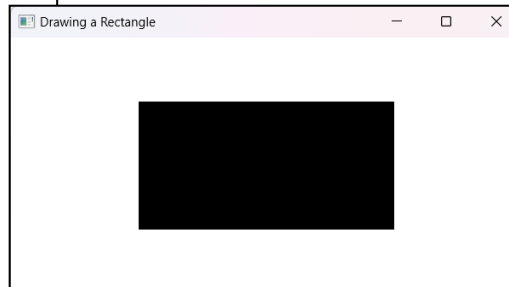
        Group root = new Group(rectangle);

        Scene scene = new Scene(root, 600, 300);
        stage.setTitle("Drawing a Rectangle");

        stage.setScene(scene);

        stage.show();
    }

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



▶ ||



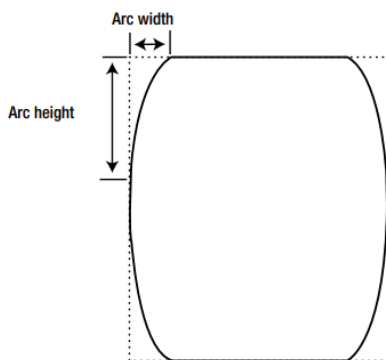
## Rounded Rectangle

▶ دائما ما يظهر المستطيل بحواف حادة الا اذا تم استخدام الخاصيتين `arcWidth` و `arcHeight` التي تجعل شكل الحواف دائري باستخدام `setArcWidth()` و `setArcHeight()` الدوال

▶ القيم الافتراضية لجميع الخصائص هي الصفر

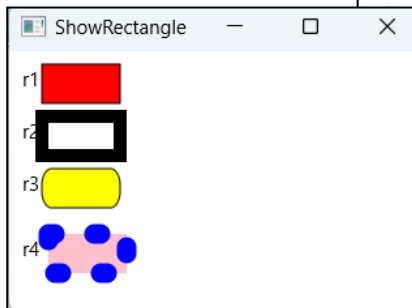
▶ ماذا اذا استخدمت احدى الخاصيتين فقط؟

▶ كيف تستخدم الخاصيتين لتحويل الشكل من شكل مستطيل الى شكل بيضاوي؟



▶ 12

## مثال



```
public class Lect3Ex8 extends Application {
    public void start (Stage primaryStage) {
        Group root = new Group();
        Rectangle rect1 = new Rectangle(25, 10, 60, 30);
        rect1.setStroke(Color.BLACK);
        rect1.setFill(Color.RED);
        root.getChildren().add(new Text(10,27, "r1"));
        root.getChildren().add(rect1);

        Rectangle rect2 = new Rectangle(25, 50, 60, 30);
        rect2.setStroke(Color.BLACK);
        rect2.setStrokeWidth(10);
        rect2.setFill(Color.TRANSPARENT);
        root.getChildren().add(new Text(10,67, "r2"));
        root.getChildren().add(rect2);

        Rectangle rect3 = new Rectangle(25, 90, 60, 30);
        rect3.setArcWidth(15);
        rect3.setArcHeight(25);
        rect3.setStroke(Color.BLACK);
        rect3.setFill(Color.YELLOW);
        root.getChildren().add(new Text(10,107, "r3"));
        root.getChildren().add(rect3);

        Rectangle rect4 = new Rectangle(30, 140, 60, 30);
        rect4.setStroke(Color.BLUE);
        rect4.setFill(Color.PINK);
        rect4.setStrokeWidth(15);
        rect4.setStrokeLineCap(StrokeLineCap.ROUND);
        rect4.setStrokeDashArray().addAll(5d, 30d);
        root.getChildren().add(new Text(10,157, "r4"));
        root.getChildren().add(rect4);

        Scene scene = new Scene(root, 320, 200);
        primaryStage.setTitle("ShowRectangle");
        primaryStage.setScene(scene);
        primaryStage.show();
    }
}
```

▶ I3



## Circle - 2D Shapes

- ▶ يتم تمثيل الدائرة بفئة Circle التي ينتمي للحزمة `.javafx.scene.shape.Circle`
- ▶ لفئة Circle ثلاث خصائص من النوع `double` هي: `centerX` و `centerY` و `radius` حيث تمثل القيمتان `centerX` و `centerY` إحداثيات مركز الدائرة والقيمة `radius` هي قيمة نصف قطر الدائرة.
- ▶ يمكن استخدام أحد الطرق الآتية لبناء `circle`:
- ▶ `Circle circle = new Circle(centerX, centerY, radius);`
- ▶ `Circle circle = new Circle();`  
`circle.setCenterX(value);`  
`circle.setCenterY(value);`  
`circle.setRadius(value);`

▶ I4

مثال

```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.stage.Stage;
import javafx.scene.shape.Circle;

public class Lect3Ex9 extends Application {
    public void start(Stage stage) {

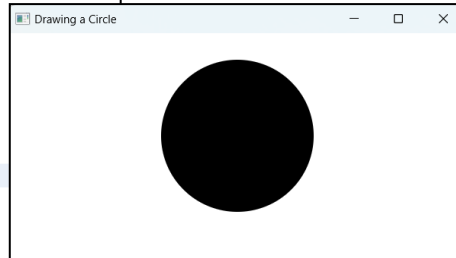
        Circle circle = new Circle();

        circle.setCenterX(300.0);
        circle.setCenterY(135.0);
        circle.setRadius(100.0);

        Group root = new Group(circle);

        Scene scene = new Scene(root, 600, 300);
        stage.setTitle("Drawing a Circle");
        stage.setScene(scene);
        stage.show();
    }

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



▶ 15

مثال

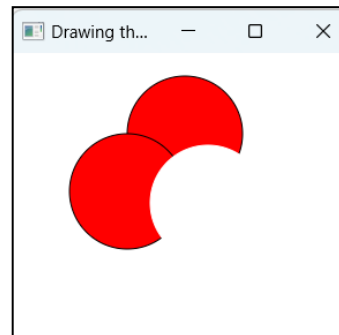
```
public class Lect3Ex10 extends Application {
    public void start(Stage stage) {

        Circle c1 = new Circle();
        c1.setRadius(50);
        c1.setCenterX(150);
        c1.setCenterY(70);
        c1.setFill(Color.RED);
        c1.setStroke(Color.BLACK);

        Circle c2 = new Circle();
        c2.setRadius(50);
        c2.setCenterX(100);
        c2.setCenterY(120);
        c2.setFill(Color.RED);
        c2.setStroke(Color.BLACK);

        Circle c3 = new Circle();
        c3.setRadius(50);
        c3.setCenterX(170);
        c3.setCenterY(130);
        c3.setFill(Color.WHITE);
        c3.setStroke(Color.WHITE);

        Group root = new Group();
        root.getChildren().addAll(c1,c2,c3);
        Scene scene = new Scene(root, 300, 250);
        stage.setTitle("Drawing three Circle");
        stage.setScene(scene);
        stage.show();
    }
}
```



▶ 16





## Ellipse – 2D Shapes

- ▶ يتم تمثيل الدائرة بفئة Ellipse التي ينتمي للحزمة javafx.scene.shape.Ellipse.
- ▶ Circle هي حالة خاصة من Ellipse
- ▶ لفئة Ellipse أربعة خصائص من النوع double هي: centerX و centerY و radiusX و radiusY.
- ▶ يمكن استخدام أحد الطرق الآتية لبناء ellipse:
- ▶ `Ellipse ellipse=new Ellipse (centerX, centerY, radiusX, radiusY);`
- ▶ `Ellipse ellipse = new Ellipse ( )`  
`ellipse.setCenterX(value);`  
`ellipse.setCenterY(value);`  
`ellipse.setRadiusX(value);`  
`ellipse.setRadiusY(value);`

17

مثال

```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.stage.Stage;
import javafx.scene.shape.Ellipse;

public class Lect3Ex11 extends Application {

    @Override
    public void start(Stage stage) {

        Ellipse ellipse = new Ellipse();

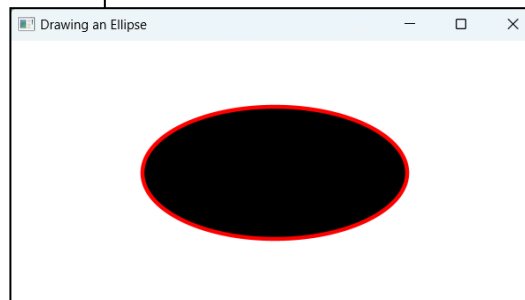
        ellipse.setCenterX(300.0);
        ellipse.setCenterY(150.0);
        ellipse.setRadiusX(150.0);
        ellipse.setRadiusY(75.0);

        ellipse.setStroke(Color.RED);
        ellipse.setStrokeWidth(4.5);

        Group root = new Group(ellipse);

        Scene scene = new Scene(root, 600, 300);
        stage.setTitle("Drawing an Ellipse");
        stage.setScene(scene);
        stage.show();
    }

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



مثال

```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Ellipse;
import javafx.stage.Stage;

public class Lec3Ex12 extends Application {

    @Override
    public void start(Stage primaryStage) {
        Group root = new Group();

        Circle sun = new Circle(250, 150, 50, Color.YELLOW);

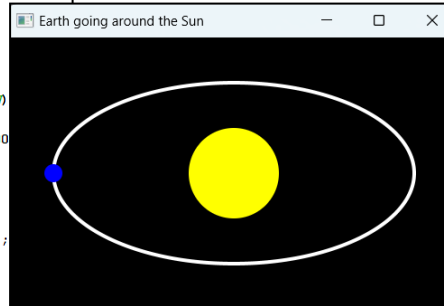
        Ellipse earthOrbit = new Ellipse(250, 150, 200, 100);
        earthOrbit.setStroke(Color.WHITE);
        earthOrbit.setStrokeWidth(4.0);
        earthOrbit.setFill(Color.TRANSPARENT);

        Circle earth = new Circle(50, 150, 10, Color.BLUE);

        root.getChildren().addAll(sun, earthOrbit, earth);

        Scene scene = new Scene(root, 500, 300, Color.BLACK);
        primaryStage.setScene(scene);
        primaryStage.setTitle("Earth going around the Sun");
        primaryStage.show();
    }

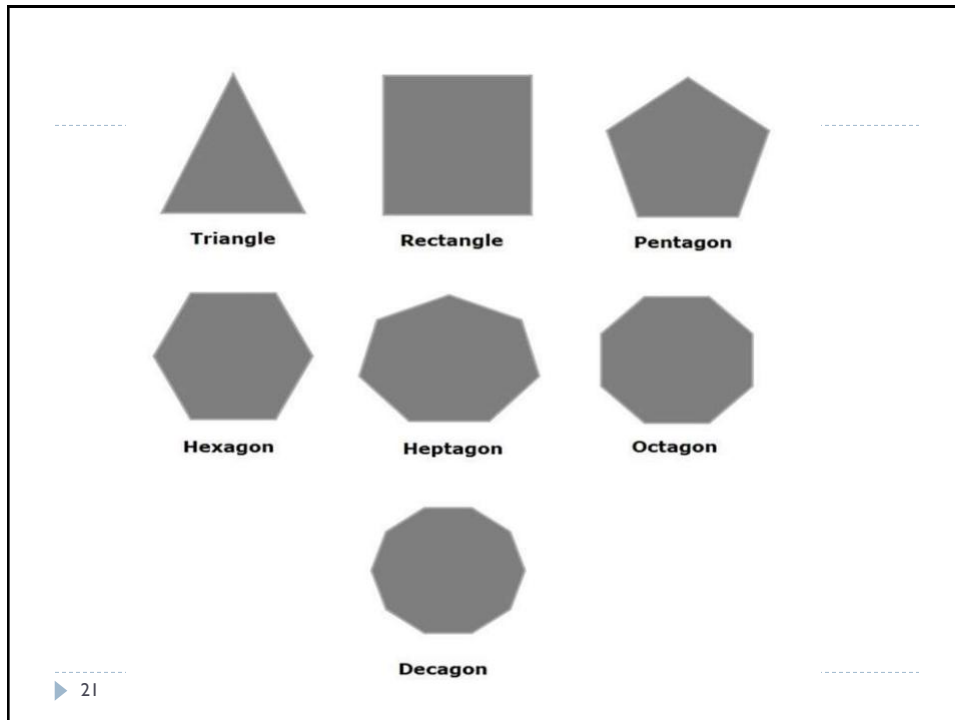
    public static void main(String[] args) {
        launch(args);
    }
}
```



## كيفية رسم المضلع Polygons في 2D

- ▶ يتم تمثيل المضلع بفئة Polygon التي ينتمي للحزمة javafx.scene.shape.Polygon.
- ▶ عليك تحديد عدة نقاط بالاحداثيات x و y لتكوين المضلع.
- ▶ يمكن استخدام احد الطرق الاتية لبناء polygon:

- › Polygon polygon = new Polygon(doubleArray);
- › Polygon polygon = new Polygon( );  
polygon.getPoints().addAll(new Double[ ] {  
List of XY coordinates separated by commas });



```

import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.shape.Polygon;
import javafx.stage.Stage;
public class Lect3Ex13 extends Application {
    @Override
    public void start(Stage stage) {

        Polygon polygon = new Polygon();

        polygon.getPoints().addAll(new Double[]{
            300.0, 50.0,
            450.0, 150.0,
            300.0, 250.0,
            150.0, 150.0,
        });

        Group root = new Group(polygon);

        Scene scene = new Scene(root, 600, 300);
        stage.setTitle("Drawing a Polygon");
        stage.setScene(scene);
        stage.show();
    }
    public static void main(String args[]){
        launch(args);
    }
}

```

مثال

▶ 22

```

import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.shape.Polygon;
import javafx.stage.Stage;

public class Lect3Ex14 extends Application {
    public void start(Stage stage) {
        Polygon polygon = new Polygon();

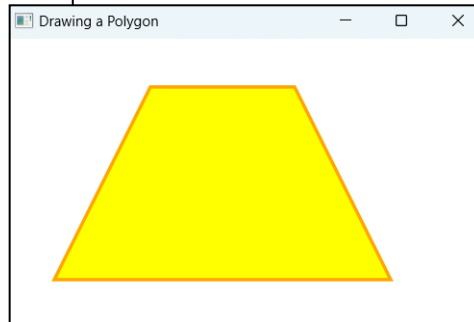
        polygon.getPoints().addAll(new Double[]{
            300.0, 50.0,
            150.0, 50.0,
            50.0, 250.0,
            400.0, 250.0,
        });

        polygon.setStroke(Color.ORANGE);
        polygon.setFill(Color.YELLOW);
        polygon.setStrokeWidth(3.5);
        Group root = new Group(polygon);

        Scene scene = new Scene(root, 500, 300);

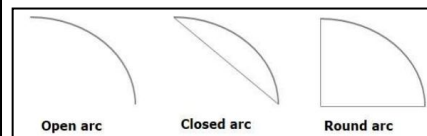
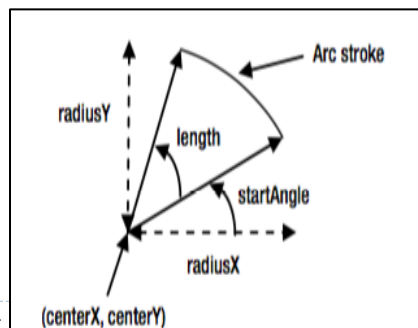
        stage.setTitle("Drawing a Polygon");
        stage.setScene(scene);
        stage.show();
    }
    public static void main(String args[]){
        launch(args);
    }
}

```



## كيفية رسم الأقواس Arcs في 2D

- ▶ يتم تمثيل القوس من الفئة Arc class.
- ▶ يتم تمثيل الأقواس Arc كجزء من curve.
- ▶ لتعريف القوس نستخدم الخصائص: centerX و centerY و radiusX و radiusY و startAngle و length و Type
- ▶ حيث أنواع القوس هي: ArcType.OPEN أو ArcType.CHORD أو ArcType.ROUND

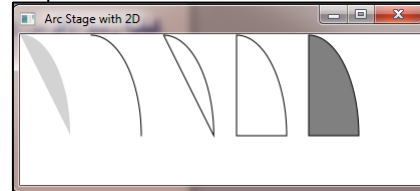


مثال

```

public void start(Stage primaryStage) {
    primaryStage.setTitle(" Arc Stage with 2D ");
    Arc A1 = new Arc(0, 0, 50, 100, 0, 90);
    A1.setFill(Color.LIGHTGRAY);
    Arc A2 = new Arc(0, 0, 50, 100, 0, 90);
    A2.setFill(Color.TRANSPARENT);
    A2.setStroke(Color.BLACK);
    Arc A3 = new Arc(0, 0, 50, 100, 0, 90);
    A3.setFill(Color.TRANSPARENT);
    A3.setStroke(Color.BLACK);
    A3.setType(ArcType.CHORD);
    Arc A4 = new Arc(0, 0, 50, 100, 0, 90);
    A4.setFill(Color.TRANSPARENT);
    A4.setStroke(Color.BLACK);
    A4.setType(ArcType.ROUND);
    Arc A5 = new Arc(0, 0, 50, 100, 0, 90);
    A5.setFill(Color.GRAY);
    A5.setStroke(Color.BLACK);
    A5.setType(ArcType.ROUND);
    HBox root = new HBox(A1, A2, A3, A4, A5);
    root.setSpacing(20);
    Scene scene = new Scene(root, 400, 150);
    primaryStage.setScene(scene);
    primaryStage.show();
}

```



25

```

import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.stage.Stage;
import javafx.scene.shape.Arc;
import javafx.scene.shape.ArcType;

public class Lect3Ex15 extends Application {

    public void start(Stage stage) {

        Arc arc = new Arc();

        arc.setCenterX(300.0);
        arc.setCenterY(150.0);
        arc.setRadiusX(90.0);
        arc.setRadiusY(90.0);
        arc.setStartAngle(40.0);
        arc.setLength(239.0);

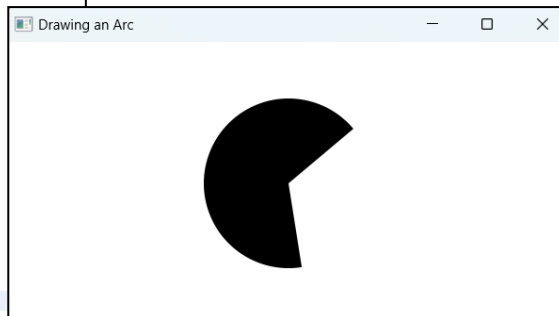
        arc.setType(ArcType.ROUND);

        Group root = new Group(arc);

        Scene scene = new Scene(root, 600, 300);
        stage.setTitle("Drawing an Arc");
        stage.setScene(scene);
        stage.show();
    }

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

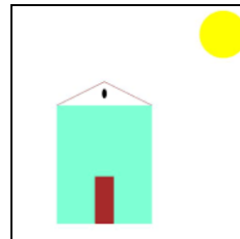
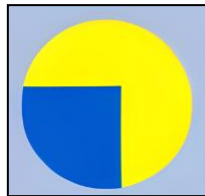
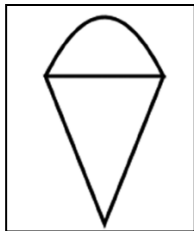
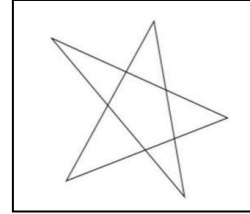
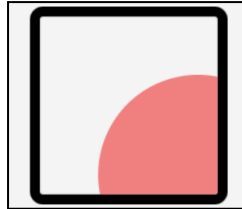
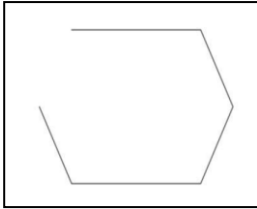
```





## امثلة للتجربة

► حاول كتابة التعليمات البرمجية اللازمة لإظهار كل من الواجهات التالية:



► 27

## نهاية المحاضرة



► 28