2D Shapes

The following table gives you the list of various shapes (classes) provided by JavaFX.

S.No	Shape & Description
1	**Line** : is a geometrical structure joining two point. The **Line** class of the package **javafx.scene.shape** represents a line in the XY plane.
2	**Rectangle** : is a four-sided polygon that has two pairs of parallel and concurrent sides with all interior angles as right angles. In JavaFX, a Rectangle is represented by a class named **Rectangle**. This class belongs to the package **javafx.scene.shape**.
3	**Rounded Rectangle** : you can draw a rectangle either with sharp edges or with arched edges and The one with arched edges is known as a rounded rectangle.
4	**Circle** : is a line forming a closed loop, every point on which is a fixed distance from a centre point. In JavaFX, a circle is represented by a class named **Circle**. This class belongs to the package **javafx.scene.shape**.
5	**Ellipse** : is defined by two points, each called a focus. If any point on the ellipse is taken, the sum of the distances to the focus points is constant. The size of the ellipse is determined by the sum of these two distances. In JavaFX, an ellipse is represented by a class named **Ellipse**. This class belongs to the package **javafx.scene.shape**.
6	**Polygon** : A closed shape formed by a number of coplanar line segments connected end to end. In JavaFX, a polygon is represented by a class named **Polygon**. This class belongs to the package **javafx.scene.shape**.
7	**Polyline** : is same a polygon except that a polyline is not closed in the end. Or, continuous line composed of one or more line segments. In JavaFX, a Polyline is represented by a class named **Polygon**. This class belongs to the package **javafx.scene.shape**.
8	**Cubic Curve** : is a Bezier parametric curve in the XY plane is a curve of degree 3. In JavaFX, a Cubic Curve is represented by a class named **CubicCurve**. This class belongs to the package **javafx.scene.shape**.
9	**QuadCurve** : is a Bezier parametric curve in the XY plane is a curve of degree 2. In JavaFX, a QuadCurve is represented by a class named QuadCurve. This class belongs to the package **javafx.scene.shape**.
10	**Arc** : is part of a curve. In JavaFX, an arc is represented by a class named **Arc**. This class belongs to the package − **javafx.scene.shape**.

	Types Of Arc In addition to this we can draw three types of arc's **Open, Chord, Round**.
11	**SVGPath** : In JavaFX, we can construct images by parsing SVG paths. Such shapes are represented by the class named **SVGPath**. This class belongs to the package **javafx.scene.shape**. This class has a property named **content** of String datatype. This represents the SVG Path encoded string, from which the image should be drawn.

Draw Line

A line is a geometrical structure which joins two points on an plane. In JavaFX, a line is represented by a class named **Line**. This class belongs to the package **javafx.scene.shape**.

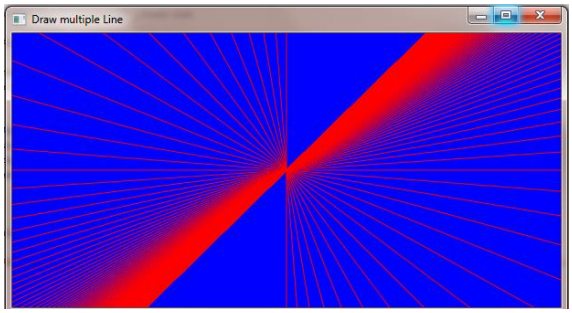
This class has 4 properties of the double data type namely −
startX − The x coordinate of the start point of the line.
startY − The y coordinate of the start point of the line.
endX − The x coordinate of the end point of the line.
endY − The y coordinate of the end point of the line.
To draw a line, you need to pass values to these properties, either by passing them to the constructor of this class, in the same order, at the time of instantiation, as follows −
Line line = new Line(startX, startY, endX, endY);
Or, by using their respective setter methods as follows −
setStartX(value);
setStartY(value);
setEndX(value);
setEndY(value);

Example
Following is the program which generates a straight line using JavaFX.

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

public class DrawingLine extends Application{
  @Override
  public void start(Stage stage) {
    //Creating a line object
    Line line = new Line();
```

```
        //Setting the properties to a line
        line.setStartX(100.0);
        line.setStartY(150.0);
        line.setEndX(500.0);
        line.setEndY(150.0);

        //Creating a Group
        Group root = new Group(line);

        //Creating a Scene
        Scene scene = new Scene(root, 600, 300);

        //Setting title to the scene
        stage.setTitle("Sample application");

        //Adding the scene to the stage
        stage.setScene(scene);

        //Displaying the contents of a scene
        stage.show();
    }
    public static void main(String args[]){
        launch(args);
    }
}
```

we can write the following program to draw multiple line as in figure bellow ,




```
package linespp;

import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.layout.StackPane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Line;

public class Linespp extends Application {

    @Override
    public void start(Stage ps) {

        //Creating a line object
        Line line = new Line();
```

```
        //Setting the properties to a line
        line.setStartX(100.0);
        line.setStartY(150.0);
        line.setEndX(500.0);
        line.setEndY(150.0);

        //Creating a Group
        Group root = new Group(line);

        StackPane sp = new StackPane();
        //Creating a line object
        for(double i= 1; i <= 720; i++){
          Line line = new Line(100, 100*i,100*i,1500);
          line.setStroke(Color.RED);
          sp.getChildren().add(line);
        }

        Scene scene = new Scene(sp, 600, 300, Color.BLUE);

        //Setting title to the scene
        ps.setTitle("Draw multiple Line");

        //Adding the scene to the stage
        ps.setScene(scene);

        //Displaying the contents of a scene
        ps.show();
    }
    public static void main(String args[]){
        launch(args);
    }
}
```

- ***Line properties example***

```
package javafxapplication20;

import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.scene.control.Slider;
import javafx.scene.shape.*;
```

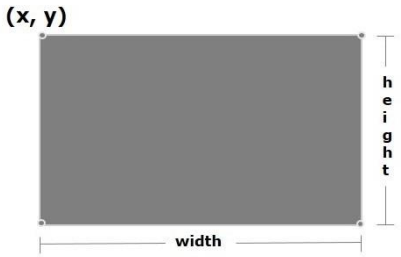
```
import javafx.scene.shape.StrokeLineCap;
import javafx.scene.text.Text;
import javafx.scene.paint.Color;

public class JavaFXApplication20 extends Application {

    @Override
    public void start(Stage stage) {

      Pane root = new Pane();
      //Creating a Red line
      Line redLine = new Line(10, 10, 200, 10);
      // setting common properties
      redLine.setStroke(Color.RED);
      redLine.setStrokeWidth(10);
      redLine.setStrokeLineCap(StrokeLineCap.BUTT);

    // creating a dashed pattern
    redLine.getStrokeDashArray().addAll(22d, 5d);
    redLine.setStrokeDashOffset(1);
    root.getChildren().add(redLine);

    // Green line
    Line greenLine = new Line(10, 30, 200, 30);
    greenLine.setStroke(Color.GREEN);
    greenLine.setStrokeWidth(10);
    greenLine.setStrokeLineCap(StrokeLineCap.ROUND);
    root.getChildren().add(greenLine);

    // Blue line
    Line blueLine = new Line(10, 50, 200, 50);
    blueLine.setStroke(Color.BLUE);
    blueLine.setStrokeWidth(20);
    root.getChildren().add(blueLine);
    blueLine.smoothProperty().set(true);

    // slider min, max, and current value
    Slider slider = new Slider(0, 200, 0);
    slider.setLayoutX(10);
    slider.setLayoutY(95);

    // bind the stroke dash offset property
    redLine.strokeDashOffsetProperty().bind(slider.valueProperty());
    root.getChildren().add(slider);
    Text offsetText = new Text("Stroke Dash Offset: 0.0");
```

```
    offsetText.setX(10);
    offsetText.setY(80);
    offsetText.setStroke(Color.PURPLE);

    // display stroke dash offset value
    root.getChildren().add(offsetText);

    Scene scene = new Scene(root, 600, 300, Color.AZURE);

      //Setting title to the scene
      stage.setTitle("Lines ");

      //Adding the scene to the stage
      stage.setScene(scene);

      //Displaying the contents of a scene
      stage.show();
   }
    public static void main(String[] args) {
        launch(args);
    }
}
```

- ***H.W how to draw vertical lines, or , horizontal lines.***

- ***Drawing Rectangle***

In general, a rectangle is a four-sided polygon that has two pairs of parallel and concurrent sides with all interior right angles.
In JavaFX, a Rectangle is represented by a class named **Rectangle**. This class belongs to the package **javafx.scene.shape**. This class has 4 properties of the double datatype namely –



X − The x coordinate of the start point (upper left) of the rectangle.
Y − The y coordinate of the start point (upper left) of the rectangle.
Width − The width of the rectangle.
height − The height of the rectangle.

To draw a rectangle, you need to pass values to these properties, either by passing them to the constructor of this class, in the same order, at the time of instantiation, as shown below –

Rectangle rectangle = new Rectangle(x, y, width, height);

Or, by using their respective setter methods as shown in the following code block −

```
setX(value);
setY(value);
setWidth(value);
setHeight(value);
```

In addition to common attributes, the Rectangle class also implements an *arc width* and *arc height* . This feature will draw rounded corners on a rectangle.

Following is a code snippet that draws a rounded rectangle positioned at (10, 10) with a width of 100, a height of 130, an arc width of 10, and an arc height of 40.

```
package javafxapplication22;

import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.scene.control.Slider;
import javafx.scene.shape.*;
import javafx.scene.shape.StrokeLineCap;
import javafx.scene.text.Text;
import javafx.scene.paint.Color;
```


```
public class JavaFXApplication22 extends Application {

    @Override
    public void start(Stage stage) {

    Pane root = new Pane();
    Rectangle roundRect = new Rectangle();
    roundRect.setX(10);
    roundRect.setY(10);
    roundRect.setWidth(100);
```

```
        roundRect.setHeight(130);
        roundRect.setArcWidth(10);
        roundRect.setArcHeight(40);
        roundRect.getStrokeDashArray().addAll(20d, 10d);
        roundRect.strokeWidthProperty().set(6);
        roundRect.setStroke(Color.BLACK);
        roundRect.setFill(Color.PINK);
        root.getChildren().add(roundRect);

        Scene scene = new Scene(root, 600, 300, Color.AZURE);

          //Setting title to the scene
          stage.setTitle("Draw Rectangle ");

          //Adding the scene to the stage
          stage.setScene(scene);

          //Displaying the contents of a scene
          stage.show();
      }
        public static void main(String[] args) {
           launch(args);
        }

}
```