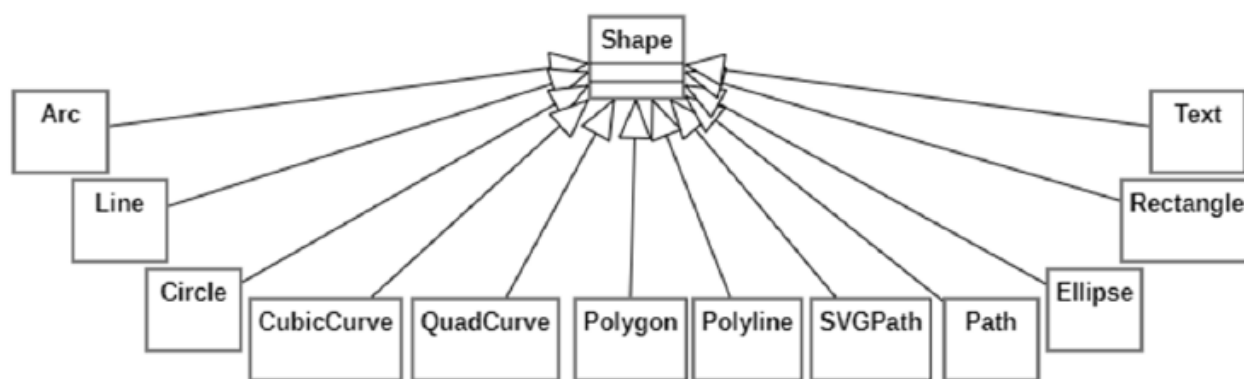


## Figuras 2D

Una figura en dos dimensiones es cualquier forma que se pueda dibujar en un plano bidimensional. JavaFX ofrece una variedad de nodos para dibujar diferentes figuras bidimensionales. Todas las clases de figuras 2D forman parte del paquete `javafx.scene.shape`. En la figura siguiente se observan las diferentes figuras 2D que soporta JavaFX, las cuales son subclases de *Shape*.



Jerarquía de clases de figuras 2D

Un objeto *Shape* tiene un tamaño y una posición, que se definen por sus propiedades, las cuales varían de acuerdo con el tipo de forma. Los objetos *Shape* tienen un interior y una línea de contorno (“stroke”). La propiedad *fill* especifica el color para rellenar el interior de la forma. El valor por defecto del relleno es de color negro (*BLACK*). La propiedad *stroke* especifica el color de la línea de contorno, que es *null* de forma predeterminada, excepto para *Line*, *Polyline* y *Path*, que tienen *Color.BLACK* como predeterminado.

A continuación, se presenta un resumen de las principales propiedades y métodos de las subclases de *Shape*.

Clase	Método	Descripción
<b>Shape</b>	<code>void setStroke(Paint valor)</code>	Establece la línea de contorno.
	<code>void setStrokeWidth(double valor)</code>	Establece la anchura de la línea de contorno.
	<code>setFill(Paint valor)</code>	Establece el relleno de la forma.
<b>Arc</b>	<code>Arc(double centroX, double centroY, double radioX, double radioY, double ánguloInicio, double longitud)</code>	Crea un arco con las propiedades pasadas como parámetros.
	<code>void setFill(Paint valor)</code>	Establece el relleno del arco.
	<code>setType(ArcType valor)</code>	Establece el tipo de arco.
<b>Line</b>	<code>Line(double inicioX, double inicioY, double finX, double finY)</code>	Crea una línea con las propiedades pasadas como parámetros.
<b>Circle</b>	<code>Circle(double radio)</code>	Crea un círculo con un radio determinado.

Clase	Método	Descripción
<b>CubicCurve</b>	<i>CubicCurve(double inicioX, double inicioY, double controlX1, double controlY1, double controlX2, double controlY2, double finX, double finY)</i>	Crea una curva paramétrica de Bezier de grado 3.
<b>QuadCurve</b>	<i>QuadCurve(double inicioX, double inicioY, double controlX1, double controlY1, double controlX2, double controlY2, double finX, double finY)</i>	Crea una curva cuadrática de grado 2.
<b>Polygon</b>	<i>Polygon()</i>	Crea un objeto polígono.
	<i>getPoints().addAll(double ... puntos)</i>	Establece las coordenadas de un polígono.
<b>PolyLine</b>	<i>Polyline(double ... puntos)</i>	Crea un objeto similar al polígono, pero no es una forma cerrada.
<b>SVGPath</b>	<i>SVGPath()</i>	Crea una forma simple que se construye analizando datos de una ruta SVG desde un <i>String</i> .
	<i>void setContent(String valor)</i>	Establece el contenido como un conjunto de coordenadas SVG.
<b>Path</b>	<i>Path()</i>	Crea una forma simple y proporciona las facilidades para la gestión de una ruta geométrica.
<b>Ellipse</b>	<i>Ellipse(double centroX, double centroY, double radioX, double radioY)</i>	Crea una elipse.
<b>Rectangle</b>	<i>Rectangle(double anchura, double altura)</i>	Crea un rectángulo con un tamaño dado.
	<i>void setArcWidth(double valor)</i>	Establece el diámetro vertical de los arcos del rectángulo.
	<i>void setArcHeight(double valor)</i>	Establece el diámetro horizontal del arco del rectángulo.
<b>Text</b>	<i>Text(double x, double y, String texto)</i>	Crea un texto en cierta posición.

## Objetivos de aprendizaje

Al finalizar este ejercicio, tendrás la capacidad para:

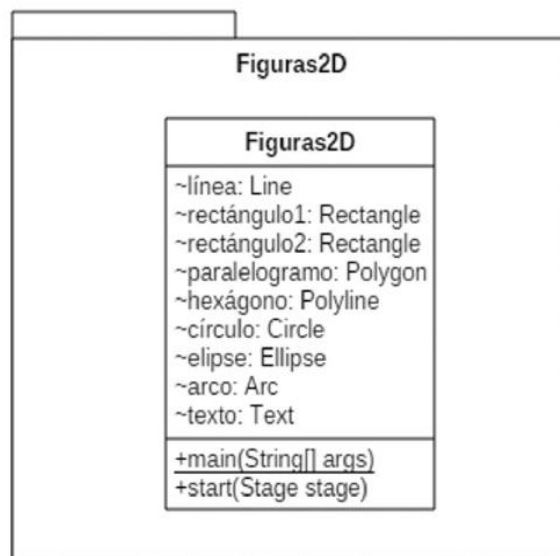
- Dibujar y presentar en ventanas gráficas diferentes figuras geométricas en 2D.
- Aplicar diferentes métodos para configurar la presentación visual de las figuras geométricas en 2D.

## Enunciado: Figuras 2D

Crear una ventana gráfica con las siguientes figuras geométricas 2D (Sharan, 2015):

- Una línea amarilla con ancho de línea 4.0 px de (0,0) a (70, 70).
- Un rectángulo verde de anchura 100 px y altura 75 px.
- Un rectángulo de color agua de anchura 100 px y altura 75 px con bordes redondeados.
- Un paralelogramo rojo con borde negro con vértices en (30, 0), (130, 0), (100, 75) y (0, 75).
- Un hexágono gris con borde negro y con vértices en (100, 0), (120, 25), (120, 50), (100, 75), (80, 50), (80, 25) y (100, 0).
- Un círculo azul con borde negro de ancho 3.0 px.
- Una elipse amarilla con borde azul de ancho 3.0 px.
- Un arco cerrado sin relleno y con borde naranja.
- Un texto con el valor "JavaFX".

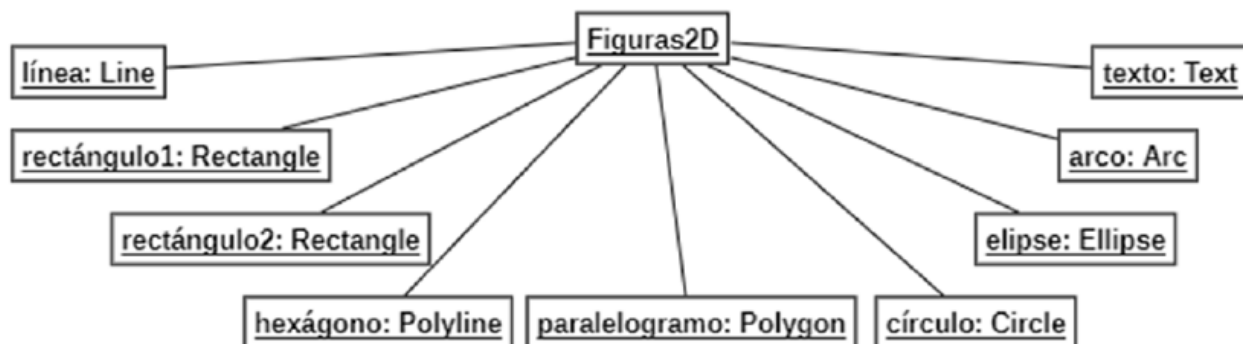
## Diagrama de clases



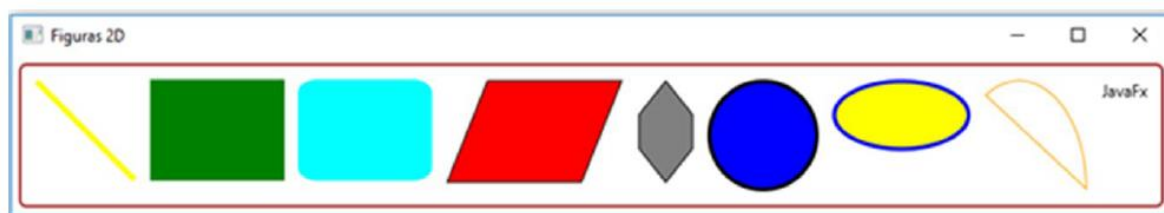
## Explicación del diagrama de clases

El diagrama de clases muestra un paquete denominado "Figuras2D", el cual contiene una sola clase denominada también "Figuras2D". Esta clase es una ventana gráfica que tiene atributos para indicar sus diferentes formas gráficas: una línea (*Line*), dos rectángulos (*Rectangle*), un paralelogramo (*Polygon*), un hexágono (*Polyline*), un círculo (*Circle*), una elipse (*Ellipse*), un arco (*Arc*) y un texto (*Text*). Los atributos tienen visibilidad de paquete indicada con el símbolo ~. La clase cuenta con dos métodos: *start* que inicia la aplicación gráfica y el método *main* (punto de entrada a la aplicación). El método *main* es un método estático, por lo cual se representa con su texto subrayado. Los dos métodos son públicos, lo cual se indica con el símbolo +.

## Diagrama de objetos



## Ejecución del programa



## Ejercicios propuestos

- Agregar al programa del ejercicio anterior las siguientes figuras 2D:
  - Una curva cuadrática.
  - Una curva cúbica.
- Realizar una ventana gráfica con las siguientes figuras 2D y colocarlas en el formato presentado:



## Instrucciones

Instrucción	Método	Descripción
<b>Stage</b>	<i>Stage(Parent root, double anchura, double altura)</i>	Crea un objeto <i>Scene</i> para un nodo específico con un tamaño específico.
	<i>void setScene(Scene valor)</i>	Especifica la escena utilizada en este escenario.
	<i>void setTitle(String valor)</i>	Establece el título del escenario.
	<i>void sizeToScene()</i>	Establece el tamaño del escenario.
	<i>void show()</i>	Muestra la ventana.
<b>HBox</b>	<i>HBox()</i>	Constructor de HBox.
	<i>void setSpacing(double valor)</i>	Establece un valor de espaciado.
	<i>void setStyle(String valor)</i>	Establece un <i>String</i> con CSS asociado.