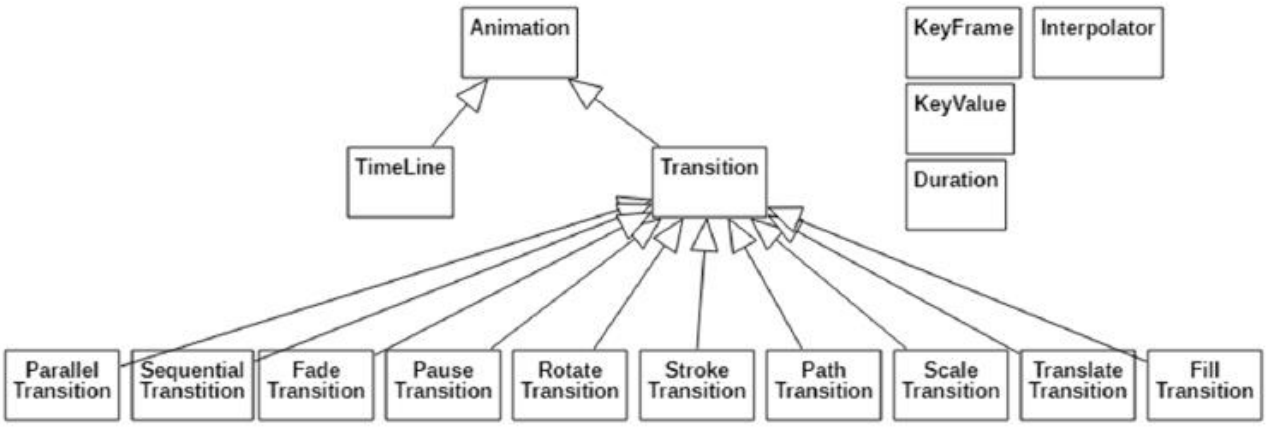
Animaciones

La animación implica algún tipo de movimiento, que se genera al mostrar imágenes en rápida sucesión. En JavaFX, la animación se define como el cambio de la propiedad de un nodo a lo largo del tiempo. Si la propiedad que cambia determina la ubicación del nodo, la animación en JavaFX producirá una ilusión de movimiento.

Las clases que proporcionan animación en JavaFX están en el paquete `javafx.animation`, excepto la clase `Duration`, que está en el paquete `javafx.util`. La figura muestra un diagrama de clase con la mayoría de las clases relacionadas con la animación.



Jerarquía de clases de animaciones

En la tabla se describen brevemente las principales clases utilizadas en JavaFX para realizar animaciones de figuras.

Clase	Descripción
Animation	Proporciona la funcionalidad básica a todas las animaciones en JavaFx.
Timeline	Una animación es caracterizada por sus propiedades asociadas, como tamaño, ubicación, color, etc. *Timeline* proporciona la capacidad de actualizar los valores de las propiedades a lo largo del tiempo.
Transition	Proporciona los medios para incorporar animaciones en una línea de tiempo interna.
ParallelTransition	Una transición paralela ejecuta varias transiciones simultáneamente.
SequentialTransition	Una transición secuencial ejecuta varias transiciones una después de otra.
FadeTransition	Esta transición crea una animación de efecto de opacidad que abarca su duración.
PauseTransition	Esta transición es utilizada para pausar entre múltiples animaciones aplicadas a un nodo de manera secuencial.
RotateTranstion	Crea una animación de rotación que abarca su duración.
StrokeTransition	Realiza una animación del color del borde del nodo para que pueda fluctuar entre dos valores de color durante la duración especificada.
PathTransition	Mueve un nodo a lo largo de cierto recorrido especificado de un extremo al otro durante un tiempo determinado.

Clase	Descripción
ScaleTransition	Esta transición realiza una animación de la escala del nodo durante la duración especificada por un factor determinado en cualquiera o en todas las tres direcciones *x*, *y*, *z*.
TranslateTransition	Traduce el nodo de una posición a otra durante la duración especificada.
FillTransition	Esta transición crea una animación que cambia el relleno de una forma a lo largo de una duración.
Duration	Define una duración de tiempo. La duración se puede crear utilizando el constructor o uno de los métodos de construcción estáticos, como segundos (*double*) o minutos (*double*).
KeyValue	Define un valor clave para ser interpolado en un intervalo particular a lo largo de la animación.
KeyFrame	Define valores objetivo en un punto específico en el tiempo para un conjunto de variables que se interpolan a lo largo de un *Timeline*.

Objetivos de aprendizaje

Al finalizar este ejercicio, el tendrás la capacidad para:

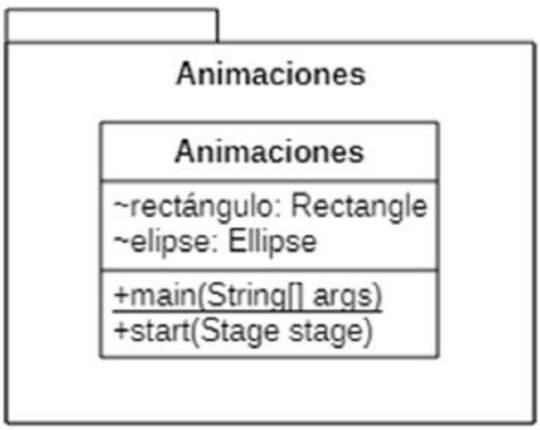
- Comprender qué es y cómo se realizan animaciones en JavaFX.
- Conocer las clases en JavaFX que se utilizan para realizar animaciones.
- Aplicar la clase `RotationTransition` para realizar animaciones de rotación de figuras.

Enunciado: Animaciones

Se requiere una ventana gráfica que muestre las siguientes figuras con sus animaciones respectivas:

- Un cuadrado de tamaño 100x100 que rote 2 segundos en un sentido y luego rote en forma contraria.
- Una elipse amarilla ubicada en (50, 50) y con radioX = 50 y radioY = 25, con línea de borde azul de 3.0 px y que tenga un cambio de opacidad de 2 segundos.

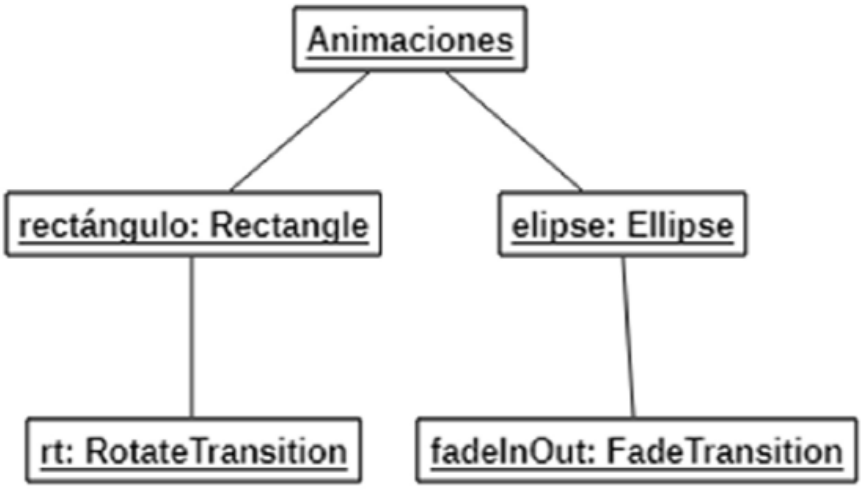
Diagrama de clases



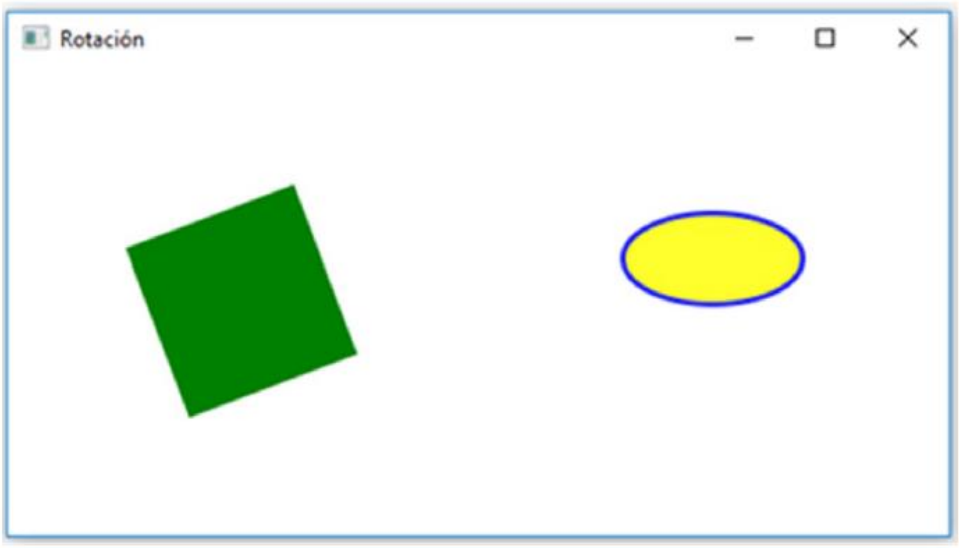
Explicación del diagrama de clases

El diagrama de clases muestra un paquete denominado “Animaciones”, este contiene una sola clase denominada también “Animaciones”. Esta clase es una ventana gráfica que tiene atributos para indicar dos figuras: un rectángulo (`Rectangle`) y una elipse (`Ellipse`), a estas se les aplicarán varios efectos de animación. 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

- Realizar un programa que muestre en la parte superior de una ventana gráfica el texto “Programación Orientada a Objetos” y lo mueva de izquierda a derecha hasta llegar al borde derecho y luego lo traslade de derecha a izquierda. Utilice la clase *TimeLine*.
- Realizar un programa que muestre en una ventana gráfica un cierto recorrido (*Path*) y permita que un círculo rojo con bordes negros de radio 10 se mueva por ese recorrido en cinco segundos.

Instrucciones

Clase	Método	Descripción
Rectangle	*Rectangle(double anchura, double altura, Paint relleno)*	Crea un rectángulo con un tamaño dado y un color de relleno.
Ellipse	*Ellipse(double centroX, double centroY, double radioX, double radioY)*	Crea una elipse.
	void setStroke(Paint valor)	Establece la línea de contorno.
	void setStrokeWidth(double valor)	Establece la anchura de la línea de contorno.
	setFill(Paint valor)	Establece el relleno de la forma.
RotateTransition	*RotateTransition(Duration duración, Node nodo)*	Crea una transición de rotación con una duración determinada para un nodo específico.
	void setFromAngle(double valor)	Especifica el ángulo inicial de rotación de la figura.
	void setToAngle(double valor)	Especifica el ángulo de detención de rotación de la figura.
	void setCycleCount(int valor)	Define el número de ciclos de la rotación.
	void setAutoReverse(boolean valor)	Define si la rotación invierte su dirección en ciclos alternos.
	void play()	Reproduce la rotación desde la posición actual en la dirección indicada por la velocidad.
FadeTransition	*FadeTransition(Duration duración, Node nodo)*	Crea una transición de opacidad con una duración para un nodo específico.
	void setFromValue(double valor)	Especifica el valor de inicio de la opacidad.
	void setToValue(double valor)	Especifica el valor de detención de la opacidad.
	void setCycleCount(int valor)	Define el número de ciclos de opacidad.
	void setAutoReverse(boolean valor)	Define si la opacidad invierte su dirección en ciclos alternos.
	void play()	Reproduce la opacidad desde la posición actual en la dirección indicada por la velocidad.

Clase	Método	Descripción
HBox	*HBox()*	Construye un HBox que permite presentar elementos horizontalmente.
	void setMargin(Node nodo, Inset valor)	Establece el margen para un nodo.
Scene	*Scene(Parent root)*	Construye una escena para un nodo específico.
Stage	*void setScene(Scene valor)*	Especifica la escena utilizada en este escenario.
	void setTitle(String valor)	Establece el título del escenario.
	void sizeToScene()	Establece el tamaño del escenario.
	void show()	Muestra la ventana.