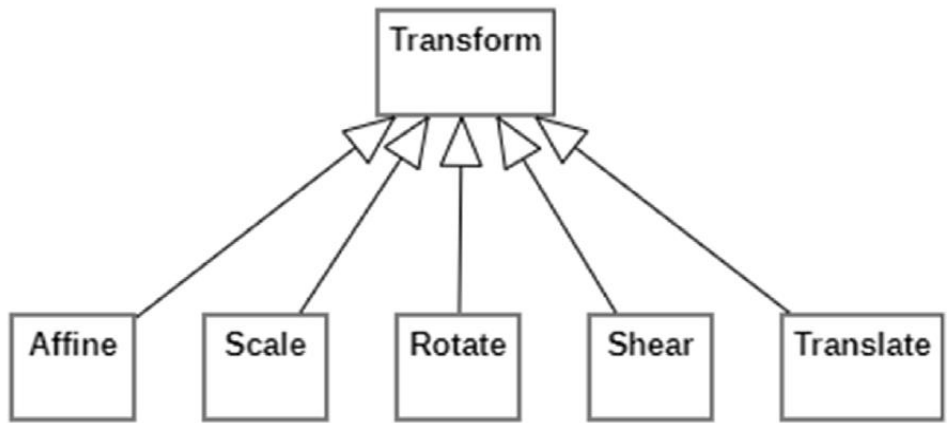


Transformaciones

Una transformación es un mapeo de puntos en un espacio de coordenadas para preservar distancias y direcciones entre ellos. Se pueden aplicar varios tipos de transformaciones a puntos en un espacio de coordenadas.

En la figura se muestran los diferentes tipos de transformación que permite JavaFX.



Jerarquía de clases de elementos de transformación

A continuación, en la tabla se describen brevemente las principales clases utilizadas para desarrollar transformaciones en formas de JavaFX.

Clase	Descripción
<i>Affine</i>	Esta transformación realiza un mapeo lineal desde coordenadas 2D/3D a otras coordenadas 2D/3D mientras preserva la rectitud y paralelismo de las líneas.
<i>Scale</i>	Representa una transformación que escala las coordenadas por los factores especificados.
<i>Rotate</i>	Esta transformación rota las coordenadas alrededor de un punto de anclaje.
<i>Shear</i>	Esta transformación corta coordenadas por los multiplicadores especificados.
<i>Translate</i>	Esta transformación traslada coordenadas por los factores especificados.

Objetivos de aprendizaje

Al finalizar este ejercicio, serás capaz de:

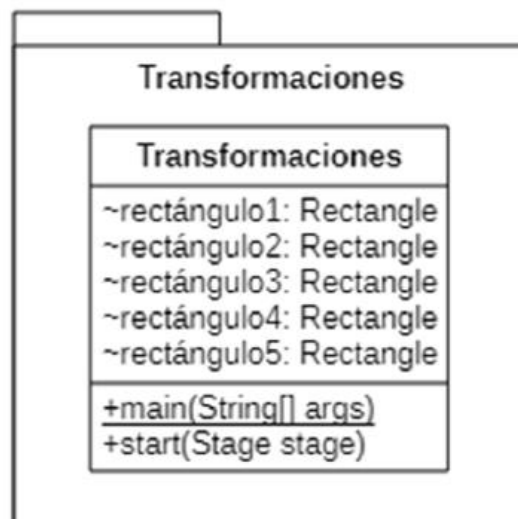
- Comprender qué es una transformación de figuras en JavaFX.
- Entender y aplicar las transformaciones de traslación, rotación, escala y corte.
- Aplicar diferentes transformaciones a componentes gráficos.

Enunciado: Transformaciones

Construir una ventana gráfica con cinco rectángulos:

- El primer rectángulo es de color rojo oscuro y tiene un tamaño de 100x50.
- El segundo rectángulo es de color índigo, tamaño de 100x50 y debe estar trasladado a la coordenada (100, 70).
- El tercer rectángulo es de color amarillo, tamaño de 100x50, debe estar ubicado en la coordenada (170, 60) y rotar 45 grados.
- El cuarto rectángulo es de color azul, tamaño de 100x50, debe estar localizado en la coordenada (250, 260) y se debe aumentar su escala un 30%.
- El quinto rectángulo es de color rojo, tamaño de 100x50, debe estar ubicado en la coordenada (360, 340) y se le debe aplicar un corte de 0.5 en ambos ejes.

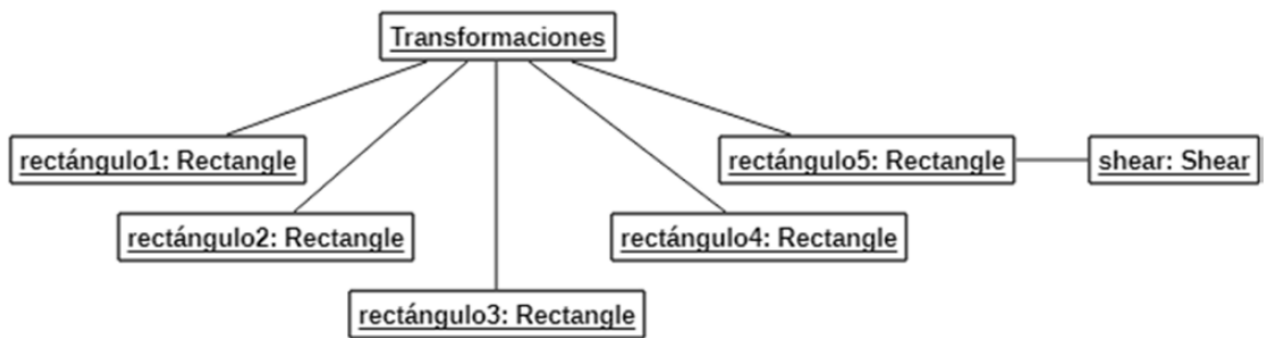
Diagrama de clases



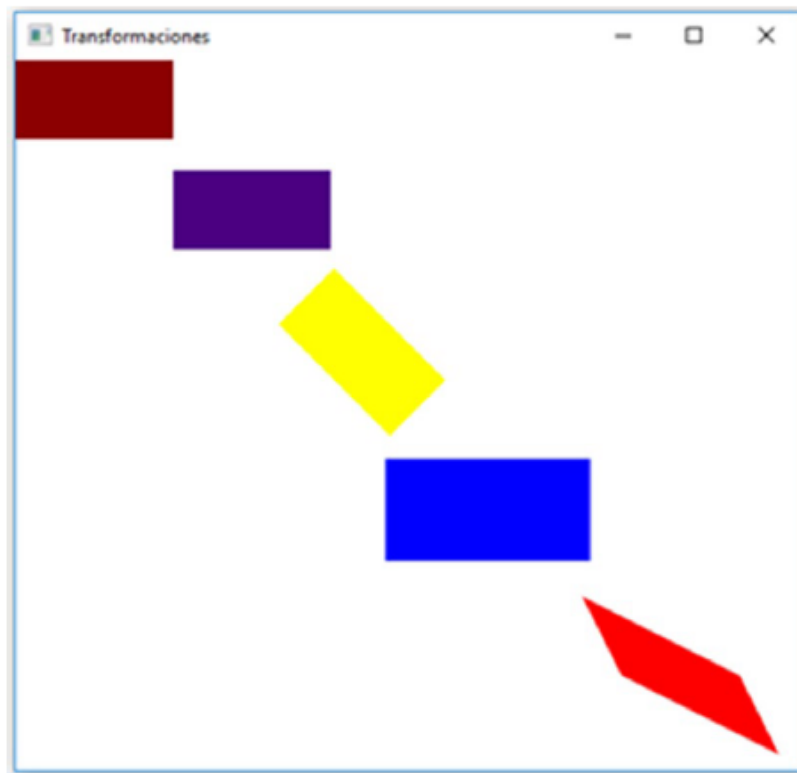
Explicación del diagrama de clases

El diagrama de clases muestra un paquete denominado “Transformaciones”, este contiene una sola clase denominada también “Transformaciones”. Esta clase es una ventana gráfica que tiene atributos para indicar diferentes rectángulos (clase *Rectangle*), a los cuales se les aplicarán varias transformaciones. 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

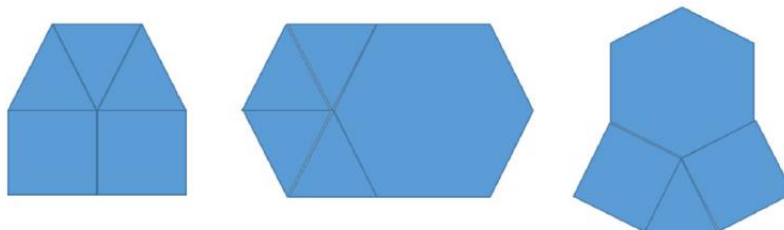


Ejecución del programa



Ejercicios propuestos

- Dibujar el teclado numérico y de operaciones aritméticas de una calculadora.
- Dibujar las siguientes figuras realizando transformaciones de JavaFX.



Instrucciones Java del ejercicio

Clase	Método	Descripción
Rectangle	<i>Rectangle(double anchura, double altura, Color value)</i>	Crea un rectángulo con un tamaño y color especificados.
	<i>setTranslateX(double valor)</i>	Traslada la figura a la coordenada x especificada.
	<i>setTranslateY(double valor)</i>	Traslada la figura a la coordenada y especificada.
	<i>setRotate(double valor)</i>	Rota la figura la cantidad de grados especificada.
	<i>setScaleX(double valor)</i>	Cambia la escala x de la figura el valor dado.
	<i>setScaleY(double valor)</i>	Cambia la escala y de la figura el valor dado.
	<i>getTransforms().addAll(Shear shear)</i>	Aplicar el recorte especificado a la figura.
Shear	<i>Shear(double valorX, double valorY)</i>	Crea un objeto <i>Shear</i> .
Group	<i>Group()</i>	Constructor de un contenedor con una lista de elementos que se representan en orden.
Stage	<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 show()</i>	Muestra la ventana.
Scene	<i>Scene(Parent root, double valorX, double valorY)</i>	Construye una escena para un nodo específico con el tamaño especificado.