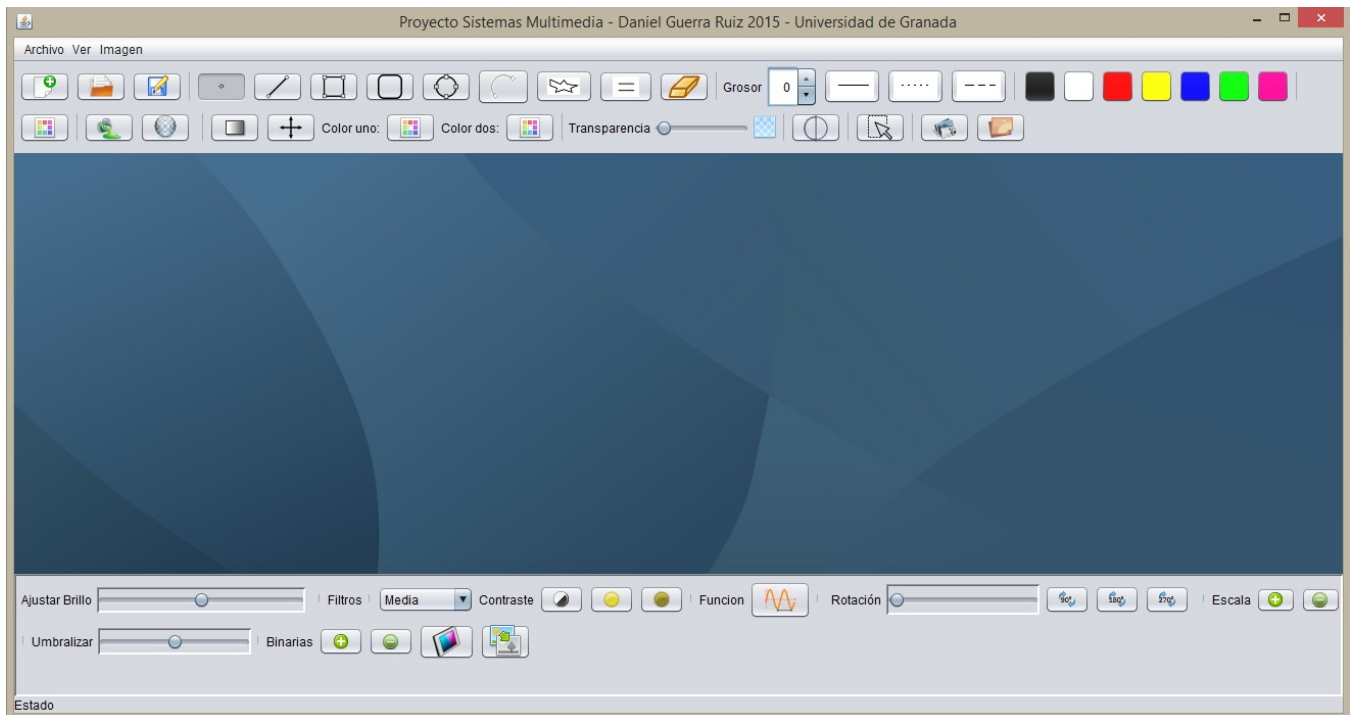


PROYECTO: SISTEMAS MULTIMEDIA

Daniel Guerra Ruiz - 2015



Biblioteca DGRShape.

Índice.

Descripción del problema y requisitos funcionales. Página 2

Diagrama de clases: Clase DGRShape, Linea2D y LineaCurva2D. Página 3

Clase MiLinea2D. Página 6

Clase MiRectanguloRedondeado2D. Página 7

Clase MiRectangulo2D. Página 8

Clase MiElipse2D. Página 10

Clase MiPoligonoEstrellado2D. Página 11

Clase MiLineaCurva2D. Página 13

Clase MiLineaParalela2D. Página 15

Clase Lienzo2D. Página 17

Clase Lienzo2DImagen. Página 18

Referencias. Página 18

1. DESCRIPCIÓN DEL PROBLEMA Y OBJETIVOS

Diseñar una API basada en Graphics2D de Java que permita dibujar elipses, rectángulos, rectángulos redondeados, líneas y puntos con propiedades independientes entre ellas. También incluirá un Lienzo2D donde se pintarán las figuras, y un Lienzo2DImagen donde se procesarán las imágenes que deseemos editar.

Las propiedades de las figuras son: color, relleno, transparencia, gradiente, textura, alisado y grosor.

El motivo de diseñar clases propias es que de otra manera, las clases ya predefinidas comparten los atributos de la interfaz Shape y tendrían todas las figuras las mismas propiedades de color, grosor y transparencia. Cuando se crean clases propias, podemos asignar valores únicos a cada figura evitando así el problema de atributos compartidos.

Cada clase tendrá un atributo de la figura correspondiente la cual se dibujará, más las propiedades adicionales anteriormente mencionadas. Así conseguimos la independencia de los valores antes compartidos por la interfaz Shape.

1.1. Descripción inicial del problema

Se parte de la necesidad de crear clases propias para cada figura, ya que como todas extienden de la clase Shape, comparten los atributos de color, grosor, transparencia y grosor.

1.2. Objetivos

Los objetivos principales que se deben cumplir deben ser los siguientes.

OBJ-1

Deberá proporcionar mecanismos de establecer propiedades a los atributos.

OBJ-2

Cada figura debe ser creada en una clase propia que incluya estos atributos

OBJ-3

Diseñar una clase abstracta que contiene las propiedades que comparten las figuras.

2. REQUISITOS FUNCIONALES

RF-1 .Gestion de figuras:

RF-1.1

Cada figura tendrá atributos de color, transparencia, relleno, grosor del trazo, gradiente, textura y alisado

RF-1.2

Se podrán editar las figuras en el modo edición y cambiar sus atributos, también podrán moverse por el lienzo.

RF-2. Gestión Imágenes:

RF-2.1

Se podrá dibujar figuras en las imagenes

RF-2.2

Se pueden copiar imágenes, obtener captura de un frame de video y operaciones de imágenes.

3. DIAGRAMA DE CLASES

Clase DGRShape, Linea2D y LineaCurva2D.



Metodos de DGRShape.

setMiFrame(): Asigna los valores del frame a la figura (Coordenadas (X,Y) , ancho y alto)

getBuffImg(): Devuelve la imagen del tipo BufferedImage, es la que se usa para hacer la textura.

setBuffimg(): Asigna la imagen tipo BufferedImage para hacer la textura.

getGradient(): Devuelve el gradiente tipo GradientPaint.

setGradient(): Asigna el GradientPaint para aplicar gradiente.

setColor(): Asigna el color a la figura.

getColor(): Devuelve el color de la figura.

setComposite(): Asigna transparencia a la figura.

getComposite(): Devuelve el objeto AlphaComposite de la figura.

getMiStroke(): Devuelve el grosor de la figura, de tipo Stroke.

setStroke(): Asigna el grosor de la figura.

getRender(): Devuelve el alisado de la figura, de tipo RenderingHints.

serRender(): Aplica el alisado a la figura.

contains(): Devuelve valor booleano si el punto contiene a una figura.

getX(): Devuelve la coordenada X de la figura.

getY(): Devuelve la coordenada Y de la figura.

getWidth(): Devuelve el ancho de la figura.
getHeight(): Devuelve la altura de la figura.

Metodos de Linea2D.

Linea2D(): Constructor sin parámetros
Linea2D(...) Constructor con parámetros.
isNear(): Devuelve booleano si el punto tiene cerca a un objeto de tipo Linea2D.Float.
contains(): Devuelve booleano si el punto contiene una Linea.
setLocation(): Asigna nueva posicion a la Linea
getStroke(): Devuelve el grosor de la Linea2D.Float.
setStroke(): Asigna el grosor a la linea.
getColor(): Devuelve el color de la linea.
setColor(): Asigna color a la linea.

Metodos de LineaCurva2D.

LineaCurva2D(): Constructor sin parámetros
Linea2CurvaD(...) Constructor con parámetros.
setLocation(): Asigna nueva posicion a la Linea
getStroke(): Devuelve el grosor de la Linea2D.Float.
setStroke(): Asigna el grosor a la linea.
getColor(): Devuelve el color de la linea.
setColor(): Asigna color a la linea.

Clase MiLinea2D.

MiLinea2D	
<ul style="list-style-type: none">- Linea2D Linea- boolean composite_bool- boolean renderinghints- boolean gradiente- boolean angulo- float gradual- int trazo- int tipotrazo- Color colorUno- Color colorDos- RenderingHints render- RenderingHints norender- AlphaComposite composite- AlphaComposite nocomposite- GradientPaint verticalGradient- GradientPaint gradient- BufferedImage bf	
<ul style="list-style-type: none">+ MiLinea2D(Point p1, Point p2, Color color, boolean relleno, boolean c, int st, int tipotrazo, boolean r, boolean grad, boolean ang, Color c1, Color c2, float gradual, BufferedImage bf)+ void setAtributos(Color color, boolean relleno, boolean c, int st, int tipotrazo, boolean r, boolean grad, boolean ang, Color c1, Color c2, float gradual, BufferedImage img)+ void draw(Graphics2D g2d)+ BufferedImage getBufferedImage()+ void setBufferedImage(BufferedImage buf)+ boolean isGradiente()+ void setGradiente(boolean gradiente)+ boolean isRenderinghints()+ void setRenderinghints(boolean r)+ boolean isComposite_bool()+ void setComposite_bool(boolean c)+ Linea2D getLinea()+ Color getColor()+ void setColor(Color c)+ Stroke getMiStroke()+ void setTrazo(Stroke trazo)+ void setLocation(Point2D pos)+ boolean isNear(Point2D p)+ boolean contains(Point2D p)+ Color getColorUno()+ void setColorUno(Color colorUno)+ Color getColorDos()+ void setColorDos(Color colorDos)+ boolean isAngulo()+ void setAngulo(boolean angulo)+ com.sun.java2d.geom.Shape impl_configShape()	

Metodos de MiLinea2D.

MiLinea2D(): Constructor con parámetros.

setAtributos(): Asigna atributos a la linea.

draw(): Pinta linea en lienzo.

getBufferedImage(): Devuelve la imagen que se usa para pintar la textura.

setBufferedImage(): Asigna la imagen de tipo BufferedImage para pintar textura

isGradiente(): Devuelve booleano si tiene activada la opcion de gradiente.

setGradiente(): Asigna valor booleano para activar el gradiente.

isRenderinghints(): Devuelve booleano si tiene activado el suavizado.

setRenderinghints(): Asigna valor de tipo boolean para aplicar el suavizado.

isComposite_bool(): Devuelve booleano si tiene activada la transparencia

setComposite_bool(): Asigna valor booleano para activar la transparencia.

getLinea(): Devuelve objeto Linea de tipo Linea2D.

getColor(): Devuelve el color de la figura.

setColor(): Asigna color a la figura.

GetColorUno(): Devuelve el primer color asignado de degradado.

GetColorDos(): Devuelve el segundo color asignado de degradado.

SetColorUno(): Asigna el primer color de degradado.

SetColorDos(): Asiga el segundo color de degradado.
 IsAngulo(): Devuelve el estado del angulo del degradado.
 SetAngulo(): Asigna el estado del angulo degradado, horizontal o vertical.
 setMiStroke(): Devuelve el grosor de la linea.
 setTrazo(): Asigna el grosor de la linea.
 setLocation(): Asigna nueva posicion a la linea.
 isNear(): Devuelve booleano si el punto esta cerca de la linea.
 contains(): Devuelve booleano si el punto contiene la linea.

Clase MiRectanguloRedondeado2D

MiRectanguloRedondeado2D	
<ul style="list-style-type: none"> - boolean relleno - boolean composite_bool - boolean renderinghints - boolean gradiente - boolean angulo - float gradual - int trazo - int tipotrazo - Color colorUno - Color colorDos - RenderingHints render - RenderingHints norender - AlphaComposite composite - AlphaComposite nocomposite - GradientPaint verticalGradient - GradientPaint gradient - RoundedRectangle2D.Double RectanguloRedondeado - final float dash1 - final BasicStroke dashed - BufferedImage bf 	
<ul style="list-style-type: none"> + MiRectanguloRedondeado2D(double x, double y, double ancho, double alto, Color color, boolean relleno, boolean c, int st, int tipotrazo, boolean r, boolean grad, boolean ang, Color c1, Color c2, float gradual, BufferedImage bf) + MiRectanguloRedondeado2D() + void setAtributos(Color color, boolean relleno, boolean c, int st, int tipotrazo, boolean r, boolean grad, boolean ang, Color c1, Color c2, float gradual, BufferedImage img) + RoundedRectangle2D.Double getRectanguloRedondeado() + void setRectanguloRedondeado(RoundedRectangle2D.Double RectanguloRedondeado) + BufferedImage getBufferedImage() + void setBufferedImage(BufferedImage buf) + boolean isGradiente() + void setGradiente(boolean gradiente) + boolean isRenderinghints() + void setRenderinghints(boolean renderinghints) + boolean isCompositeBoolean() + boolean isRelleno() + void setRelleno(boolean relleno) + AlphaComposite getTransparencia() + void setCompositeBoolean(boolean c) + void draw(Graphics2D g2d) + boolean contains(Point2D p) + Color getColorUno() + void setColorUno(Color colorUno) + Color getColorDos() + void setColorDos(Color colorDos) + boolean isAngulo() + void setAngulo(boolean angulo) + com.sun.javafx.geom.Shape impl_configShape() 	

Metodos de MiRectanguloRedondeado2D.

MiRectanguloRedondeado2D(...): Constructor con parámetros.
 MiRectanguloRedondeado2D(): Constructor sin parámetros.
 getRectanguloRedondeado(): Devuelve objeto tipo RoundedRectangle2D.Double.
 setRectanguloRedondeado(): Asigna valor de un RoundedRectangle2D.Double.
 setAtributos(): Asigna atributos a la linea.
 draw(): Pinta linea en lienzo.
 getBufferedImage(): Devuelve la imagen que se usa para pintar la textura.
 setBufferedImage(): Asigna la imagen de tipo BufferedImage para pintar textura
 isGradiente(): Devuelve booleano si tiene activada la opcion de gradiente.
 setGradiente(): Asigna valor booleano para activar el gradiente.
 isRenderinghints(): Devuelve booleano si tiene activado el suavizado.
 setRenderinghints(): Asigna valor de tipo boolean para aplicar el suavizado.
 isCompositeBoolean(): Devuelve booleano si tiene activada la transparencia
 setCompositeBoolean() Asigna valor booleano para activar la transparencia.

IsRelleno(): Devuelve booleano si tiene activado el relleno.
 SetRelleno(): Asigna booleano para activar relleno.
 setLocation(): Asigna nueva posicion a la linea.
 GetTransparencia(): devuelve el objeto AlphaComposite / transparencia
 contains(): Devuelve booleano si el punto contiene el rectangulo redondeado.
 getColor(): Devuelve el color de la figura.
 setColor(): Asigna color a la figura.
 GetColorUno(): Devuelve el primer color asignado de degradado.
 GetColorDos(): Devuelve el segundo color asignado de degradado.
 SetColorUno(): Asigna el primer color de degradado.
 SetColorDos(): Asiga el segundo color de degradado.
 IsAngulo(): Devuelve el estado del angulo del degradado.
 SetAngulo(): Asigna el estado del angulo degradado, horizontal o vertical.

Clase MiRectangulo2D.

MiRectangulo2D	
<ul style="list-style-type: none"> - boolean relleno - boolean composite_bool - boolean renderinghints - boolean gradiente - boolean angulo - float gradual - int trazo - int tipotrazo - Color colorUno - Color colorDos - RenderingHints render - RenderingHints norender - AlphaComposite composite - AlphaComposite nocomposite - GradientPaint verticalGradient - GradientPaint gradient - BufferedImage bf - Rectangle2D.Double Rectangulo 	
<ul style="list-style-type: none"> + MiRectangulo2D(double x, double y, double ancho, double alto, Color color, boolean relleno, boolean c, int st, int tipotrazo, boolean r, boolean grad, boolean ang, Color c1, Color c2, float gradual, BufferedImage bf) + MiRectangulo2D() + void setAtributos(Color color, boolean relleno, boolean c, int st, int tipotrazo, boolean r, boolean grad, boolean ang, Color c1, Color c2, float gradual, BufferedImage img) + BufferedImage getBufferedImage() + void setBufferedImage(BufferedImage buf) + boolean isGradiente() + void setGradiente(boolean gradiente) + boolean isRenderinghints() + void setRenderinghints(boolean renderinghints) + boolean isCompositeBoolean() + boolean isRelleno() + void setRelleno(boolean relleno) + AlphaComposite getTransparencia() + void setCompositeBoolean(boolean c) + Rectangle2D.Double getRectangulo() + void draw(Graphics2D g2d) + boolean contains(Point2D p) + com.sun.java2d.geom.Shape impl_configShape() + Color getColorUno() + void setColorUno(Color colorUno) + Color getColorDos() + void setColorDos(Color colorDos) + boolean isAngulo() + void setAngulo(boolean angulo) 	

Metodos de MiRectangulo2D.

MiRectangulo2D(...): Constructor con parámetros.
 getRectangulo(): Devuelve objeto tipo Rectangle2D.Double.
 setRectangulo(): Asigna valor de un Rectangle2D.Double.
 setAtributos(): Asigna atributos a la linea.
 draw(): Pinta linea en lienzo.
 getBufferedImage(): Devuelve la imagen que se usa para pintar la textura.
 setBufferedImage(): Asigna la imagen de tipo BufferedImage para pintar textura

isGradiente(): Devuelve booleano si tiene activada la opcion de gradiente.
 setGradiente(): Asigna valor booleano para activar el gradiente.
 isRenderinghints(): Devuelve booleano si tiene activado el suavizado.
 setRenderinghints(): Asigna valor de tipo boolean para aplicar el suavizado.
 isCompositeBoolean(): Devuelve booleano si tiene activada la transparencia.
 setCompositeBoolean() Asigna valor booleano para activar la transparencia.
 IsRelleno(): Devuelve booleano si tiene activado el relleno.
 SetRelleno(): Asigna booleano para activar relleno.
 setLocation(): Asigna nueva posicion a la linea.
 GetTransparencia(): devuelve el objeto AlphaComposite / transparencia
 contains(): Devuelve booleano si el punto contiene el rectangulo.
 getColor(): Devuelve el color de la figura.
 setColor(): Asigna color a la figura.
 GetColorUno(): Devuelve el primer color asignado de degradado.
 GetColorDos(): Devuelve el segundo color asignado de degradado.
 SetColorUno(): Asigna el primer color de degradado.
 SetColorDos(): Asiga el segundo color de degradado.
 IsAngulo(): Devuelve el estado del angulo del degradado.
 SetAngulo(): Asigna el estado del angulo degradado, horizontal o vertical.

Clase MiEllipse2D.

MiEllipse2D	
<ul style="list-style-type: none"> - boolean relleno - boolean composite_bool - boolean renderinghints - boolean gradiente - boolean angulo - float gradual - int trazo - int tipotrazo - Color colorUno - Color colorDos - RenderingHints render - RenderingHints norender - AlphaComposite composite - AlphaComposite nocomposite - GradientPaint verticalGradient - GradientPaint gradient - Ellipse2D.Double Ellipse - BufferedImage bf 	
<ul style="list-style-type: none"> + MiEllipse2D(double x, double y, double ancho, double alto, Color color, boolean relleno, boolean c, int st, int tipotrazo, boolean r, boolean grad, boolean ang, Color c1, Color c2, float gradual, BufferedImage bff) + void setAtributos(Color color, boolean relleno, boolean c, int st, int tipotrazo, boolean r, boolean grad, boolean ang, Color c1, Color c2, float gradual, BufferedImage img) + Color getColorUno() + void setColorUno(Color colorUno) + Color getColorDos() + void setColorDos(Color colorDos) + boolean isAngulo() + void setAngulo(boolean angulo) + BufferedImage getBufferedImage() + void setBufferedImage(BufferedImage buf) + boolean isGradiente() + void setGradiente(boolean g) + boolean isRenderinghints() + void setRenderinghints(boolean renderinghints) + boolean isCompositeBoolean() + boolean isRelleno() + void setRelleno(boolean relleno) + AlphaComposite getTransparencia() + void setCompositeBoolean(boolean c) + Ellipse2D.Double getEllipse() + void draw(Graphics2D g2d) + boolean contains(Point2D p) + Shape impl_configShape() 	

Metodos de MiEllipse2D.

MiEllipse2D(...): Constructor con parámetros.

getEllipse(): Devuelve objeto tipo Ellipse2D.Double.

setEllipse(): Asigna valor de un Ellipse2D.Double.

setAtributos(): Asigna atributos a la linea.

draw(): Pinta linea en lienzo.

getBufferedImage(): Devuelve la imagen que se usa para pintar la textura.

setBufferedImage(): Asigna la imagen de tipo BufferedImage para pintar textura

isGradiente(): Devuelve booleano si tiene activada la opcion de gradiente.

setGradiente(): Asigna valor booleano para activar el gradiente.

isRenderinghints(): Devuelve booleano si tiene activado el suavizado.

setRenderinghints(): Asigna valor de tipo boolean para aplicar el suavizado.

isCompositeBoolean(): Devuelve booleano si tiene activada la transparencia

setCompositeBoolean() Asigna valor booleano para activar la transparencia.

IsRelleno(): Devuelve booleano si tiene activado el relleno.

SetRelleno(): Asigna booleano para activar relleno.

setLocation(): Asigna nueva posicion a la linea.

GetTransparencia(): devuelve el objeto AlphaComposite / transparencia

contains(): Devuelve booleano si el punto contiene la elipse.

getColor(): Devuelve el color de la figura.

setColor(): Asigna color a la figura.

GetColorUno(): Devuelve el primer color asignado de degradado.

GetColorDos(): Devuelve el segundo color asignado de degradado.

SetColorUno(): Asigna el primer color de degradado.

SetColorDos(): Asiga el segundo color de degradado.

IsAngulo(): Devuelve el estado del angulo del degradado.

SetAngulo(): Asigna el estado del angulo degradado, horizontal o vertical.

Clase MiPoligonoEstrellado2D.

MiPoligonoEstrellado2D	
<ul style="list-style-type: none">- GeneralPath estrella- boolean relleno- boolean composite_bool- boolean renderinghints- boolean gradiente- boolean angulo- float gradual- int trazo- int tipotrazo- Color colorUno- Color colorDos- RenderingHints render- RenderingHints norender- AlphaComposite composite- AlphaComposite nocomposite- GradientPaint verticalGradient- GradientPaint gradient- BufferedImage bf	
<ul style="list-style-type: none">+ MiPoligonoEstrellado2D(double x, double y, double ancho, double alto, Color color, boolean relleno, boolean c, int st, int tipotrazo, boolean r, boolean grad, boolean ang, Color c1, Color c2, float gradual, BufferedImage bf)+ MiPoligonoEstrellado2D()+ Color getColorUno()+ void setColorUno(Color colorUno)+ Color getColorDos()+ void setColorDos(Color colorDos)+ boolean isAngulo()+ void setAngulo(boolean angulo)+ void setAtributos(Color color, boolean relleno, boolean c, int st, int tipotrazo, boolean r, boolean grad, boolean ang, Color c1, Color c2, float gradual, BufferedImage img)+ BufferedImage getBufferedImage()+ void setBufferedImage(BufferedImage buf)+ boolean isGradiente()+ void setGradiente(boolean gradiente)+ boolean isRenderinghints()+ void setRenderinghints(boolean renderinghints)+ boolean isCompositeBoolean()+ boolean isRelleno()+ void setRelleno(boolean relleno)+ AlphaComposite getTransparencia()+ void setCompositeBoolean(boolean c)+ GeneralPath getEstrella()+ void draw(Graphics2D g2d)+ boolean contains(Point2D p)+ void setPosicion(Point2D p)+ com.sun.java2d.geom.Shape impl_configShape()	

Metodos de MiPoligonoEstrellado2D.

MiPoligonoEstrellado2D(...): Constructor con parámetros.

MiPoligonoEstrellado2D(): Constructor sin parámetros.

getEstrella(): Devuelve objeto tipo GenralPath

setAtributos(): Asigna atributos a la linea.

draw(): Pinta linea en lienzo.

getBufferedImage(): Devuelve la imagen que se usa para pintar la textura.

setBufferedImage(): Asigna la imagen de tipo BufferedImage para pintar textura

isGradiente(): Devuelve booleano si tiene activada la opcion de gradiente.

setGradiente(): Asigna valor booleano para activar el gradiente.

isRenderinghints(): Devuelve booleano si tiene activado el suavizado.

setRenderinghints(): Asigna valor de tipo boolean para aplicar el suavizado.

isCompositeBoolean(): Devuelve booleano si tiene activada la transparencia

setCompositeBoolean() Asigna valor booleano para activar la transparencia.

IsRelleno(): Devuelve booleano si tiene activado el relleno.

SetRelleno(): Asigna booleano para activar relleno.

setPosicion(): Asigna nueva posicion a la figura.

GetTransparencia(): devuelve el objeto AlphaComposite / transparencia

contains(): Devuelve booleano si el punto contiene la elipse.

getColor(): Devuelve el color de la figura.

setColor(): Asigna color a la figura.

GetColorUno(): Devuelve el primer color asignado de degradado.

GetColorDos(): Devuelve el segundo color asignado de degradado.

SetColorUno(): Asigna el primer color de degradado.

SetColorDos(): Asiga el segundo color de degradado.

IsAngulo(): Devuelve el estado del angulo del degradado.

SetAngulo(): Asigna el estado del angulo degradado, horizontal o vertical.

Clase MiLineaCurva2D.

MiLineaCurva2D	
<ul style="list-style-type: none">- LineaCurva2D Linea- Point p3- boolean composite_bool- boolean renderinghints- boolean gradiente- boolean angulo- float gradual- int trazo- int tipotrazo- Color colorUno- Color colorDos- RenderingHints render- RenderingHints norender- AlphaComposite composite- AlphaComposite nocomposite- GradientPaint verticalGradient- GradientPaint gradient- BufferedImage bf	
<ul style="list-style-type: none">+ MiLineaCurva2D(Point p1, Point p2, Color color, boolean relleno, boolean c, int st, int tipotrazo, boolean r, boolean grad, boolean ang, Color c1, Color c2, float gradual, BufferedImage bff)+ void setCurva(Point p3)+ void setAtributos(Color color, boolean relleno, boolean c, int st, int tipotrazo, boolean r, boolean grad, boolean ang, Color c1, Color c2, float gradual, BufferedImage img)+ Color getColorUno()+ void setColorUno(Color colorUno)+ Color getColorDos()+ void setColorDos(Color colorDos)+ boolean isAngulo()+ void setAngulo(boolean angulo)+ void draw(Graphics2D g2d)+ BufferedImage getBufferedImage()+ void setBufferedImage(BufferedImage buf)+ boolean isGradiente()+ void setGradiente(boolean gradiente)+ boolean isRenderinghints()+ void setRenderinghints(boolean r)+ boolean isComposite_bool()+ void setComposite_bool(boolean c)+ LineaCurva2D getLinea()+ Color getColor()+ void setColor(Color c)+ Stroke getMiStroke()+ void setTrazo(Stroke trazo)+ void setLocation(Point2D pos)+ boolean contains(Point2D p)+ com.sun.java2d.geom.Shape impl_configShape()	

Metodos de MiLineaCurva2D.

MiLineaCurva2D(...): Constructor con parámetros.

getEstrella(): Devuelve objeto tipo GenralPath

setAtributos(): Asigna atributos a la linea.

draw(): Pinta linea en lienzo.

getBufferedImage(): Devuelve la imagen que se usa para pintar la textura.

setBufferedImage(): Asigna la imagen de tipo BufferedImage para pintar textura

isGradiente(): Devuelve booleano si tiene activada la opcion de gradiente.

setGradiente(): Asigna valor booleano para activar el gradiente.

isRenderinghints(): Devuelve booleano si tiene activado el suavizado.

setRenderinghints(): Asigna valor de tipo boolean para aplicar el suavizado.

isCompositeBoolean(): Devuelve booleano si tiene activada la transparencia

setCompositeBoolean() Asigna valor booleano para activar la transparencia.

IsRelleno(): Devuelve booleano si tiene activado el relleno.

SetRelleno(): Asigna booleano para activar relleno.

setPosicion(): Asigna nueva posicion a la figura.

GetTransparencia(): devuelve el objeto AlphaComposite / transparencia

contains(): Devuelve booleano si el punto contiene la elipse.

getColor(): Devuelve el color de la figura.

setColor(): Asigna color a la figura.

GetColorUno(): Devuelve el primer color asignado de degradado.

GetColorDos(): Devuelve el segundo color asignado de degradado.

SetColorUno(): Asigna el primer color de degradado.

SetColorDos(): Asiga el segundo color de degradado.

IsAngulo(): Devuelve el estado del angulo del degradado.

SetAngulo(): Asigna el estado del angulo degradado, horizontal o vertical.

Clase MiLineaParalela2D

MiLineaParalela2D
<ul style="list-style-type: none">- GeneralPath trazolibre- boolean relleno- ArrayList puntosX- ArrayList puntosY- boolean composite_bool- boolean renderinghints- boolean gradiente- boolean angulo- float gradual- int trazo- int tipotrazo- Point p1- Color colorUno- Color colorDos- RenderingHints render- RenderingHints norender- AlphaComposite composite- AlphaComposite nocomposite- GradientPaint verticalGradient- GradientPaint gradient- BufferedImage bf
<ul style="list-style-type: none">♦ MiLineaParalela2D(Point pt1, Point pt2, double ancho, double alto, Color color, boolean relleno, boolean c, int st, int tipotrazo, boolean r, boolean grad, boolean ang, Color c1, Color c2, float gradual, BufferedImage bff)♦ MiLineaParalela2D()+ void setAtributos(Color color, boolean relleno, boolean c, int st, int tipotrazo, boolean r, boolean grad, boolean ang, Color c1, Color c2, float gradual, BufferedImage img)+ BufferedImage getBufferedImage()+ void setBufferedImage(BufferedImage buf)+ boolean isGradiente()+ void setGradiente(boolean gradiente)+ boolean isRenderinghints()+ void setRenderinghints(boolean renderinghints)+ boolean isCompositeBoolean()+ boolean isRelleno()+ void setRelleno(boolean relleno)+ AlphaComposite getTransparencia()+ void setCompositeBoolean(boolean c)+ GeneralPath getTrazolibre()+ void draw(Graphics2D g2d)+ boolean contains(Point2D p)+ void setPosicion(Point2D p)+ com.sun.java2d.geom.Shape impl_configShape()+ Color getColorUno()+ void setColorUno(Color colorUno)+ Color getColorDos()+ void setColorDos(Color colorDos)+ boolean isAngulo()+ void setAngulo(boolean angulo)

Metodos de MiLineaParalela2D.

MiLineaParalela2D(...): Constructor con parámetros.

MiLineaParalela2D(): Constructor sin parámetros.

getEstrella(): Devuelve objeto tipo GenralPath

setAtributos(): Asigna atributos a la linea.

draw(): Pinta linea en lienzo.

getBufferedImage(): Devuelve la imagen que se usa para pintar la textura.

setBufferedImage(): Asigna la imagen de tipo BufferedImage para pintar textura

isGradiente(): Devuelve booleano si tiene activada la opcion de gradiente.

setGradiente(): Asigna valor booleano para activar el gradiente.

isRenderinghints(): Devuelve booleano si tiene activado el suavizado.

setRenderinghints(): Asigna valor de tipo boolean para aplicar el suavizado.

isCompositeBoolean(): Devuelve booleano si tiene activada la transparencia

setCompositeBoolean() Asigna valor booleano para activar la transparencia.

IsRelleno(): Devuelve booleano si tiene activado el relleno.

SetRelleno(): Asigna booleano para activar relleno.

setPosicion(): Asigna nueva posicion a la figura.

GetTransparencia(): devuelve el objeto AlphaComposite / transparencia

contains(): Devuelve booleano si el punto contiene la elipse.

getColor(): Devuelve el color de la figura.

setColor(): Asigna color a la figura.

GetColorUno(): Devuelve el primer color asignado de degradado.
GetColorDos(): Devuelve el segundo color asignado de degradado.
SetColorUno(): Asigna el primer color de degradado.
SetColorDos(): Asigna el segundo color de degradado.
IsAngulo(): Devuelve el estado del angulo del degradado.
SetAngulo(): Asigna el estado del angulo degradado, horizontal o vertical.

Clase Lienzo2D.

Lienzo2D	
<ul style="list-style-type: none">- Point p1- int caso- int altura- int anchura- Color color- Color colorUno- Color colorDos- boolean relleno- boolean transparencia- BufferedImage img- boolean gradiente- boolean angulo- float transparenciaGradual- int tipoTrazo- boolean alisar- boolean editar- boolean borrar- int formaActiva- Graphics2D g2d- float patronDiscontinuidad- int trazo- ArrayList<DGRShape> vShape- DGRShape s	
<ul style="list-style-type: none">+ Lienzo2D()+ Color getColorUno()+ void setColorUno(Color colorUno)+ Color getColorDos()+ void setColorDos(Color colorDos)+ int getTipoTrazo()+ void setTipoTrazo(int tipoTrazo)+ float getTransparenciaGradual()+ void setTransparenciaGradual(float t)+ boolean isAngulo()+ void setAngulo(boolean angulo)+ void setTrazo(int trazo)+ boolean isGradiente()+ void setGradiente(boolean gradiente)+ BufferedImage getImg()+ void setImg(BufferedImage img)+ void setBorrar(boolean b)+ void setFormaActiva(int caso)+ void setP1(Point p)+ void setP2(Point p)+ void setAlisar(boolean a)+ boolean getAlisar()+ void setEditar(boolean a)+ boolean getEditar()+ void setColor(Color c)+ Color getColor()+ void setRelleno(boolean r)+ boolean getRelleno()+ void setTransparencia(boolean t)+ boolean getTransparencia()+ void paint(Graphics g)+ DGRShape getSelectedShape(Point2D p)+ void setLocation(DGRShape s, Point2D posicion)+ DGRShape createShape(Point p1, Point p2, Color c, boolean relleno, boolean compo, int st, int tipotrazo, boolean r, boolean grad, boolean ang, Color c1, Color c2, float gradual, BufferedImage imagen)- // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-BEGIN: initComponents void initComponents()- void formMouseClicked(java.awt.event.MouseEvent evt)- void formMouseReleased(java.awt.event.MouseEvent evt)- void formMousePressed(java.awt.event.MouseEvent evt)- void formMouseDragged(java.awt.event.MouseEvent evt)	

Metodos Lienzo2D.

Lienzo2D(): constructor sin parámetros.

setStroke(): Asigna grosor del trazo al lienzo.

isGradiente(): Devuelve booleano si esta activado el gradiente.

setGradiente(): Asigna valor booleano si esta activado el gradiente.

getImg(): Devuelve el BufferedImage usado para la textura.









setImg(): Asigna BufferedImage para pintar con textura.

setBorrar(): Limpia el vector de vShape, con lo cual borra las figuras pintadas.

setFormaActiva(): Asigna el valor de la forma activa con la que se pintaran las distintas figuras.

setP1(): Asigna el valor al punto 1, normalmente de inicio de la figura a crear.
 setP2(): Asigna el valor al punto 2, normalmente de final de la figura a crear.
 setAlisar(): Asigna valor booleano si esta activada la funcion de alisar.
 getAlisar(): Devuelve valor booleano si esta activada la funcion alisar.
 setEditar(): Asigna valor booleano si esta activada la funcion edicion.
 getEditar(): Devuelve valor booleano de la funcion edicion.
 setColor(): Asigna color al lienzo, consecuentemente cuando se pinte una figura se le pasa este color.
 getColor(): Se obtiene el color activado del lienzo.
 setRelleno(): Se asigna valor booleano si esta activa la funcion de relleno.
 getRelleno(): Devuelve valor booleano de la funcion de relleno.
 setTransparencia(): Asigna valor booleano si esta activa la funcion de transparencia.
 getTransparencia(): Devuelve valor booleano si esta activa la funcion de transparencia.
 paint(): Metodo dedicado a pintar las figuras en el lienzo.
 getSelectedShape(): Devuelve el DGRShape si éste esta dentro del punto pasado por parámetro.
 setLocation(): Asigna la nueva posicion de la figura seleccionada pasada por parámetro.
 createShape(): Metodo encargado de crear las distintas figuras correspondientes con sus atributos propios.

Clase Lienzo2DImagen.

 Lienzo2DImagen	
 - BufferedImage bufImg	
 +Lienzo2DImagen()	
 +BufferedImage getImagen(boolean drawVector)	
 +void setImagen(BufferedImage bf)	
 +BufferedImage getImagen()	
 +void paintComponent(Graphics g)	
 -// <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-BEGIN: initComponents void initComponents()	

Metodos de Lienzo2DImagen.

Lienzo2DImagen(): Constructor sin parámetros.
 getImagen(...): Devuelve la Imagen con las figuras pintadas sobre el lienzo.
 setImagen(): Asigna al lienzo la imagen pasada por parámetro, de tipo BufferedImage.
 getImagen(): Devuelve la Imagen sin figuras pintadas.
 paintComponent(): Metodo encargado de pintar sobre la imagen BufferedImage las figuras de la clase DGRShape.

Referencias:

<https://joseluisdelarosa.wordpress.com/2007/05/15/%C2%BFcomo-documentar-una-api/>

http://www.aprenderaprogramar.es/index.php?option=com_content&view=article&id=646:documentar-proyectos-java-con-javadoc-comentarios-simbolos-tags-deprecated-param-etc-cu00680b&catid=68:curso-aprender-programacion-java-desde-cero&Itemid=188

