**Escuela Colombiana de Ingeniería Julio Garavito**

**LABORATORIO NUMERO. 01**

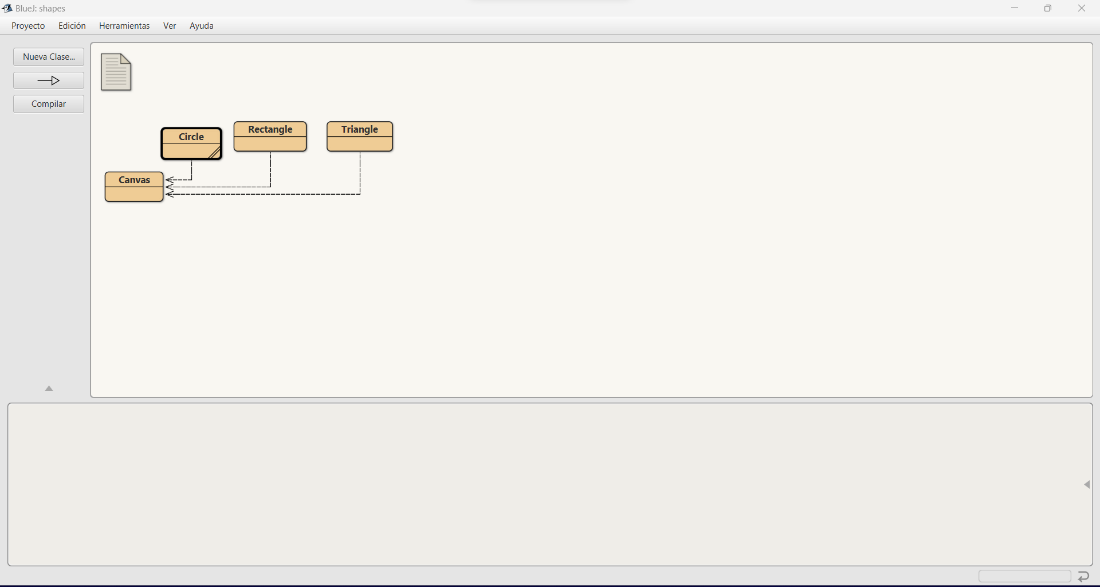
Nombres: Camilo Murcia Espinosa, Jeisson Casallas

25/8/2023

**SHAPES**

1. **CONOCIENDO EL PROYECTO SHAPES**

**1) El proyecto “shapes” es una versión modificada de un recurso ofrecido por BlueJ. Para trabajar con él, bajen shapes.zip y ábranlo en BlueJ. Capturen la pantalla.**



**2) El diagrama de clases permite visualizar las clases de un artefacto software y las relaciones entre ellas. Considerando el diagrama de clases de “shapes”**

**A.** **¿Qué clases ofrece?**

* Estan las clases de Canvas, Circle, Rectangle, Triangle

**B.** **¿Qué relaciones existen entre ellas?**

* Todas las clases se relacionan gracias a la clase de Canvas que es donde llegan a parar las demas clases, cuyo es fin es dibujar las anteriores clases.

**3) La documentación presenta las clases del proyecto y, en este caso, la especificación de sus componentes públicos. De acuerdo con la documentación generada:**

**A. ¿Qué clases tiene el paquete shapes?**

* Circle, Rectangle, Triangle, Canvas

**B. ¿Qué atributos tiene la clase Triangle?**

* La clase Triangle tiene como atributos a Vertices.

**C. ¿Cuántos métodos ofrece la clase Triangle?**

* La clase triangle ofrece los metodos triangle, makeVisible, makeInvisible, moveRight, moveLeft, moveUp, moveDown, moveHorizontal, moveVertical, slowMoveHorizontal, slowMoveVertical, changeSize, changeColor.

**D.)** **¿Cuáles métodos ofrece la clase Triangle para que la figura cambie su tamaño (incluya sólo el nombre)?**

* El metodo para cambiar el tamaño de la figura es changeSize.

**4) En el código de cada clase está el detalle de la implementación. Revisen el código de la clase Triangle. Con respecto a los atributos:**

**A. ¿Cuántos atributos realmente tiene?**

* En total hay 7 atributos

**B. ¿Cuáles atributos determinan su forma?**

* 2 atributos determinan su forma, que es height y width

**C. ¿Cuántos métodos tiene en total?**

* Tiene un total de 15 metodos (12 publicos, 2 privados y el constructor).

**D. ¿Quiénes usan los métodos privados?**

* Metodos privados solo tenemos a draw y a erase, y se usan por algunos de los metodos publicos.
* Draw: (makeVisible, moveHorizontal, moveVertical, slowMoveHorizontal, slowMoveVertical, changeSize, changeColor).
* Erase: (makeInvisible, changeSize, moveVertical, moveHorizontal).

**5. Comparando la documentación con el código**

**A. ¿Qué no se ve en la documentación?**

Los atributos y métodos que no tienen el modificador de acceso Public, es decir los que tienen en private

**B. ¿por qué debe ser así?**

* Los atributos y métodos privados no se ven en la documentación, debido a que solo se usan dentro de la clase y no es necesario que el usuario final tenga conocimiento de ello.

**6. En el código de la clase Triangle, revise el atributo VERTICES**

**A. ¿Qué significa que sea public?**

* Que es el unico atributo que el cliente/usuario puede acceder

**B. ¿Qué significa que sea static?**

* Que es un valor que, si es modificado, se va a ver reflejado en donde haya sido referenciado o instanciado

**C. ¿Qué significaría que fuera final? ¿Debe serlo?**

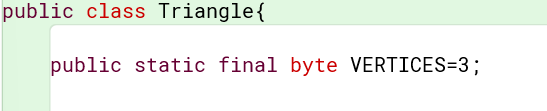
* Que sea final significa que no se le puede cambiar el valor una vez sea asignado, si debe serlo, ya que al ser un triángulo, su número de vértices no debería cambiar nunca

**D. ¿De qué tipo de dato debería ser (byte, short, int, long)? ¿Por qué?**

* Deberia ser byte, ya que este tipo de dato tiene un numero de rango mas limitado que el tipo int, y al ser un valor de 3, que no cambia, es mas eficiente usar este tipo de dato

**E. Actualícenlo.**

* **Se encuentra en el BlueJ**

****

**7. En el código de la clase Triangle revisen el detalle del tipo del atributo height**

**(a) ¿Qué se está indicándo al decir que es int?**

* Que la posicion solo puede tomar un valor entero como valor de altura

**(b) Si height fuera byte, ¿cuál sería el triangulo más grande posible?**

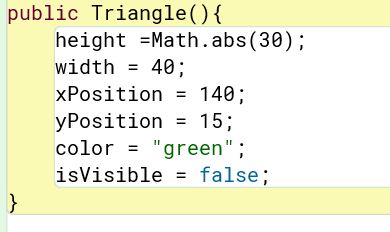
* Hasta 127 podria ser el triangulo mas grande

**(c) y ¿si fuera long?**

* Podria ser hasta 9,223,372,036,854,775,807.

**(d) ¿qué restricción adicional deberían tener este atributo? Refactoricen el código considerando (d).**

* Deberia ser unicamente un valor positivo, ya que no podemos tener una altura negativa



**B. Manipulando objetos. Usando un objeto.**

**1. Creen un objeto de cada una de las clases que lo permitan**

**(a) ¿Cuántas clases hay?**

* Hay 4 clases: Circle(), Rectangle(), Canvas(), Triangle()

**(b) ¿Cuántos objetos crearon?**

* Son solo 3 objetos (circle1, rectangle1, triangle1) debido a que la clase canvas no nos permite crear un objeto como tal

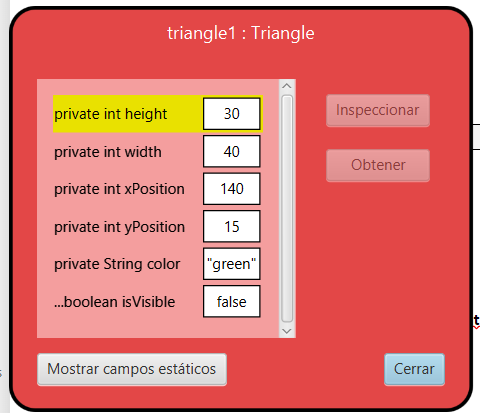
**(c) ¿Quién se crea de forma diferente? ¿Por qué?**

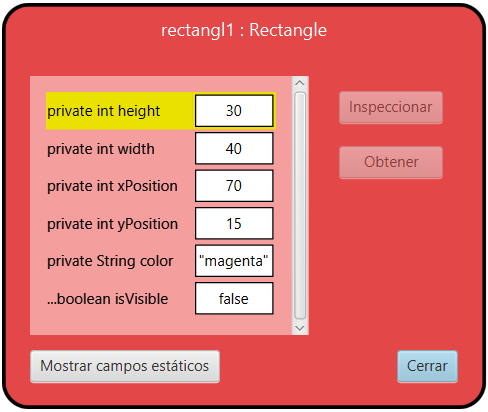
* Se crea diferente en Canvas debido a que es el lienzo/boceto que recopila las otras clases.

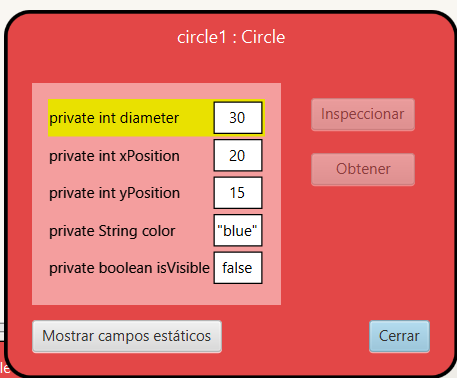
**2. Inspeccionen el estado del objeto :’Triangle’**

**A. ¿Cuáles son los valores de inicio de todos sus atributos?**

**B. Capturen la pantalla.**

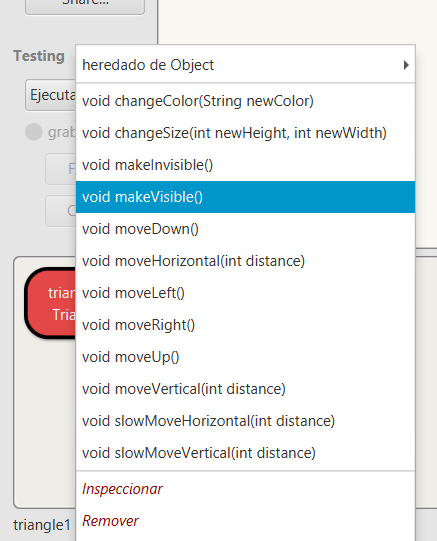






**3. Inspeccionen el comportamiento que ofrece el objeto :Triangle.**

**A. Capturen la pantalla.**



**B. ¿Por qué no aparecen todos los que estan en el codigo?**

* Debido a que solo nos va a mostrar todo lo que se encuentra público.

**4. Construyan, con “shapes” sin escribir código, una propuesta de la imagen del logo de su red social favorita.**

**A. ¿Cuántas y cuáles clases se necesitan?**

* Se necesitaron 3 clases: Canvas, Circle y rectangle.

**B. ¿Cuántos objetos se usan en total?**

* Se necesitaron 10 de tipo rectangle y 2 de tipo circle.

**C. Capturen la pantalla.**

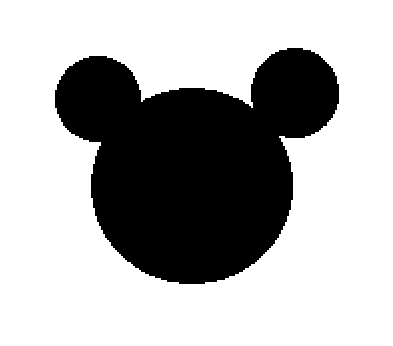


**D. Incluyan el logo original.**



**C. Manipulando objetos. Analizando y escribiendo código**

**1. Lean el código anterior. (a) ¿cuál creen que es la figura resultante? (b) Píntenla.**



**2. Habiliten la ventana de código en línea6 , escriban el código. Para cada punto señalado indiquen:**

**(a) ¿cuántas variables existen?**

Hay 3 variables

**(b) ¿cuántos objetos existen? (No cuenten ni los objetos String ni el objeto Canvas)**

2 Objetos, face, leftEar

**(c)¿qué color tiene cada uno de ellos?**

Todos son negros

**(d) ¿cuántos objetos se ven?**

Solo se ven 3

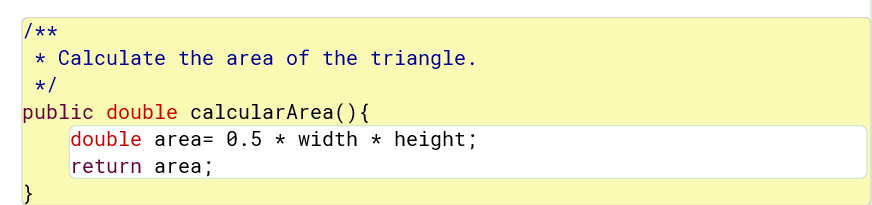
**3. Compare figura pintada en 1. con la figura capturada en 2., (a) ¿son iguales? (b) ¿por qué?**

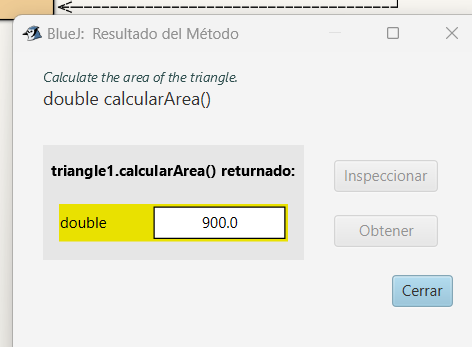


Tienen una forma parecida, pero cambio la posición de las “orejas”. Por qué determinar la posición sobre otra, es más complicado cuando se hace de forma manual.

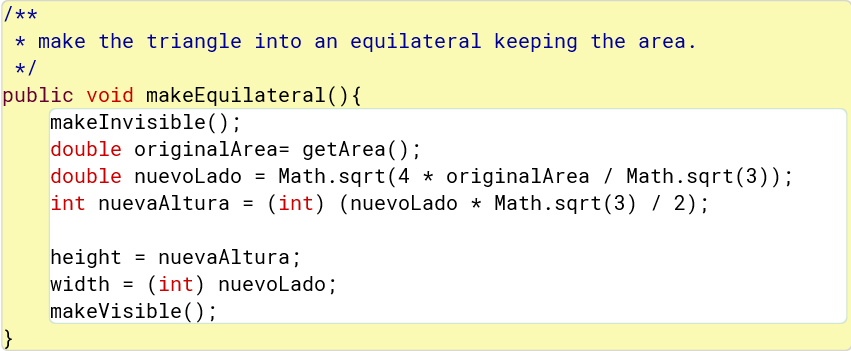
**D. Extendiendo una clase. Triangle.**

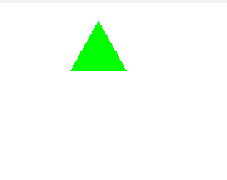
**1. Desarrollen en Triangle el método area (). ¡Pruébenlo! Capturen una pantalla.**



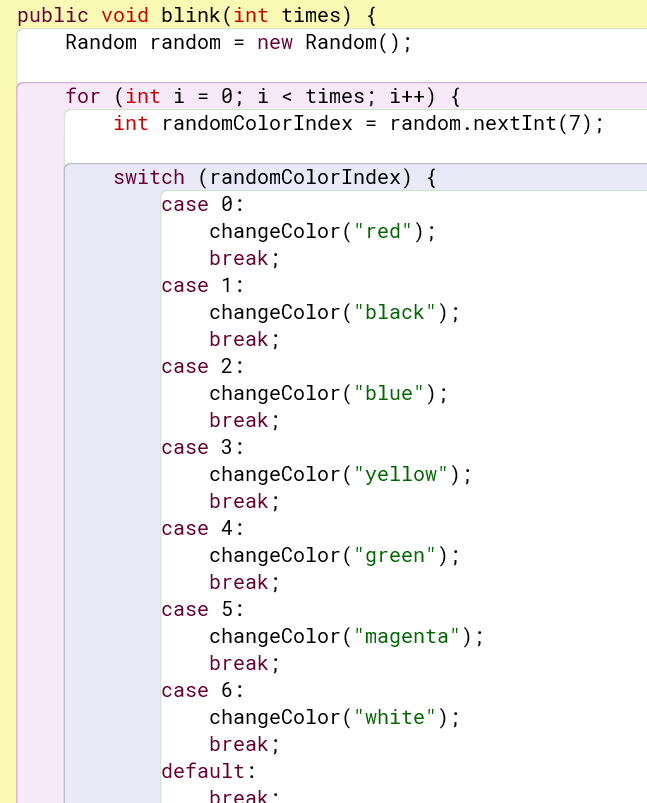


**2. Desarrollen en Triangle el método equilateral () (convierte el triangulo en un equilatero manteniendo el área). ¡Pruébenlo! Capturen dos pantallas.**



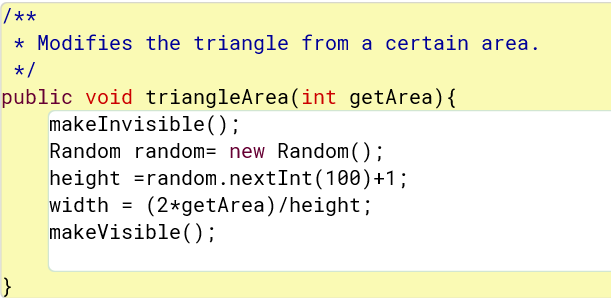


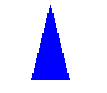
**3. Desarrollen en Triangle el método blink(times:int) (si times>0, cambia de color el número de veces indicado (Por ejemplo, original – nuevo – original serían dos veces. El color nuevo se escoge al azar entre los disponibles ). ¡Pruébenlo! Capturen dos pantallas.**



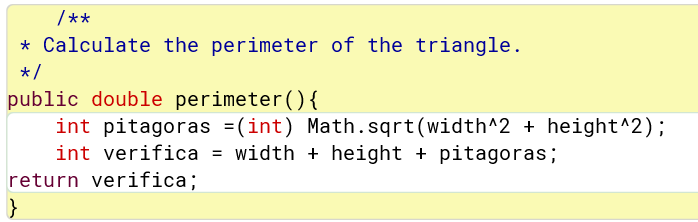


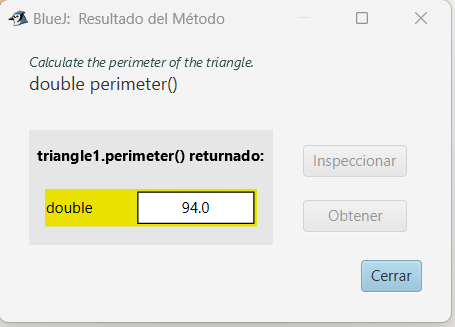
**4. Desarrollen en Triangle un nuevo creador que dada el area crea un triangulo al azar con dicha área.**





**5. Propongan un nuevo método para esta clase. Desarrolle y prueban el método.**





**6. Generen nuevamente la documentación y revise la información de estos nuevos métodos. Capturen la pantalla.**

* Capturas con la documentación en cada punto.

**E. Extendiendo un paquete. Shapes + Hexagono**

**1. Para la construcción, tomen como base la clase Triangle inicial.**

**(a) ¿Qué métodos podrían quedar iguales?**

* Pueden quedar iguales todos los metodos de cambiar colores, mover hacia los lados y los demas que estan relacionados a las anteriores figuras.

**(b) ¿Qué atributos podrian quedar iguales?**

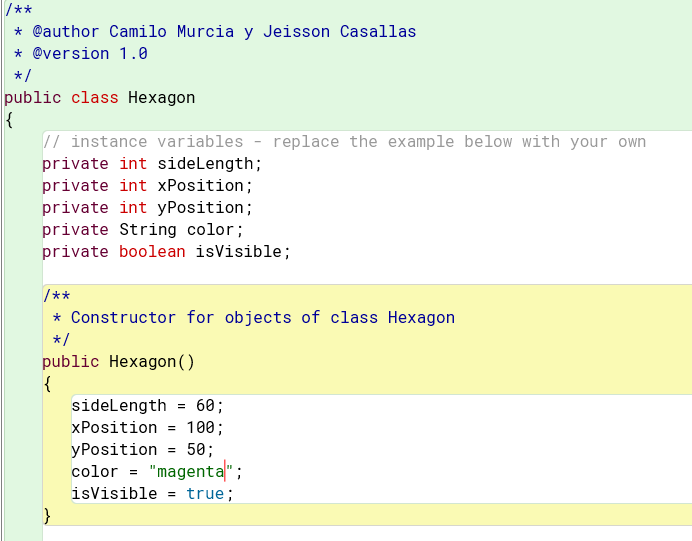
* Los atributos que pueden quedar iguales pueden ser las posiciones, tanto en x como en y, el color, y si es visible o no.

**(c) Justifiquen su respuesta.**

* Quedan estos atributos, ya que en el caso de xPosition y yPosition, son necesarios para dibujar el hexágono y deben estar para que este funcione. Y el isVisible de igual forma, ya que se debe determinar la visibilidad de la figura y no cambia sin importar el polígono que se dibuje.

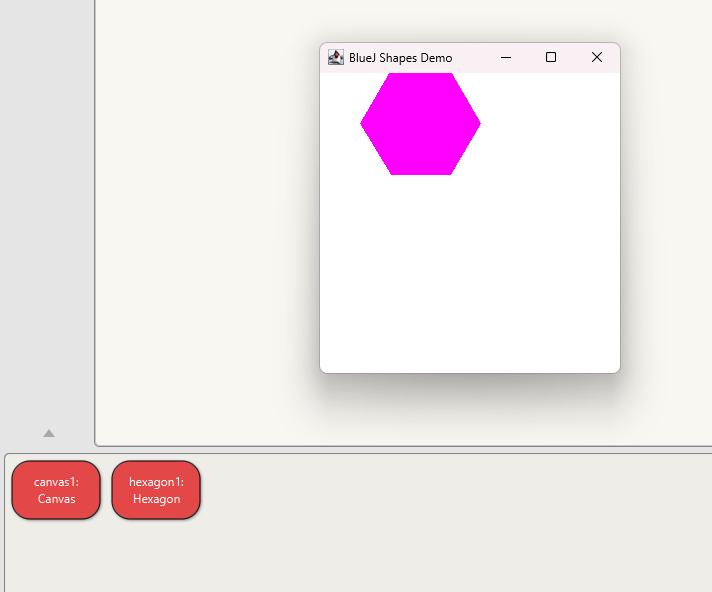
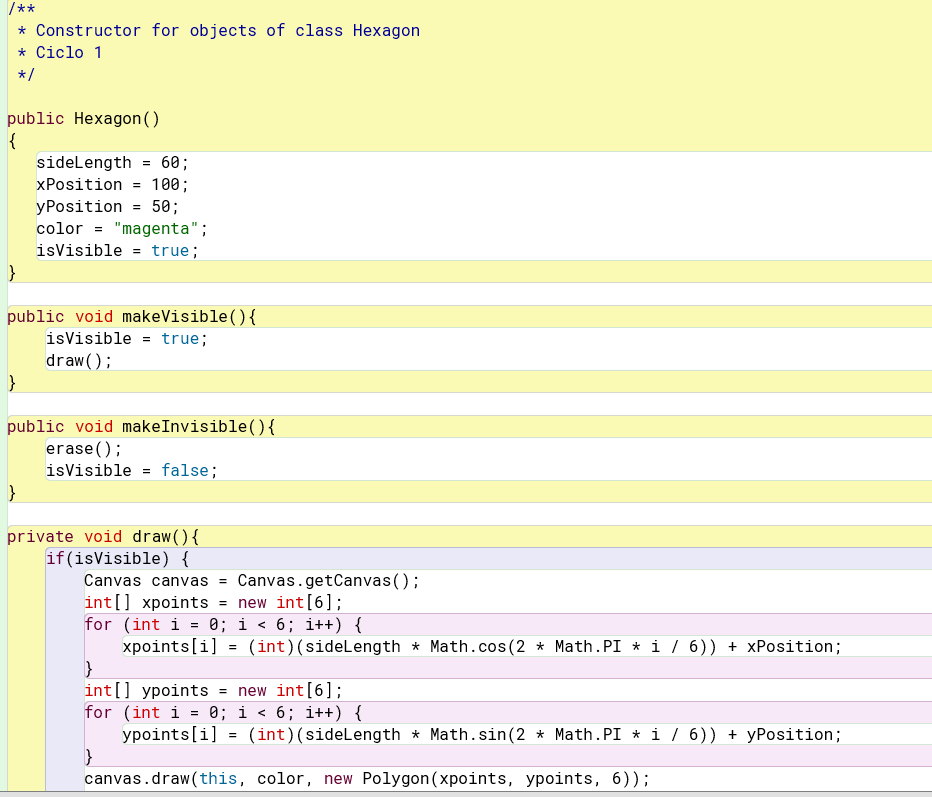
**2. Inicie la construcción de la clase Hexagon únicamente con los atributos.**

**(a) Adicione pantallazo con los atributos.**



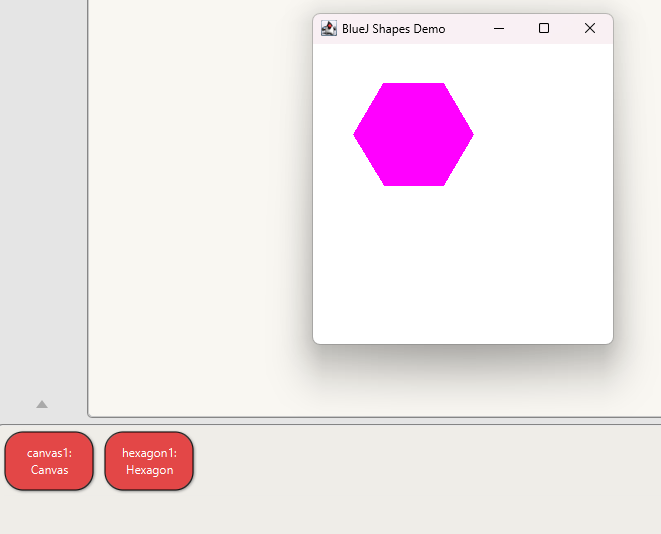
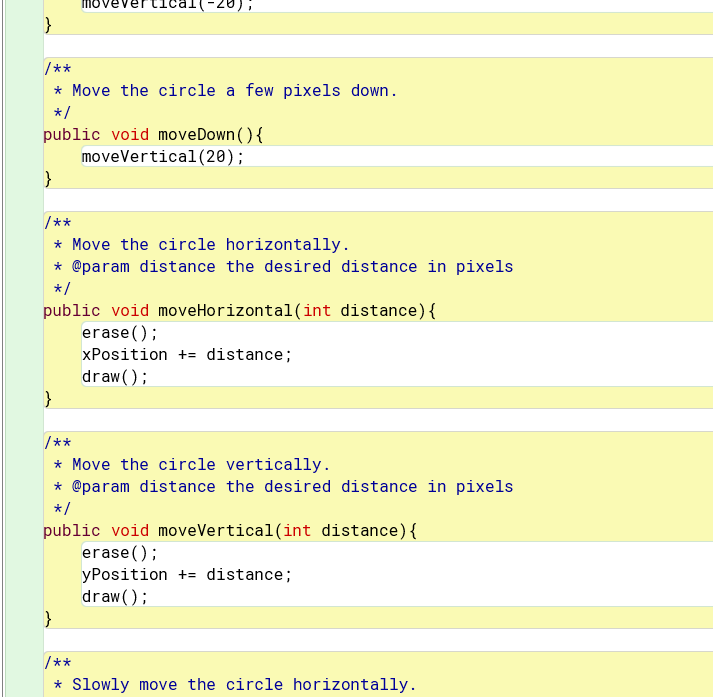
**3. Desarrollen la clase Hexagon considerando los 3 mini-ciclos. Al final de cada miniciclo realicen una prueba indicando su propósito. Capturen las pantallas relevantes.**

**Ciclo 1:**



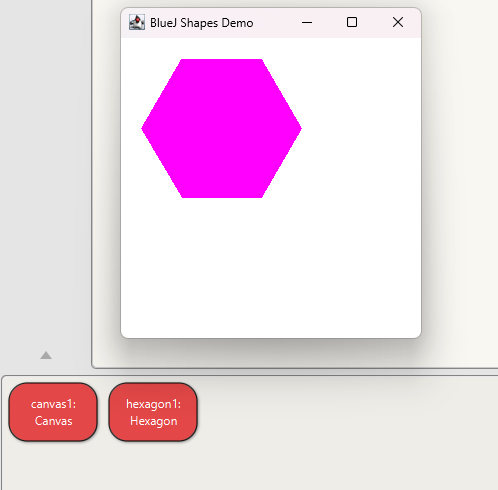
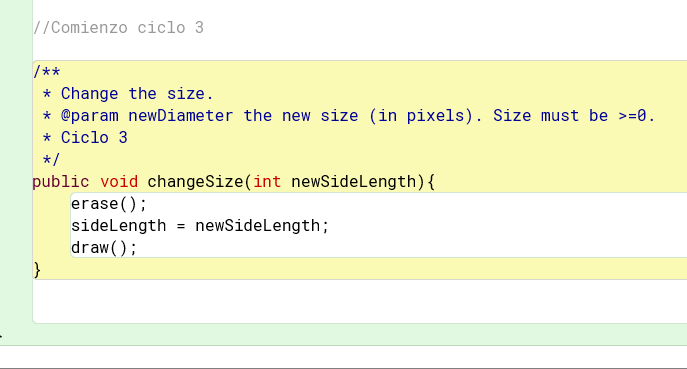
* Se usa el metodo makeVisible para poder verlo en pantalla, de igual forma se puede aplicar el metodo makeInvisible para que no aparezca.

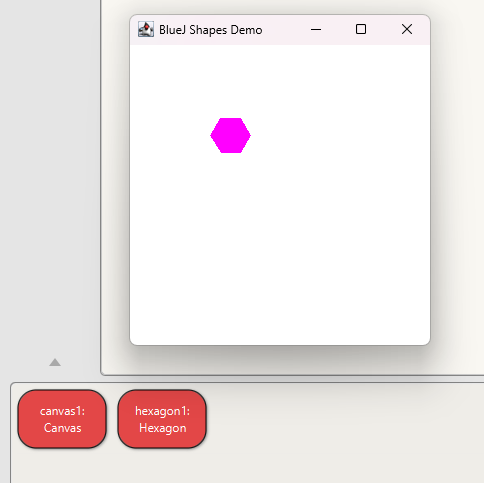
**Ciclo 2:**



* Se usa el metodo moveDown para cambiar su posicion y quede mas abajo de la posicion original, igualmente se puede mover para cualquiera de las otras direcciones para que aparezca en la pantalla.

**Ciclo 3:**





* Se utiliza el método changeSize para cambiar el tamaño del hexágono bien sea para que sea más pequeño o más grande según se le asigne el valor.

**F. Definiendo y creando una nueva clase. Replicate.**

El objetivo de este trabajo es programar una mini-aplicación para **Replicate.**

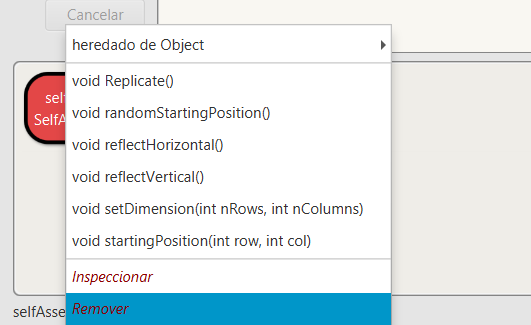
**Requisitos funcionales**

* Crear un replicate indicando su dimensión.
* Llenar una celda. Deben ofrecer dos formas de crear la pieza: (a) dada su posición (b) al azar
* Replicar. Debe seguir las reglas.
* Reflejar vertical u horizontalmente el replicate

**Requisitos de interfaz**

* Las celdas se identifican por su fila y su columna.
* En caso que no sea posible realizar una de las acciones, debe generar un mensaje de error. Use JoptionPane.

**1. Diseñen la clase, es decir, definan los métodos que debe ofrecer.**

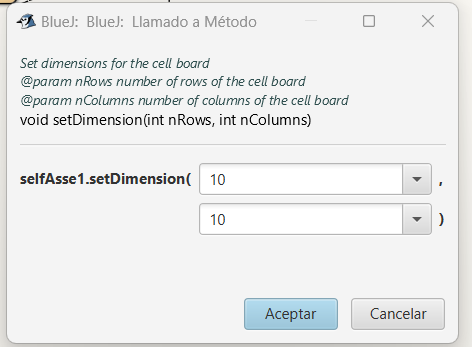


**2. Planifiquen la construcción considerando algunos miniciclos.**

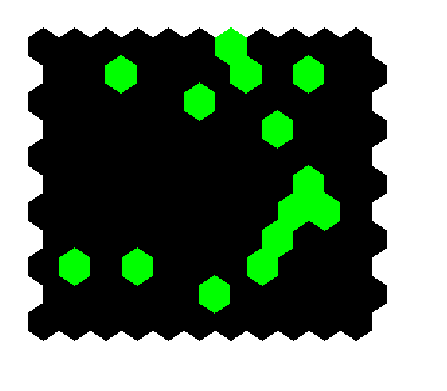
* **Ciclo 1:** setDimension(int nRows, int nColumns), Replicate ()
* **Ciclo 2:** StartingPosition(int row, int col), randomStartingPosition()
* **Ciclo 3:** reflectVertical(), reflectHorizontal()

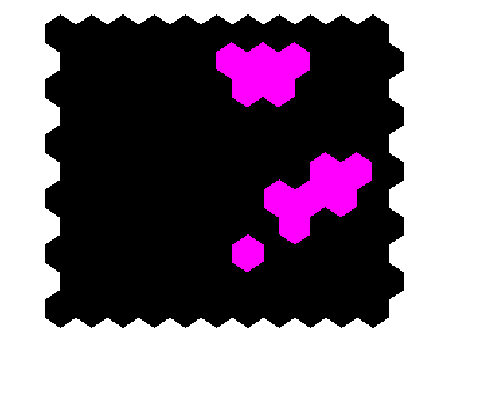
**3. Implementen la clase . Al final de cada mini-ciclo realicen una prueba indicando su propósito. Capturen las pantallas relevantes.**

**Ciclo 1:**

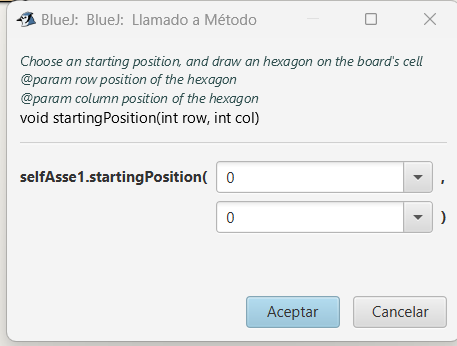


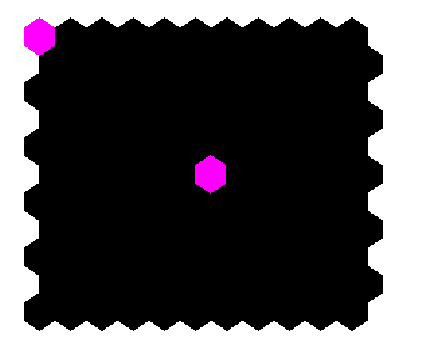


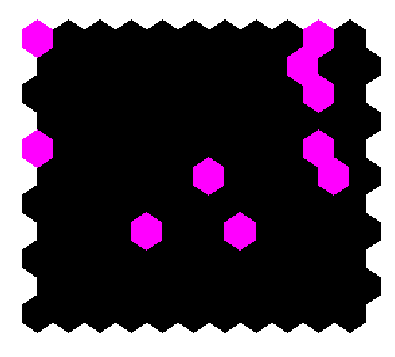




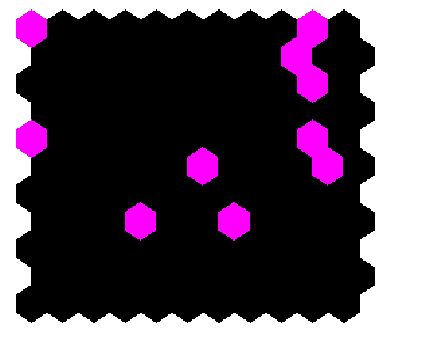
**Ciclo 2:**

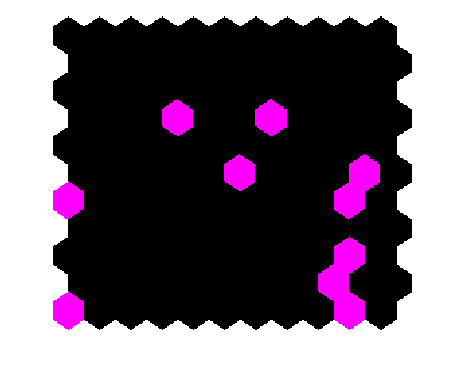


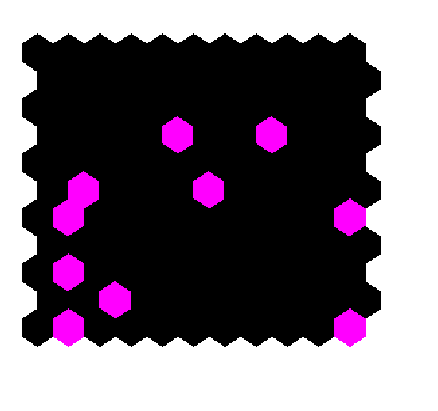




**Ciclo 3:**







**4. Indiquen las extensiones necesarias para reutilizar el paquete shapes y la clase Hexagon. Explique.**

import java.awt.Polygon;

import java.awt.\*;

import java.awt.geom.\*;

import java.lang.Math;

Public void changeColor(String newColor)

Public void changeSize(int newSideLength)

Public void makeVisible()

Public void moveHorizontal()

Public void moveVertical()

Todos estos son las extensiones para reutilizar el paquete Shapes y la clase Hexagon, ya que estos son los requeridos para el funcionamiento de la creación de cada una de las figuras teniendo en cuenta que son métodos nuevos que utilizan, estos mismos antes creados en otras clases para desarrollar nuevas funcionalidades.

**RETROSPECTIVA**

1. **¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes? (Horas/Hombre)**

Para este laboratorio nos tomamos 22 horas por persona

1. **¿Cuál es el estado actual del laboratorio? ¿Por qué?**

Casi finalizado, aunque falta pulir pequeños detalles

1. **Considerando las prácticas XP del laboratorio. ¿cuál fue la más útil? ¿por qué?**

La práctica de programación a pares fue la más útil, ya que en un trabajo tan extenso como este, es muy necesario una colaboración mutua y una buena comunicación.

1. **¿Cuál consideran fue el mayor logro? ¿Por qué?**

Terminar el laboratorio, ya que estaba bastante extenso y complejo

1. **¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolverlo?**

La creación del metodo para implementar el replicate.

1. **¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?**

La programación a pares que se vio en las practicas XP, ya que fue clave para poder culminar el laboratorio