Escuela Colombiana de Ingeniería Julio Garavito

Andrés Cardozo

Tulio Riaño

Laboratorio 01

Programacion Orientada a Objetos

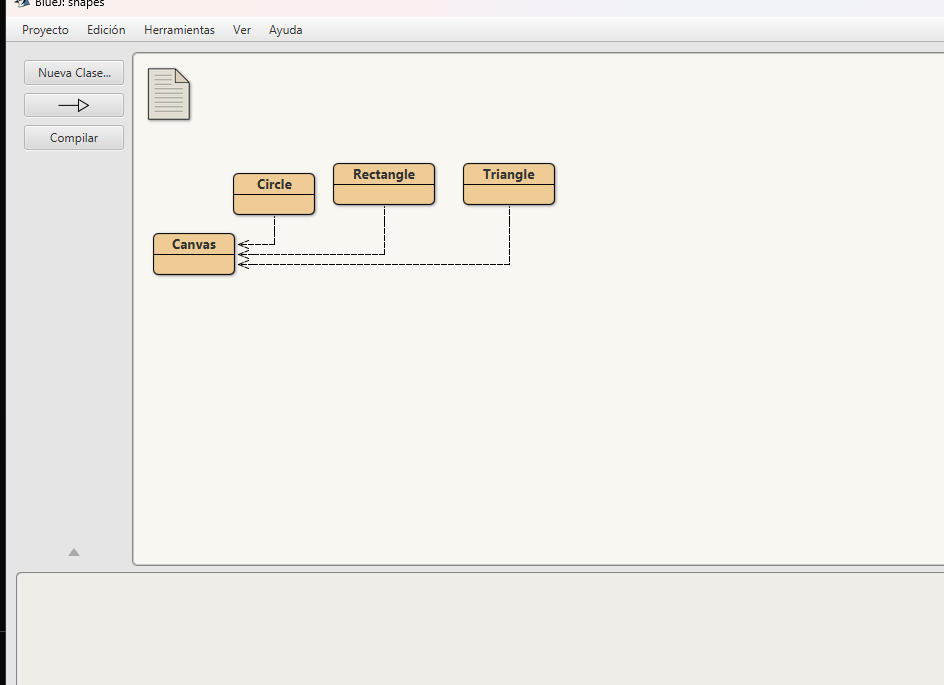
FECHA:

8/02/2025

**SHAPES**

**A. Conociendo el proyecto shapes**

1. El proyecto “shapes” es una versión modificada de un recurso ofrecido por BlueJ.



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? Ofrece 4 clases: Circle, Rectangle, Triangle y Canvas.

(b) ¿Qué relaciones existen entre ellas?

Las clases Circle, Rectangle, Triangle están relacionadas con Canvas, las figuras geometricas utilizan a canvas para pintar.

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? Canvas, circle, Rectangle y Triangle.

(b) ¿Qué atributos tiene la clase Triangle? Tiene Vertices.

(c) ¿Cuántos métodos ofrece la clase Triangle? La clase Triangle ofrece 12 métodos

(d) ¿Qué atributos determinan el tamaño de un Triangle? Los atributos que determinan el tamaño de un Triangle es height y width.

(d) ¿Cuáles métodos ofrece la clase Triangle para que la figura cambie su tamaño (incluya sólo el nombre)? Solo 1 método ofrece la clase Triangle que sería Change Size

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?

Tiene 7 atributos que son: Vertices, height, width, xPosition, yPosition, color, isVisible.

(b) ¿Quiénes pueden usar los atributos públicos?

Cualquier clase tiene acceso a estos atributos.

Con respecto a los métodos:

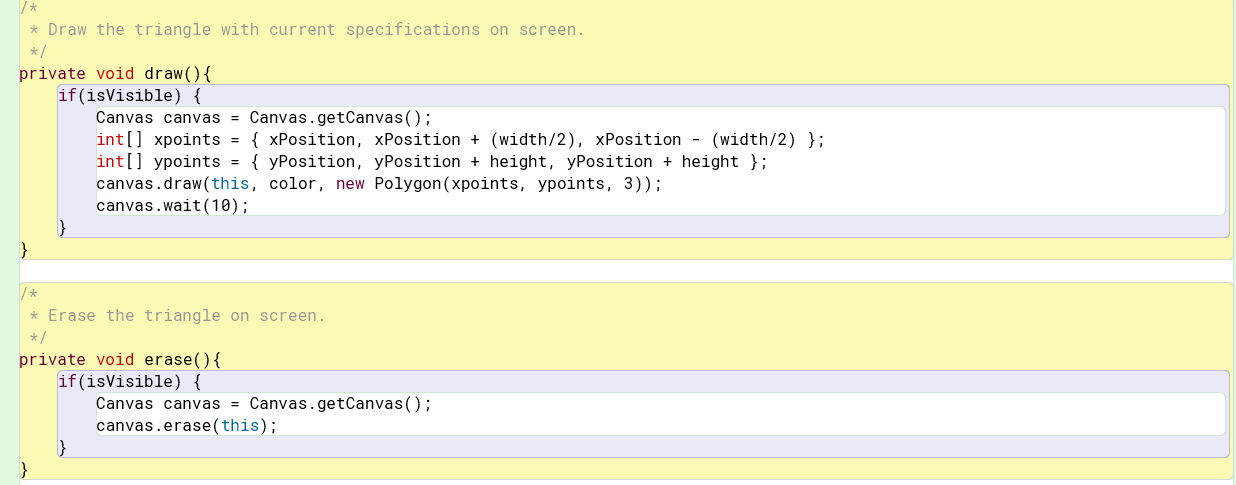
(c) ¿Cuántos métodos tiene en total? 14 metodos.

(d) ¿Quiénes usan los métodos privados? Solo la misma clase los puede usar.

5. Comparando la documentación con el Código

(a) ¿Qué no se ve en la documentación?

Los métodos y atributos que son declarados como privados.



(b) ¿Por qué debe ser así?

En el caso de los atributos es debido a que van a estar definidos específicamente para la misma clase y así mismo estas presentan características especiales como la cantidad de vértices que tiene un triángulo, etc. Por lo que las demas clases y objetos no podran iteractuar con las privadas, y no sale en la documentación.

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

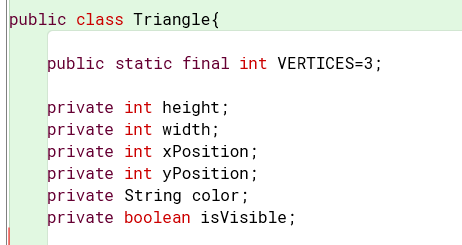
(a) ¿Qué significa que sea public?

Que se puede ver en la documentación y otra clase puede acceder a esta.

(b) ¿Qué significa que sea static? Significa que el atributo pertenece a la clase en sí, en lugar de a instancias individuales de esta.

(c) ¿Qué significaría que fuera final? ¿Debe serlo? Significa que este valor nunca nunca va a cambiar una vez asignado, de la misma manera los metodos no van a tener acceso a esta variable para cambiarla. Si deberia serlo ya que los triangulos por lo general tienen 3 vertices.

(d) Actualícenlo.



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 el tipo de atributo es un entero.

(b) Si fuera byte, ¿cuál sería el área del Triangle más grande posible?

El tipo de atributo va de -128 a 127, en este sentido:

Si es un triángulo equilátero, seria : (127\*127)/2 = 8064.5

Si no, tocaria especificar cual es el valor de width: (width\*127)/2

(c) y ¿si fuera long?

El tipo de atributo va de -9,223,372,036,854,775,808 a 9,223,372,036,854,775,807 en este sentido:

Si es un triangulo equilatero, seria : (9,223,372,036,854,775,807 \* 9,223,372,036,854,775,807)/2 = 4.25x10^37.

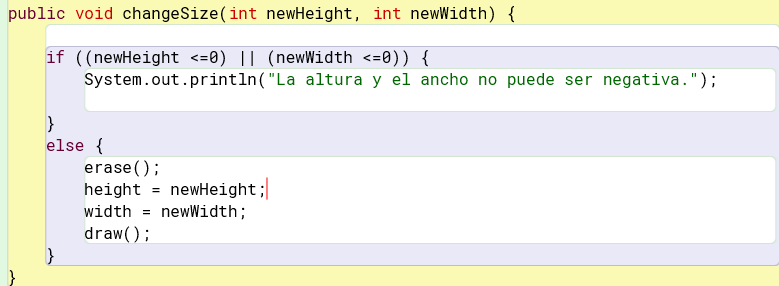
Si no, tocaria especificar cual es el valor de width:

(width\* 9,223,372,036,854,775,807)/2

d) ¿qué restricción adicional deberían tener este atributo?

Que no sea un valor negativo

(e) Refactoricen el código considerando (d).



8. ¿Cuál dirían es el propósito del proyecto “shapes”?

El propósito podría ser crear dibujos a partir de ciertas figuras geométricas con canvas.

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

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





(a) ¿Cuántas clases hay?

4 Clases.

(b) ¿Cuántos objetos crearon?

3 objetos.

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

El canvas es el que se crea de forma diferente debido a que es la clase donde se dibujan los objetos de las otras clases, por otra parte, no es necesario crearlo debido que apenas se dibuje un objeto este se crea simultáneamente.

2. Inspeccionen los creadores de cada una de las clases.

(a) ¿Cuál es la principal diferencia entre ellos?

En las clases de las figuras geométricas, los autores son Michael Kolling y David J. Barnes, mientras que en la clase canvas son: Bruce Quig con Michael Kolling.

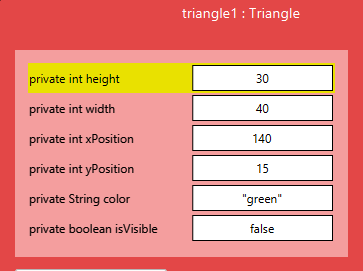
(b) ¿Qué se busca con la clase que tiene el creador diferente?

La clase que tiene un creador diferente es CANVAS esta busca permitir visualizar los gráficos o dibujos específicos de cada una de las clases asociadas.

3. Inspeccionen el estado del objeto: Triangle

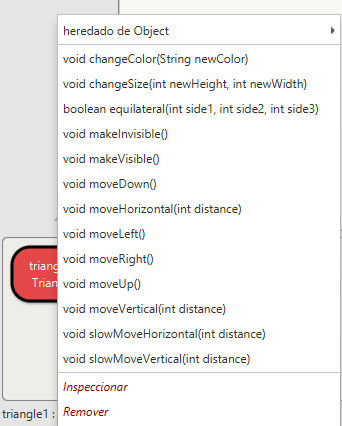
(a) ¿Cuáles son los valores de inicio de todos sus atributos? Los definidos por default.

(b) Capturen la pantalla.



4. Inspeccionen el comportamiento que ofrece el objeto :Triangle.

(a) Capturen la pantalla.



(b) ¿Por qué no aparecen todos los que están en el código?

Porque hay métodos que fueron definidos como privados en la respectiva clase.

5. Construyan, con “shapes” sin escribir código, una propuesta de la imagen del logo de

de su chatbot IA favorito.

(a) ¿Cuántas y cuáles clases se necesitan?

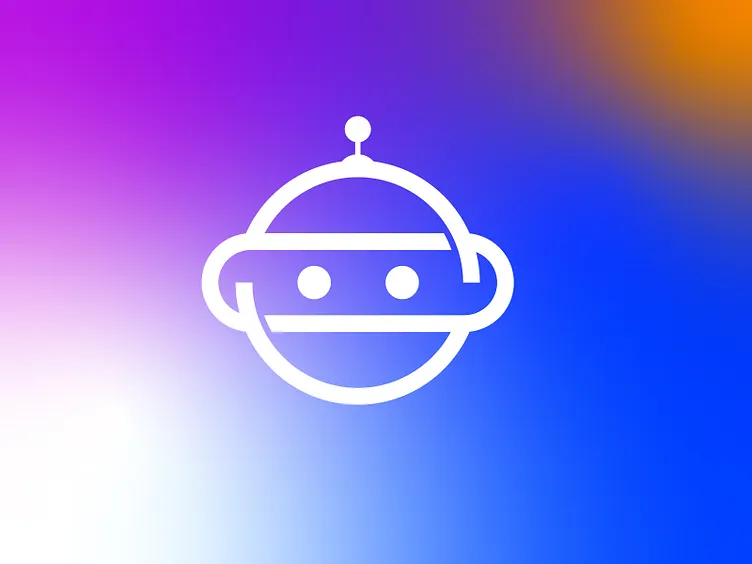
Se necesitan 2 clase, las figuras geométricas definidas: Circle y Rectangle.

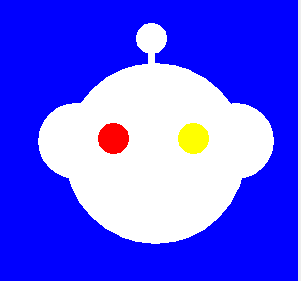
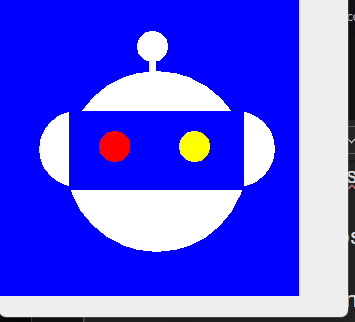
(b) ¿Cuántos objetos se usan en total?

9 objetos fueron usados

(c) Capturen la pantalla. (d) Incluyan el logo original.

Phi-1.5





<https://clarifai.com/microsoft/text-generation/models/phi-1_5>

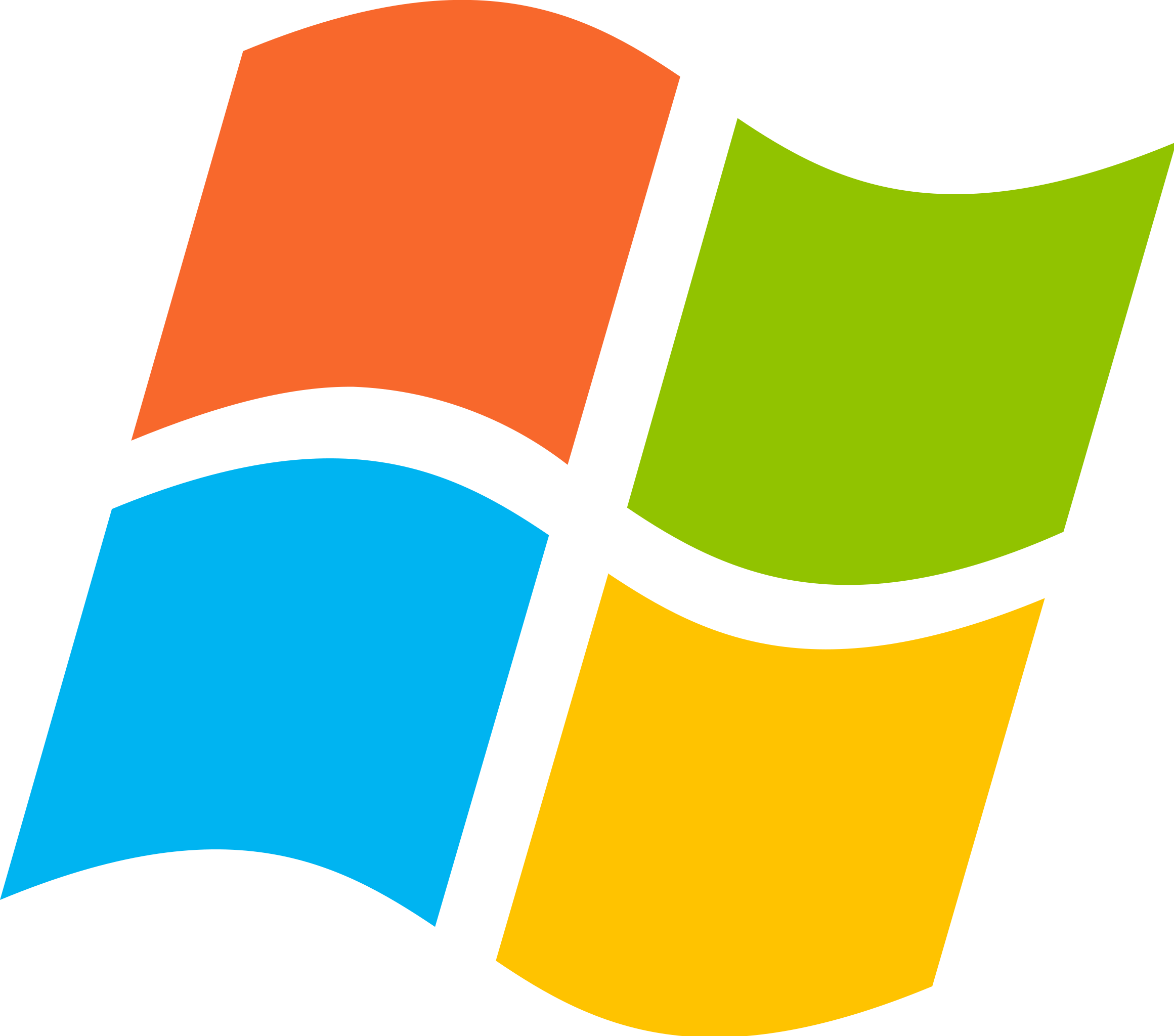
**C. Manipulando objetos. Analizando y escribiendo código.**



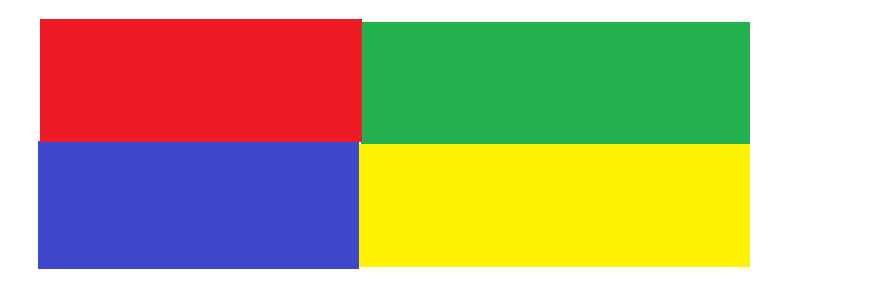
1. Lean el código anterior.

(a) ¿cuál creen que es la figura resultante?

Podria ser similar al logo de windows:



(b) Píntenla.



2. Habiliten la ventana de código en línea 7, escriban el código. Para cada punto señalado indiquen:

(a) ¿cuántas variables existen? 3 variables

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

Existen 3 objetos ya que el objeto rojo tiene dos referencias. Azul y rojo.

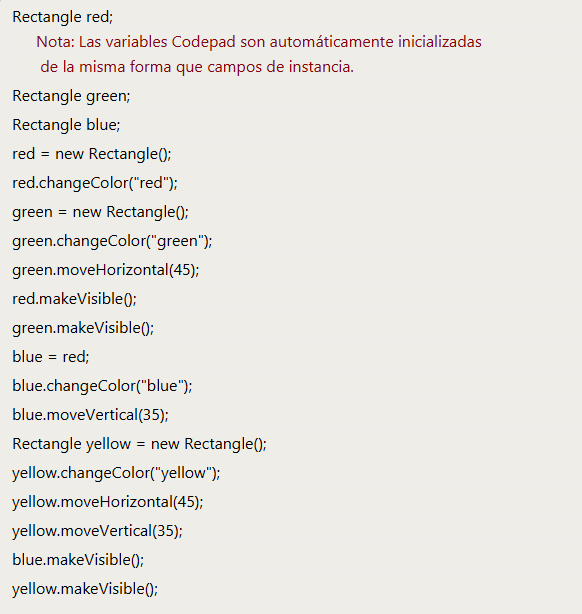
(c) ¿qué color tiene cada uno de ellos?

Azul, amarillo y verde.

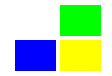
(d) ¿cuántos objetos se ven?

Se ven 3 objetos.

3. Habiliten la ventana de código en línea, escriban el código. (a) Capturen la pantalla.



(f) Capturen la pantalla.



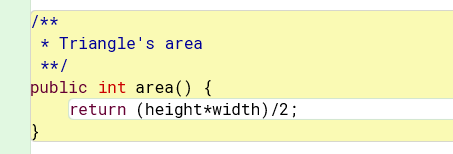
4. Compare figura pintada en 1. con la figura capturada en 3.

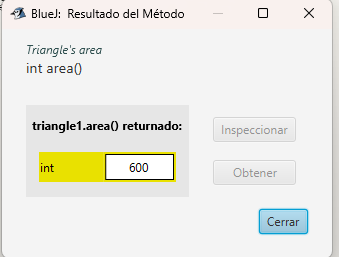
(a) ¿son iguales? (b) ¿por qué?

No, ya que no tomamos en cuenta que al hacer la asignación blue = red, se trataba del mismo objeto.

D. Extendiendo una clase. Triangle.

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



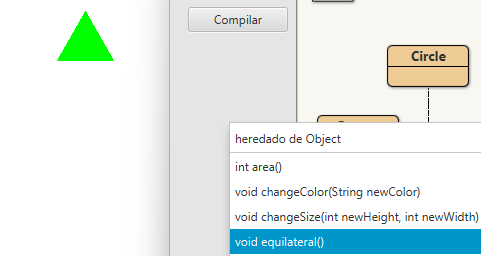


2. Desarrollen en Triangle el método equilateral()

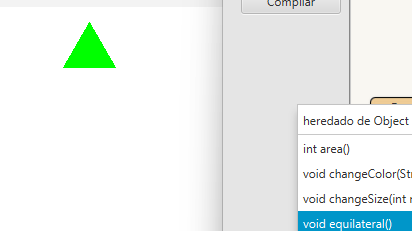
(transforma el triángulo en un triángulo equilátero de área equivalente ) .

¡Pruébenlo! Capturen dos pantallas.

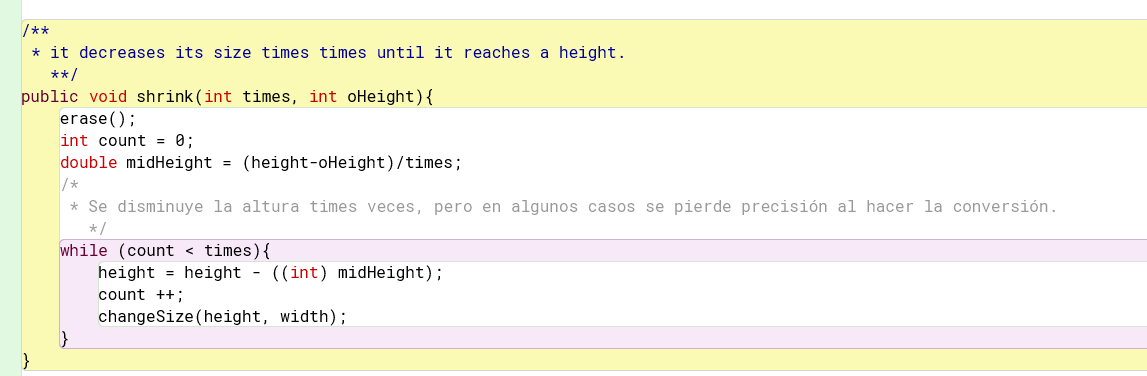
Como estamos trabajando con tipos de datos int para la altura y la anchura, perdemos precisión:

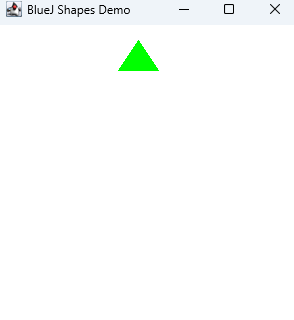
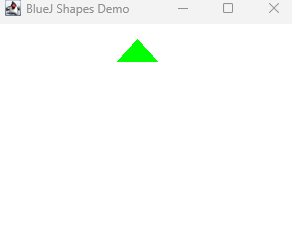
 

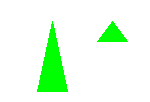
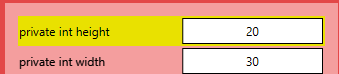
3. Desarrollen en Triangle el método shrink(times:int, height: int) (disminuye su tamaño times veces. Hasta llegar a una altura de height.) ¡Pruébenlo! Capturen tres pantallas.



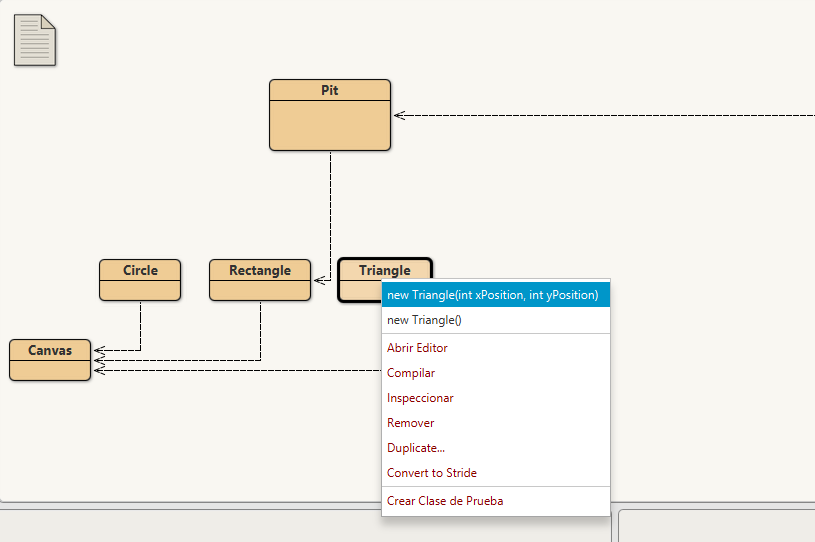
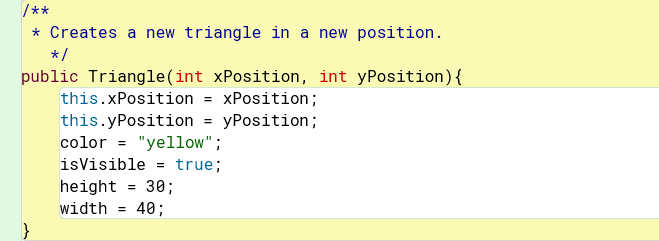
 

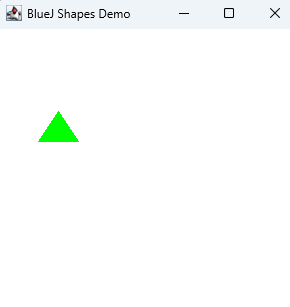
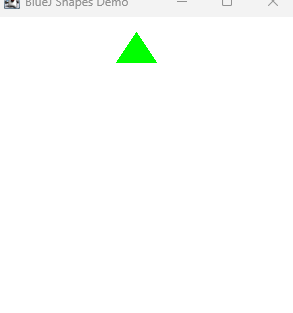
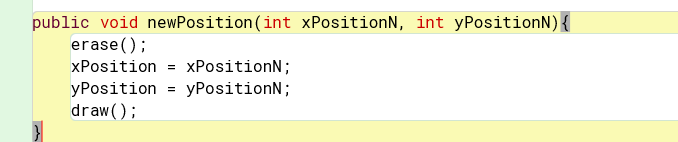
Con 5 veces hasta una altura 20:

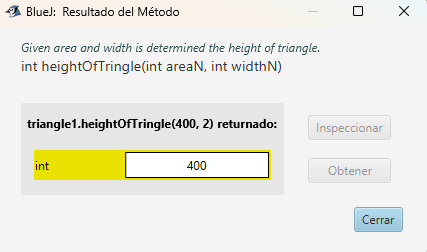
 

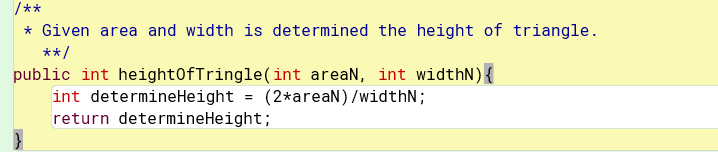
4. Desarrollen en Triangle un nuevo creador que permita crear un triangulo en una

posición específica.¡Pruébenlo! Capturen una pantalla.

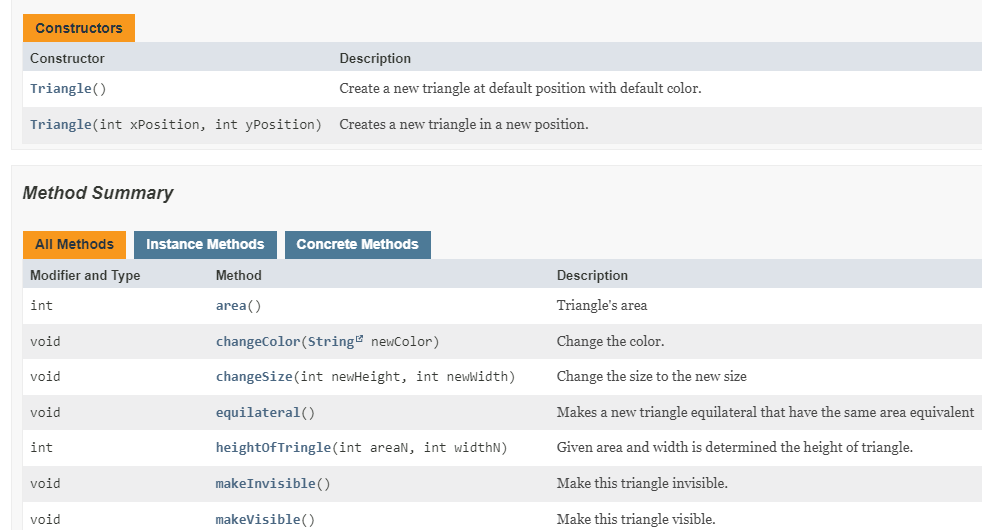


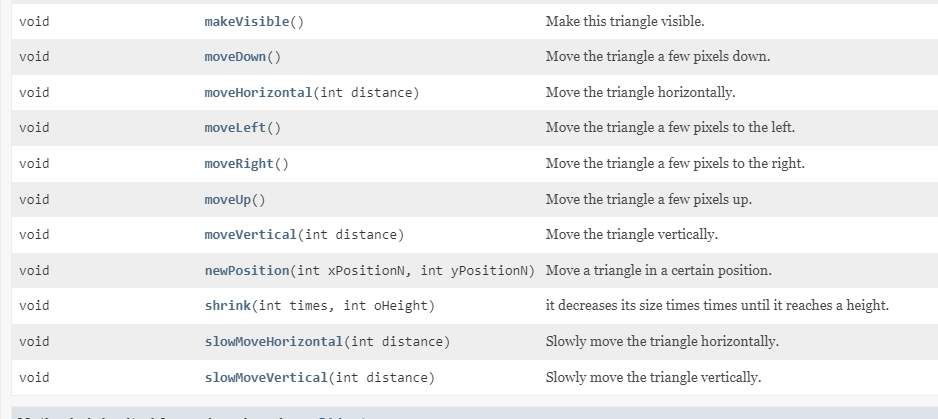
5. Propongan un nuevo método para esta clase. Desarrollen y prueban el método.



6. Generen nuevamente la documentación y revise la información de estos nuevos

métodos. Capturen la pantalla.



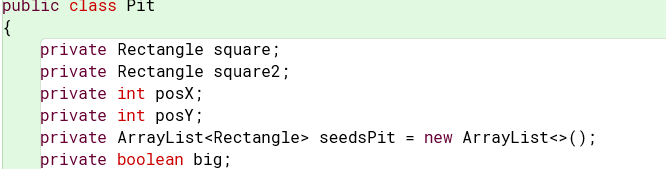


**E. Usando un paquete. Shapes** [En lab01.doc y \*.java]

En este punto vamos a crear huecos en cuadrados para guardar semillas. El diseño gráfico lo

definen ustedes. Estos son algunos ejemplos.

1. Inicie la construcción únicamente con los atributos. Justifique su selección. Adicione pantallazo con los atributos.

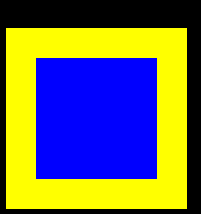
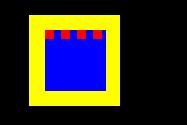
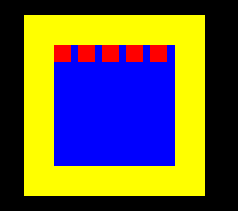


Utilizamos dos atributos de tipo rectangulo para crear la forma de pit (hueco), tambien guardamos en otros dos atributos las posiciones del pit. Tambien decidimos que un atributo sea una lista, en donde guardamos la direccion de las semillas (rectangulos) que creemos. Asi es para manejar cuando las queramos mover o eliminar. EL ultimo atributo es big, que guarda si es grande o no pit.

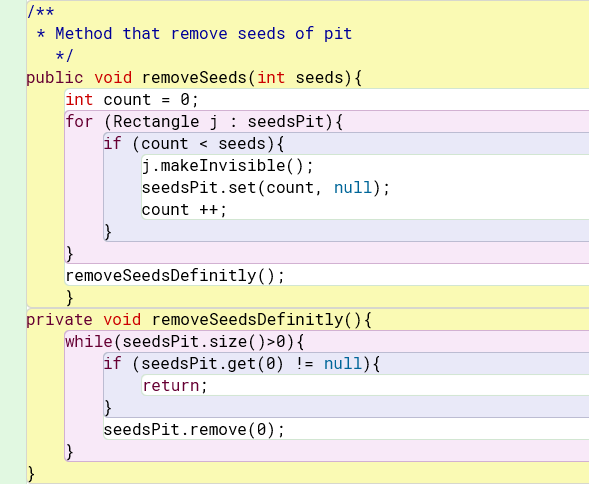
1. Desarrollen la clase considerando los 3 mini-ciclos. Al final de cada mini- ciclo realicen dos pruebas indicando su propósito. Capturen las pantallas relevantes.

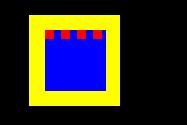
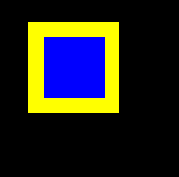
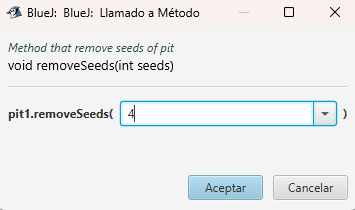
**Mini-ciclo: 1**

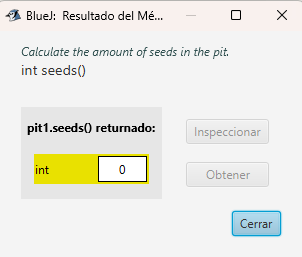
Para el primer ciclo, realizamos dos pruebas que buscan mostrar la creación del Pit y colocar las respectivas semillas. Tomamos como máximo 25 semillas en un pit big, mientras que el no big tiene un máximo de 15 semillas.

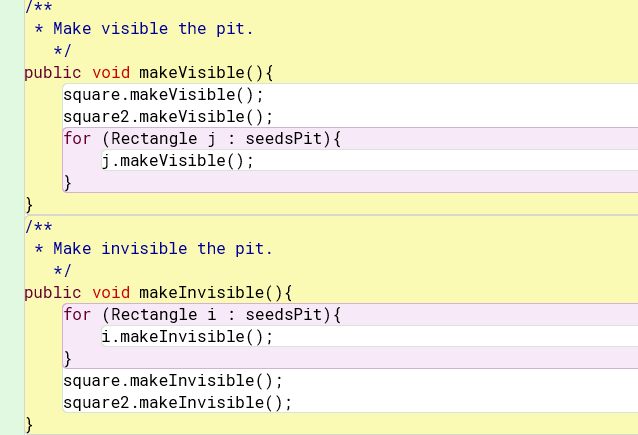
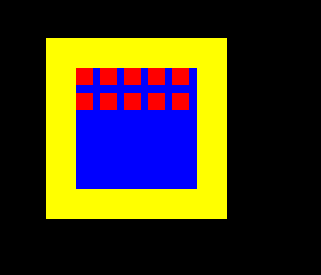
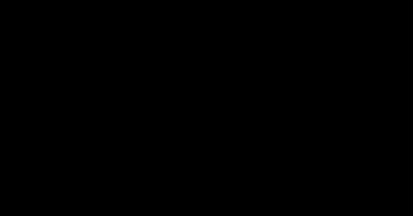
Para la segunda prueba buscamos remover una cantidad de semillas y verificar la cantidad que hay tras su eliminación.





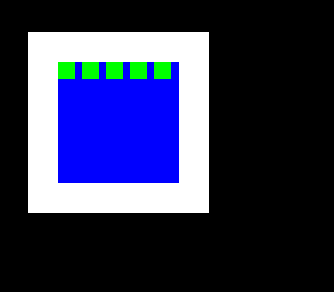
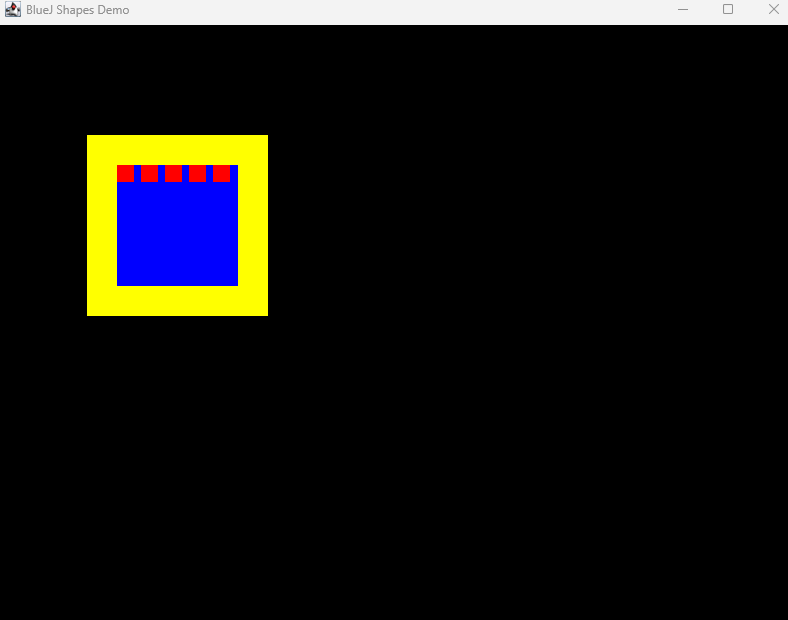
**Mini-ciclo: 2**

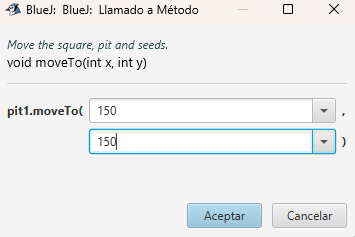
  

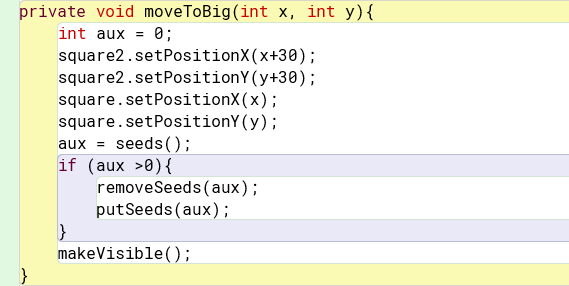
Como vimos en las imágenes, los métodos van de la mano y hacen visible o invisible todo el Pit, incluso con semillas (si tiene).

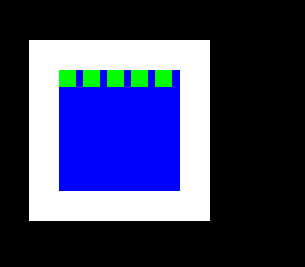
**Mini-ciclo: 3**

Para este último mini ciclo lo que buscamos es cambiar de color el background así mismo como las semillas y mostrar la manera en la que podemos mover el pit.









**G. De python a java**

En este punto vamos a evaluar dos recursos de apoyo para la transición de Python a Java en

las encuestas preparadas con ese objetivo.

* Las encuestas correspodientes fueron formalmente diligenciadas.

**KALAH**

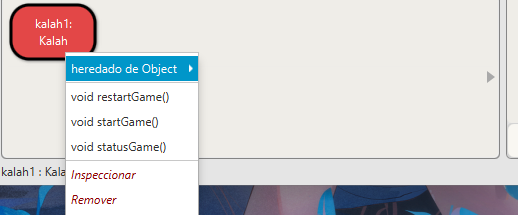
**F. Definiendo y creando una nueva clase. Kalah.**

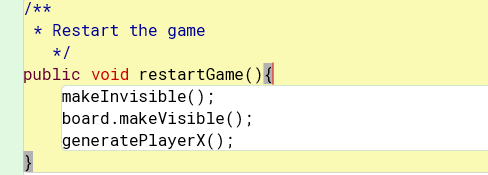
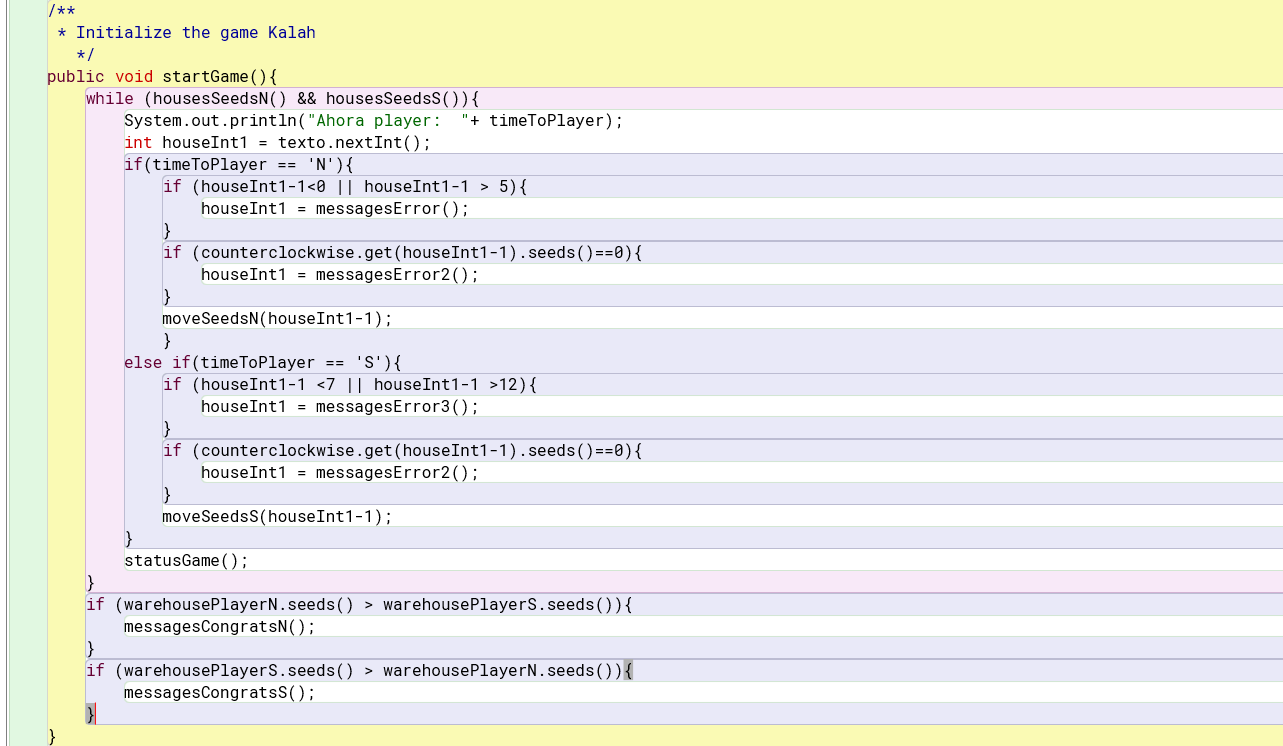
El objetivo de este trabajo es programar una mini-aplicación para Kalah.

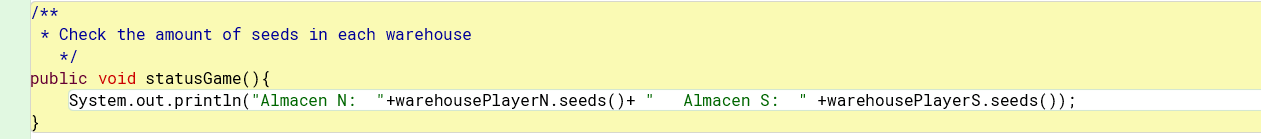
1. Diseñen la clase,es decir, definan los métodos que debe ofrecer.

Para la solución de este punto, importamos librerías como ArrayList, para poder crear listas de objetos en este caso Pit y Rectangle, la librería scanner para poder capturar la información del usuario y swing.\* para poder dar los mensajes de error que se nos indicaba.

Los métodos que consideramos que debe ofrecer Kalah es inicializar el juego, reiniciar el juego.







1. Planifiquen la construcción considerando algunos mini-ciclos.

**MINI-CICLO 1 – Start Game**

**MINI-CICLO 2 – Restart Game**

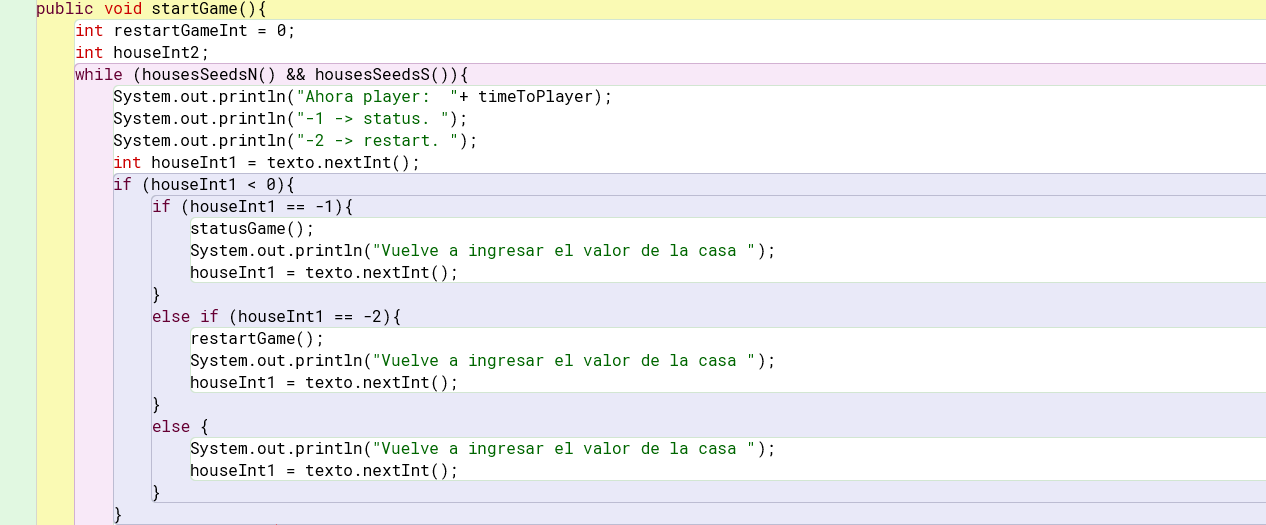
**MINI-CICLO 3 – Status Game**

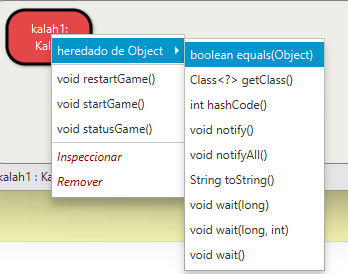
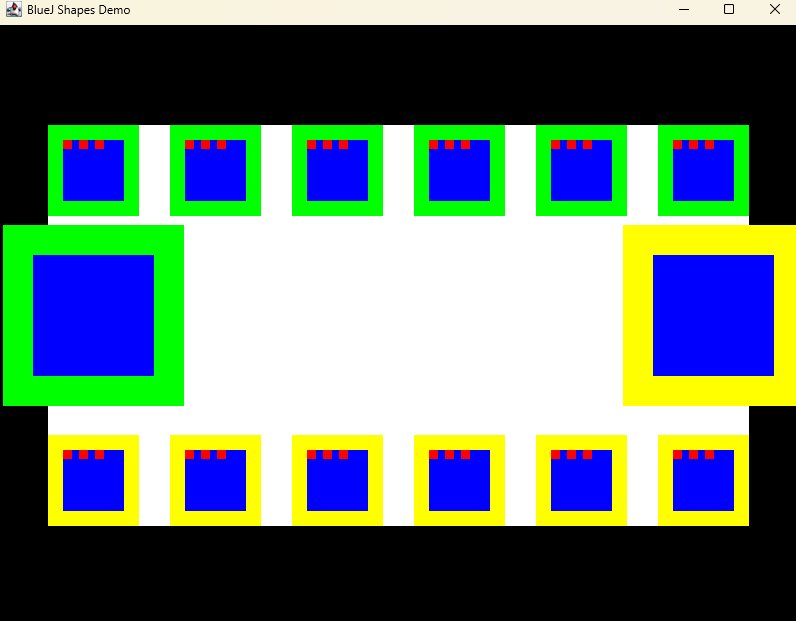
**MINI-CICLO4 – Respetando reglas kalah**

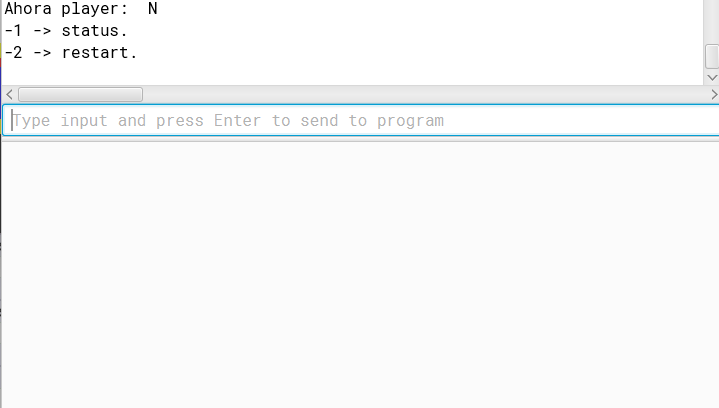
1. Implementen la clase . Al final de cada mini-ciclo realicen una prueba indicando su propósito. Capturen las pantallas relevantes.

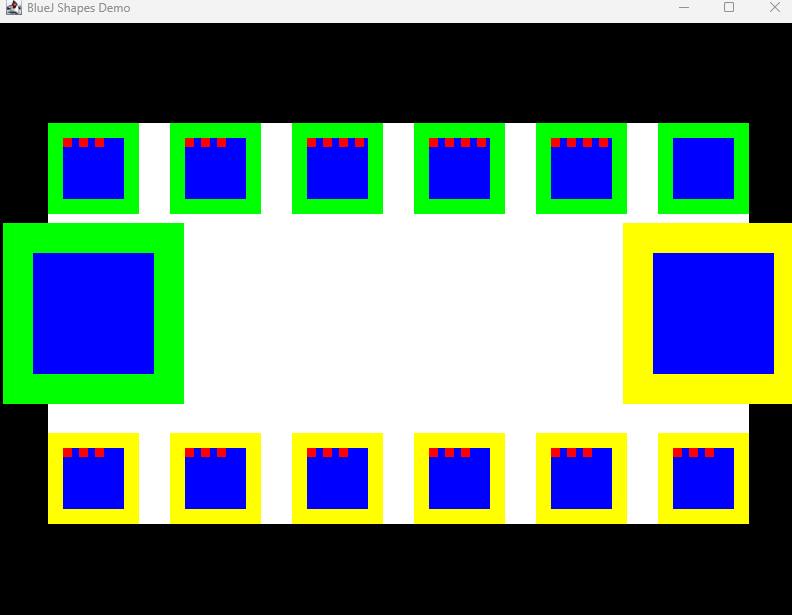
**MINI-CICLO 1**

La manera en la que abordamos el desarrollo del juego Kalah fue mediante un método público en este caso un primer mini ciclo que busca mostrar el tablero de juego completo y mediante el ingreso de números mediante consola vamos a ir jugando, respetando un respectivo orden.

En esta primera parte observamos el tablero de Kalah

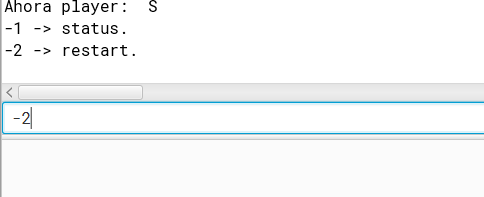


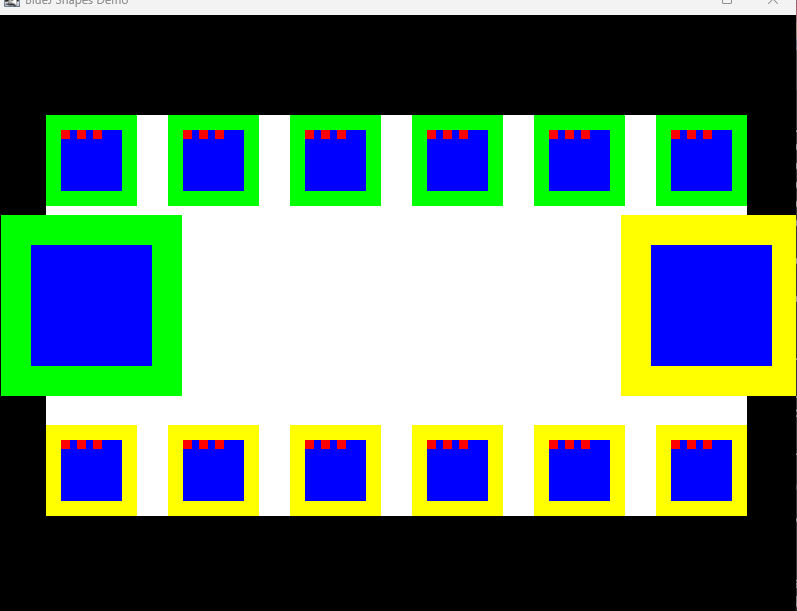
Al momento de ingresar 1 va a empezar el juego en sentido antihorario cogiendo todas las semillas de esta primera casa y colocándolas 1 a 1 en las siguientes casas (n cantidad de semillas a colocar). Para este Mini-Ciclo buscamos resaltar la idea principal del juego manejamos los números de (1-6) para los jugadores N y (8-13) para los jugadores S, así respetando las reglas del juego.



**MINI-CICLO 2**

Para este mini-cilo queremos mostrar el método de restart game, esto lo implementamos en una parte del start game y lo vamos a utilizar mediante consola al ingresar el número -2.

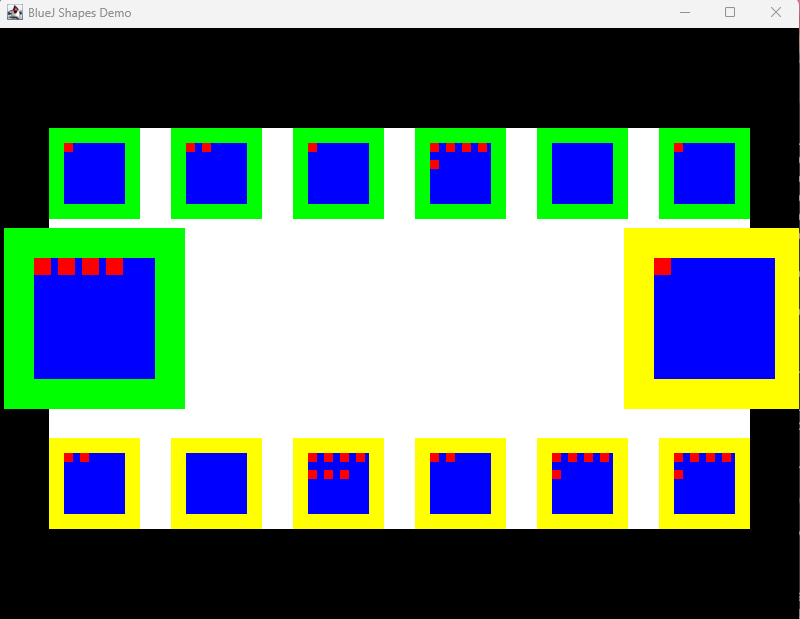


Así teniendo en cuenta el primer mini ciclo se observa la manera en la que se reinicia el juego.

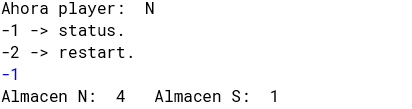
Al momento de finalizar el juego, van a salir mensajes de felicitación teniendo en cuenta quien salió vencedor si el usuario ingresa –3 nuevamente se va a reiniciar el juego volviendo a empezar.

**MINI-CICLO 3**

Para este mini-cilo queremos mostrar el método de status game, esto lo implementamos en una parte del start game y lo vamos a utilizar mediante consola al ingresar el número -1.

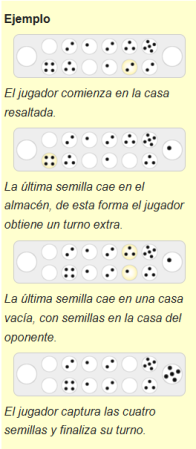


A lo largo del juego podemos llegar a situaciones de este estilo, por lo que podemos consultar el estado del juego.



MINI-CICLO4:

Ya el juego funciona segun las reglas:



1. Indiquen las extensiones necesarias para reutilizar la clase de los huecos y el paquete shapes. Expliquen.

La manera en la que reutilizamos la clase Pit y así mismo shapes fue mediante la creación de objetos para poder acceder a todos los métodos que habíamos creado anteriormente, por otra parte, en shapes utilizamos get y set para acceder a información de métodos privados.

**BONO. Nuevos requisitos funcionales. Kalah.**

El objetivo de este trabajo es extender la mini-aplicación Kalah.

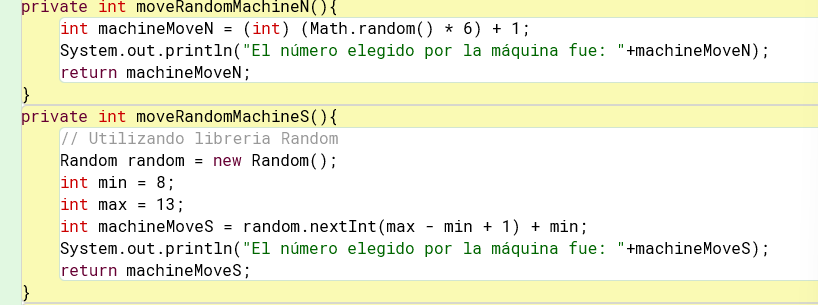
Nuevos requisitos funcionales

- Hacer un movimiento (la máquina decide). Explique la estrategia.

- Deshacer el último movimiento

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

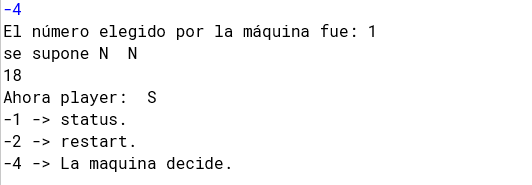
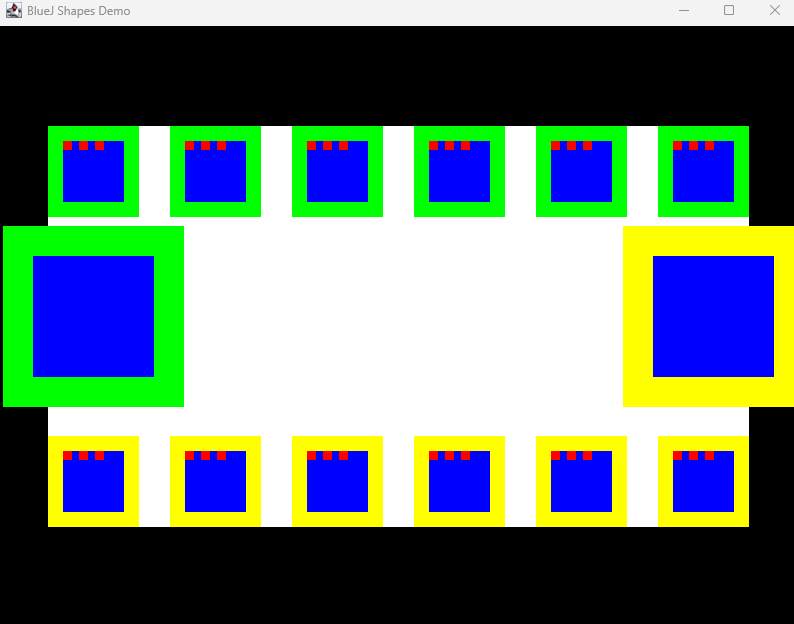
Para el primer método hacer un movimiento(la máquina decide) utilizamos la libreria Random y Math para poder generar el número aleatorio como explicamos anteriormente en el mini-ciclo.

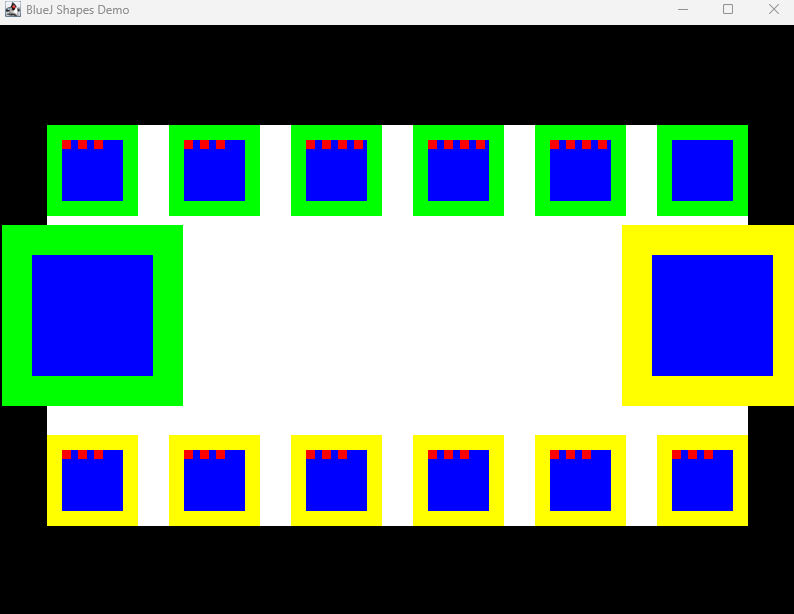


2. Implementen los nuevos métodos . Al final de cada método realicen una prueba

indicando su propósito. Capturen las pantallas relevantes.

Para el primer método hacer un movimiento(la máquina decide) utilizamos la libreria Random y Math para poder generar el número aleatorio como explicamos anteriormente en el mini-ciclo.





RETROSPECTIVA

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

Nombre)

Tulio Riaño Sánchez: 20 horas aprox

Andrés Cardozo: 20 horas aprox

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

El estado actual del laboratorio es completo, resaltando la comuniación y el compromiso se logró llevar a este estado.

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

Las dos prácticas XP fueron demasiado utiles ya que nos permitía definir como realizar el respectivo laboratorio, pero encontramos la más relevante la del trabajo a par ya que mediante la comunicación logramos desarrollar las respectivas ideas para cualquier punto.

4. ¿Cuál consideran fue el mayor logro? ¿Por qué?

Aprender diferentes maneras y métodos en las que podemos desarrollar una especie de juego mediante java.

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

Al momento de realizar kalah encontramos que su complejidad era la manera en la que nos movimos en sentido antihorario, esto dificultaba la manera de pensar como hacerlo de manera eficiente o correcta.

6. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los

Resultados?

De nuevo resaltamos la comunicación y disposición para desarrollar el laboratorio, así mismo mantenemos nuestro compromiso por la ayuda mutua que se evidencia en el desempeño.

7. ¿Qué referencias usaron? ¿Cuál fue la más útil? Incluyan citas con estándares

Adecuados.

Se adjuntan las respectivas citas bibliográficas, todas fueron demasiadas utiles ya que al ser un nuevo lenguaje de programación, desconocíamos sintaxis y formas logicas de resolver diferentes problemas.

Referencias Bibliograficas

* <https://www.geeksforgeeks.org/java-joptionpane/>
* <https://www.w3schools.com/java/ref_string_equalsignorecase.asp>
* <https://stackoverflow.com/questions/3120922/joptionpane-input-to-int>
* <https://es.wikipedia.org/wiki/Kalah>
* <https://www.w3schools.com/java/java_user_input.asp>
* <https://es.stackoverflow.com/questions/475600/unir-dos-listas-de-tipo-list-en-java>
* <https://buenasintencions.blogspot.com/2011/10/la-clase-joptionpane.html>
* <https://www.tempmail.us.com/es/java/generar-enteros-aleatorios-en-java-dentro-de-un-rango>