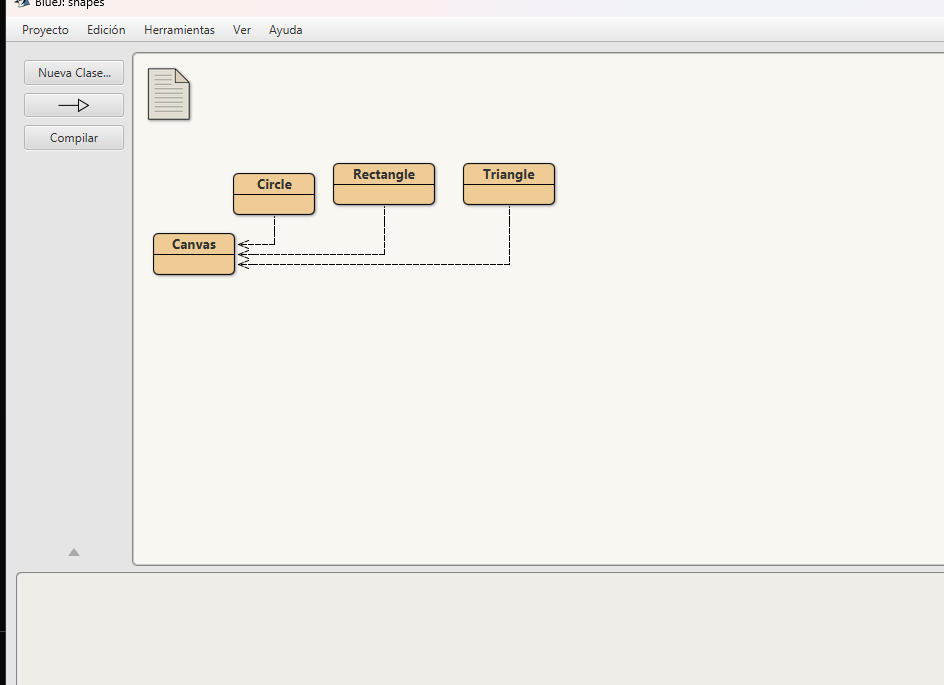
SHAPES

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, son utilizadas en la clase 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? Vertices, height, width, xPosition, yPosition, color, isVisible.

(c) ¿Cuántos métodos ofrece la clase Triangle? La clase Triangle ofrece 14 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 metodos y atributos que son delcarados como privados.

(b) ¿por qué debe ser así?

FALTA

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

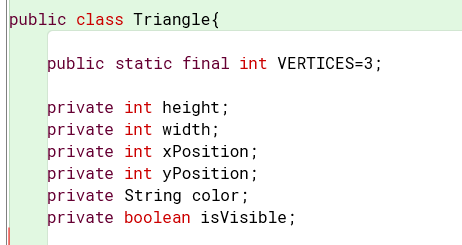
(a) ¿Qué significa que sea public?

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 veces.

(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 triangulo equilatero, 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

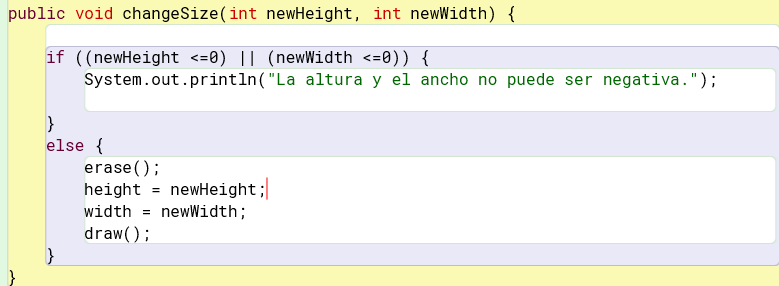
Si no, tocaria especificar cual es el valor de width:

(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 proposito podria ser crear dibujos apartir de ciertas figuras geométricas.

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? 4 Objetos.

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

El objeto de la clase Canvas se crea diferente ya que

FALTA

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

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

FALTA

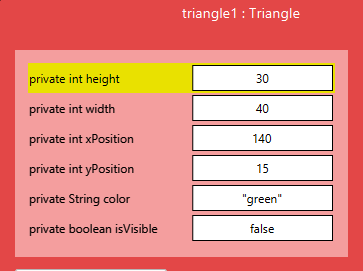
(b) ¿Qué se busca con la clase que tiene el creador diferente?

FALTA

3. Inspeccionen el estado del objeto: Triangle

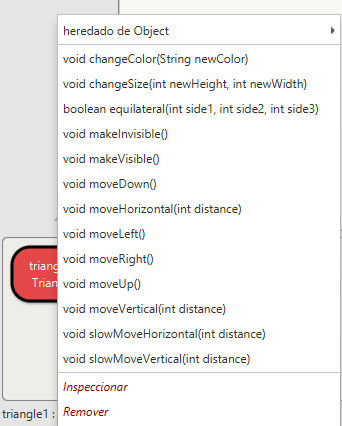
(a) ¿Cuáles son los valores de inicio de todos sus atributos?

(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 son comportamientos privados definidos en la clase.

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

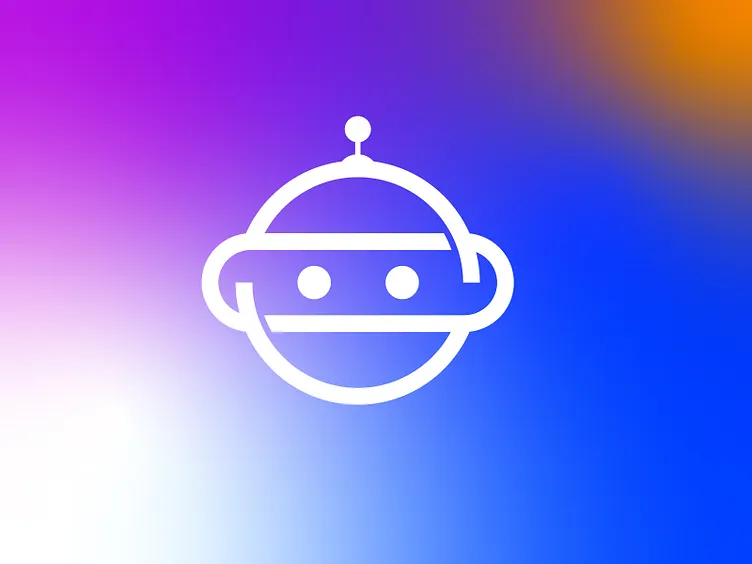
de su chatbot IA favorito.

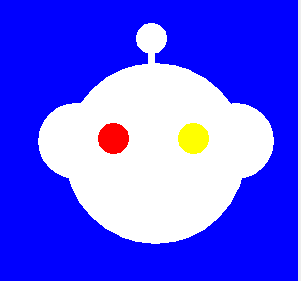
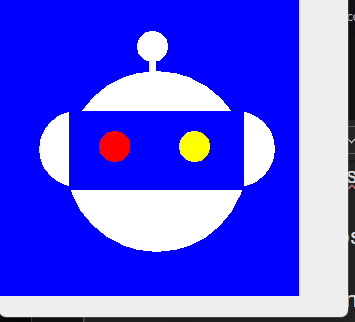
(a) ¿Cuántas y cuáles clases se necesitan? Se necesitan 2 clase, las figuras geometricas definidas: Circle y Rectangle.

(b) ¿Cuántos objetos se usan en total?

9 objetos fueron usados

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





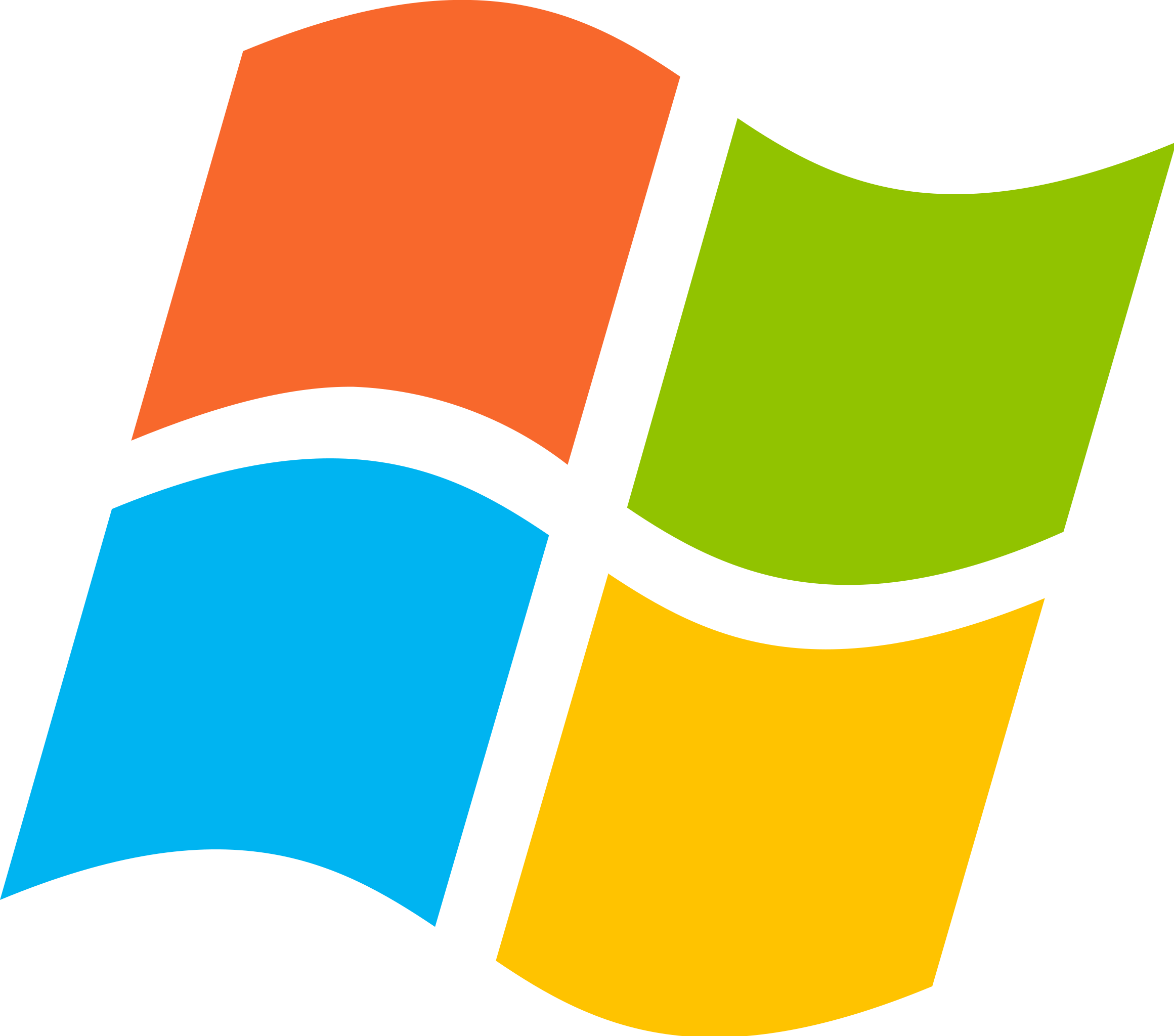
C. Manipulando objetos. Analizando y escribiendo código.



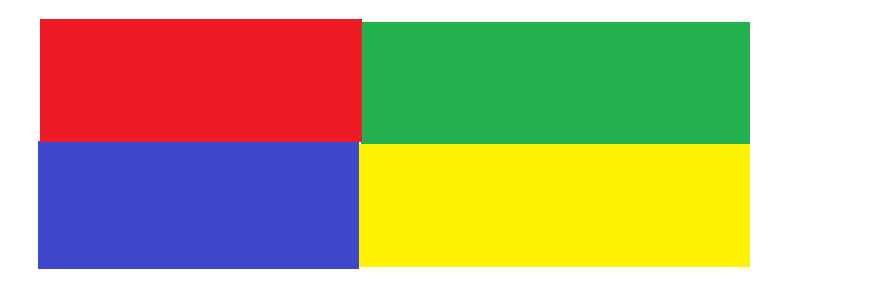
1. Lean el código anterior.

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

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?

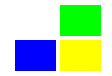
Se ven 3 objetos.

3. Al final,

(e) Expliquen sus respuestas.

Ya que inicializamos tres objetos, rojo, verde y azul, al momento de poner blue=red hace que el “globo” del objeto rojo tenga dos referencias.

(f) Capturen la pantalla.

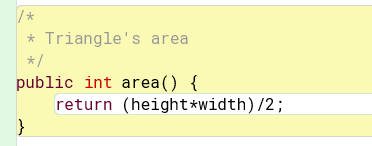


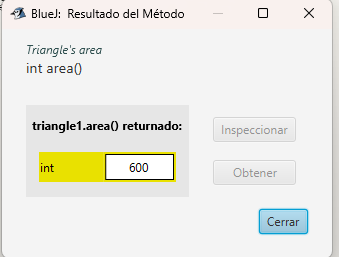
4. Compare figura pintada en 1. con la figura capturada en 2. , (a) ¿son iguales? (b) ¿por qué?

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.

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.

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.