

LAB 01 PROGRAMACIÓN ORIENTADA A OBJETOS
CONSTRUCCIÓN.CLASES Y OBJETOS

PRESENTADO A:

MARIA IRMA DIAZ ROZO

PRESENTADO POR:

CAROLINA CEPEDA VALENCIA

JUANITA RUBIANO FERNANDEZ

ESCUELA COLOMBIANA DE INGENIERIA JULIO GARAVITO

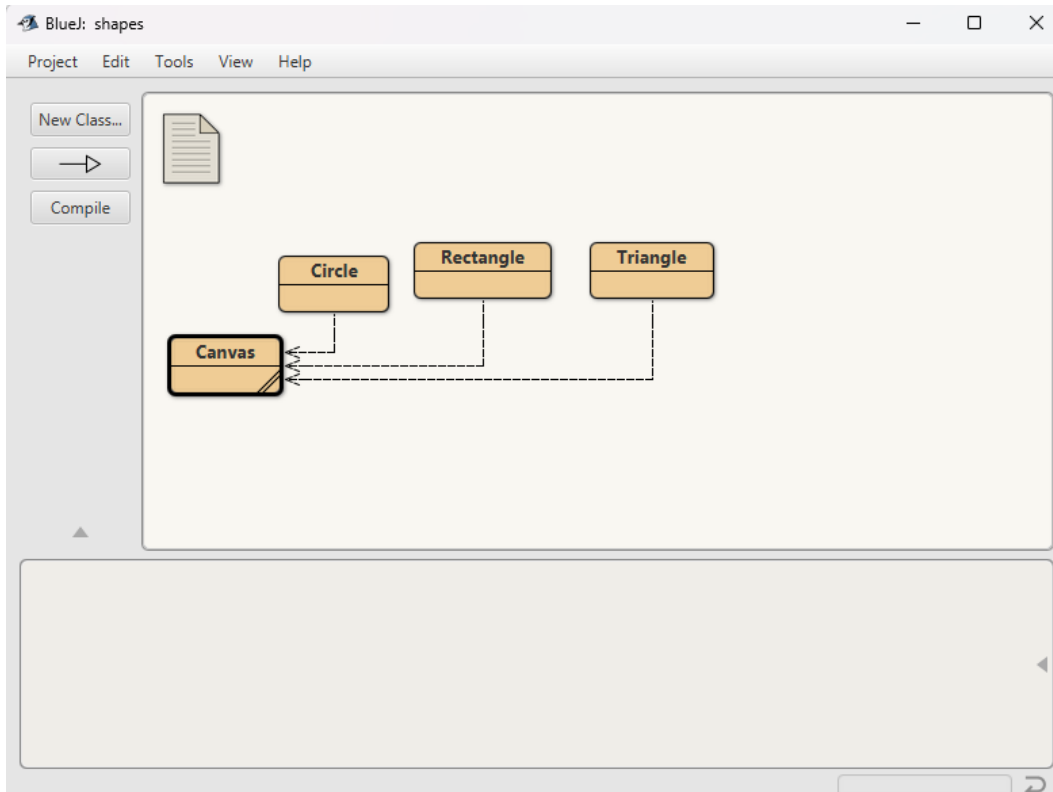
BOGOTÁ. D.C

2025-1

SHAPES

A. 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 ábralo en Bluej. Capturen 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?

- Canvas
- Circle
- Rectangle
- Triangle

b ¿Qué relaciones existen entre ellas?

Las clases, Circle Rectangle y Triangle dependen de la clase Canvas.

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?

El paquete shapes cuenta con la clase canvas, circle, triangle y rectangle.

b. ¿Qué atributos tiene la clase Triangle?

Dentro de la documentación solo sale el atributo vértices ya que este público.

c ¿Cuántos métodos ofrece la clase Triangle?

La clase Triangle ofrece 12 métodos.

Method Summary	
void	<code>changeColor(String newColor)</code> Change the color.
void	<code>changeSize(int newHeight, int newWidth)</code> Change the size to the new size
void	<code>makeInvisible()</code> Make this triangle invisible.
void	<code>makeVisible()</code> Make this triangle visible.
void	<code>moveDown()</code> Move the triangle a few pixels down.
void	<code>moveHorizontal(int distance)</code> Move the triangle horizontally.
void	<code>moveLeft()</code> Move the triangle a few pixels to the left.
void	<code>moveRight()</code> Move the triangle a few pixels to the right.
void	<code>moveUp()</code> Move the triangle a few pixels up.
void	<code>moveVertical(int distance)</code> Move the triangle vertically.
void	<code>slowMoveHorizontal(int distance)</code> Slowly move the triangle horizontally.
void	<code>slowMoveVertical(int distance)</code> Slowly move the triangle vertically.

d ¿Qué atributos determinan el tamaño de un triangle?

Lo determinan los enteros, Height y Width.

e. ¿Cuales métodos ofrece la clase Triangle para cambiar su tamaño?

```
void | changeSize(int newHeight, int newWidth)  
      Change the size to the new 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?

La clase Triangle tiene realmente 7 atributos.

```
public class Triangle{  
  
    public static int VERTICES=3;  
  
    private int height;  
    private int width;  
    private int xPosition;  
    private int yPosition;  
    private String color;  
    private boolean isVisible;
```

b. ¿Quiénes pueden usar los atributos públicos?

Cualquier usuario que haga uso del programa.

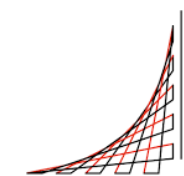
c. ¿Cuántos métodos tiene en total?

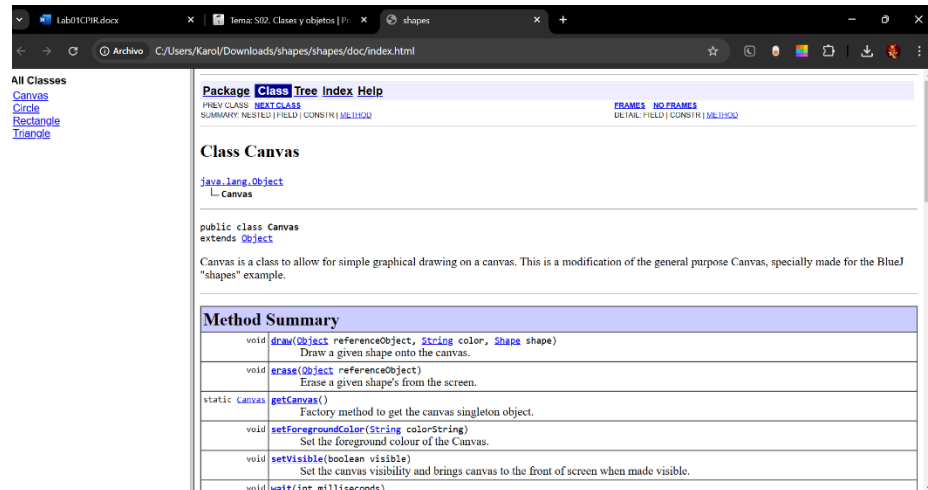
La clase Triangle cuenta con 15 métodos en total. Contando el constructor.

d. ¿Quiénes usan los métodos privados?

Los programadores

e. Desde el editor consulte la documentación. Capture pantalla.





5. Comparando la documentación con el código.

a. ¿Qué no se ve en la documentación?

Los atributos que no tienen visibilidad porque su modificador de acceso es privado.

b. ¿por qué debe ser así?

Al ser privado no debe estar en la documentación ya que esta es de acceso público.

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

a. ¿Qué significa que sea public?

Esto quiere decir que cualquier clase que este dentro del mismo paquete o no, puede hacer uso de ese atributo y si deseas cambiar su valor.

b. ¿Qué significa que sea static?

Que no se necesita crear algún objeto para poderla usar

c. ¿Qué significaría que fuera final?

Si fuera final significaría que, una vez asignado su valor, no se puede ser cambiado.

d. ¿Debe serlo? Actualícelo

```
public static final int VERTICES=3;
```

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

a. ¿Qué se está indicando al decir que es int?

Se indica que la variable es de tipo número entero y puede almacenar valores en el rango de: (-2^{31}) hasta $(2^{31}-1)$.

b. si fuera byte, ¿Cuál sería el área rectángulo más grande posible?

Si fuera byte, la mayor área posible sería: $(127)^2$ debido a que los bytes van desde -128 a 127 y teniendo en cuenta que el área de un rectángulo se calcula base por altura, se toma el mayor valor de los bytes y se eleva.

c. y ¿si fuera Long?

Si fuera Long sería: $(-2^{63}-1)^2$.

d. ¿Qué restricción adicional deberían tener este atributo?

Que su altura no sea mayor a la altura del canvas.

e. Refactorice el código considerando el inciso (d).

```
public void changeSize(int newHeight, int newWidth) {  
    erase();  
    if(height <= 300){  
        height = newHeight;}  
    if(width <= 300){  
        width = newWidth;}  
    draw();  
}
```

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

Basándonos en la lectura del README el propósito del proyecto es tener un acercamiento ameno a la programación orientada a objetos y java, mediante un entorno que familiar como lo es una versión miniatura de Paint.

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.

b. ¿Cuántos objetos crearon?

Se crearon 4 objetos.

c. ¿Quién se crea de forma diferente? ¿Por qué?

El canvas se crea de forma diferente ya que esta clase fue diseñada para crear un único objeto en donde se colocan los objetos de las clases que dependen de esta.

2. Inspeccionen los creadores (constructores) de cada una de las clases.

a. ¿Cuál es la principal diferencia entre ellos?

Una de las principales diferencias es que la visibilidad del canvas predeterminada es true y tiene una condición relacionada a una variable llamada canvasSingleton que tiene la función de que solo se cree un canvas con la clase.

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

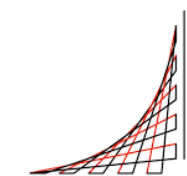
Se busca hacer un lienzo en donde plasmar objetos de las clases de triangle, rectangle and circle.

3. Inspecciones el estado del objeto: Triangle

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

Teniendo en cuenta que todas las medidas numéricas están en pixeles:

- Altura inicial = 30(Height)
- Anchura inicial= 40(Width)
- Posición inicial en x=140 (xPosition)
- Posición inicial en y=15 (yPosition)
- Color inicial= "green" (color)
- Visibilidad = false (isVisible)

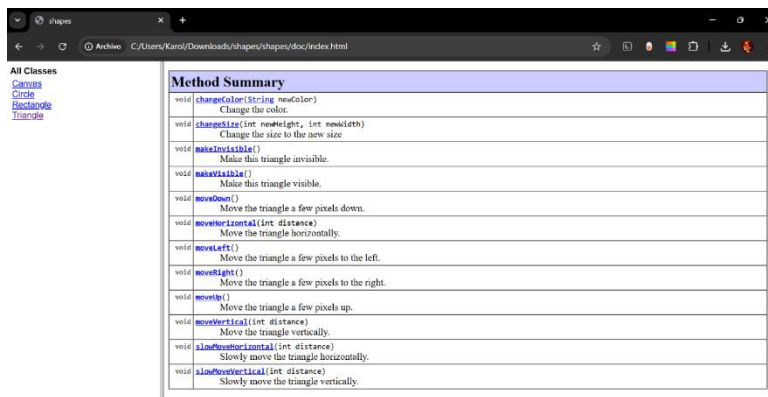


b. Capturen pantalla.

```
public Triangle(){  
    height = 30;  
    width = 40;  
    xPosition = 140;  
    yPosition = 15;  
    color = "green";  
    isVisible = false;  
}
```

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?

No aparecen todos los que están en el código porque los otros comportamientos tienen modificador de acceso privado.

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

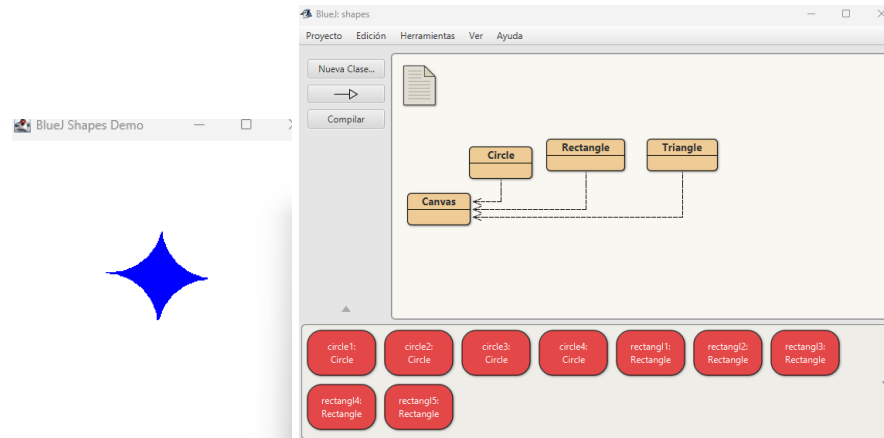
a. ¿Cuántas y cuáles clases se necesitan?

Se necesitaron tres clases, canvas, circle y rectangle.

b. ¿Cuántos objetos se usan en total?

Nueve objetos se usaron en total, 4 círculos y 5 rectángulos.

c. Capturen la pantalla.



d. Incluyan el logo original.



C. Manipulando objetos. Analizando y escribiendo código.

<pre>Rectangle red; Rectangle green; Rectangle blue; //1 red=new Rectangle(); red.changeColor("red"); //2 green=new Rectangle(); green.changeColor("green"); green.moveHorizontal(45); red.makeVisible(); green.makeVisible(); //3</pre>	<pre>blue=red; blue.changeColor("blue"); blue.moveVertical(35); //4 Rectangle yellow=new Rectangle() yellow.changeColor("yellow"); yellow.moveHorizontal(45); yellow.moveVertical(35); //5 blue.makeVisible(); yellow.makeVisible(); //6</pre>
--	--

1. Lean el código anterior.
 - a. ¿cuál creen que es la figura resultante?

Un cuadro en donde hay 4 rectángulos de distintos colores; del lado izquierdo hay un rectángulo rojo y arriba de este un rectángulo azul, de la misma manera en el lado derecho con un rectángulo verde y arriba un rectángulo azul.

- b. Píntenla.



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

- a. ¿cuántas variables existen?

Hay 4 variables.

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

Existen 3 objetos

- c. ¿qué color tiene cada uno de ellos?

Azul, verde y amarillo

- d. ¿cuántos objetos se ven?

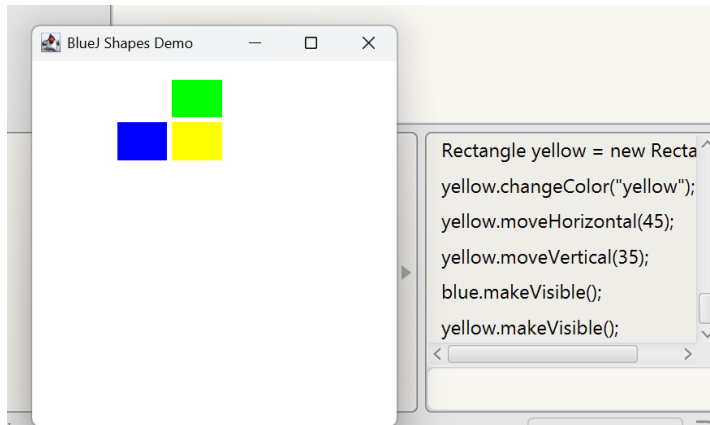
Se ven 4 objetos, contando el canvas.

3. Al final,

- a. Expliquen sus respuestas.

Considerábamos que se hacía una construcción de 4 objetos por qué no tuvimos en cuenta la asignación (blue = red;) .

- b. Capturen la pantalla.



4. Compare figura pintada en 1. con la figura capturada en 2,

a. ¿son iguales?

No

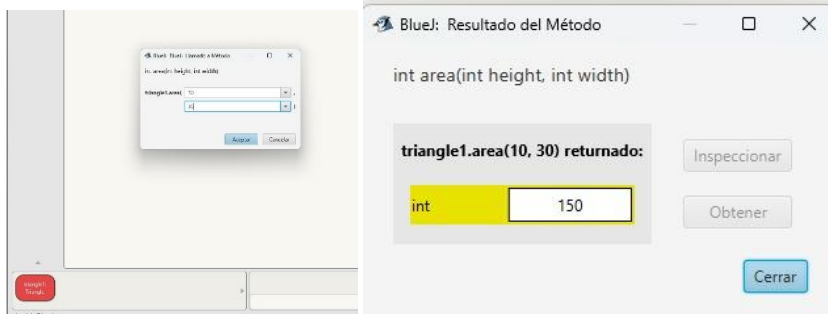
b. ¿Por qué?

En la figura hay un objeto(rectángulo) más de color rojo, esto se debe a que no se tuvo en cuenta la línea en la que a blue se le asigna el rectángulo red.

D. Extendiendo una clase. Triangle.

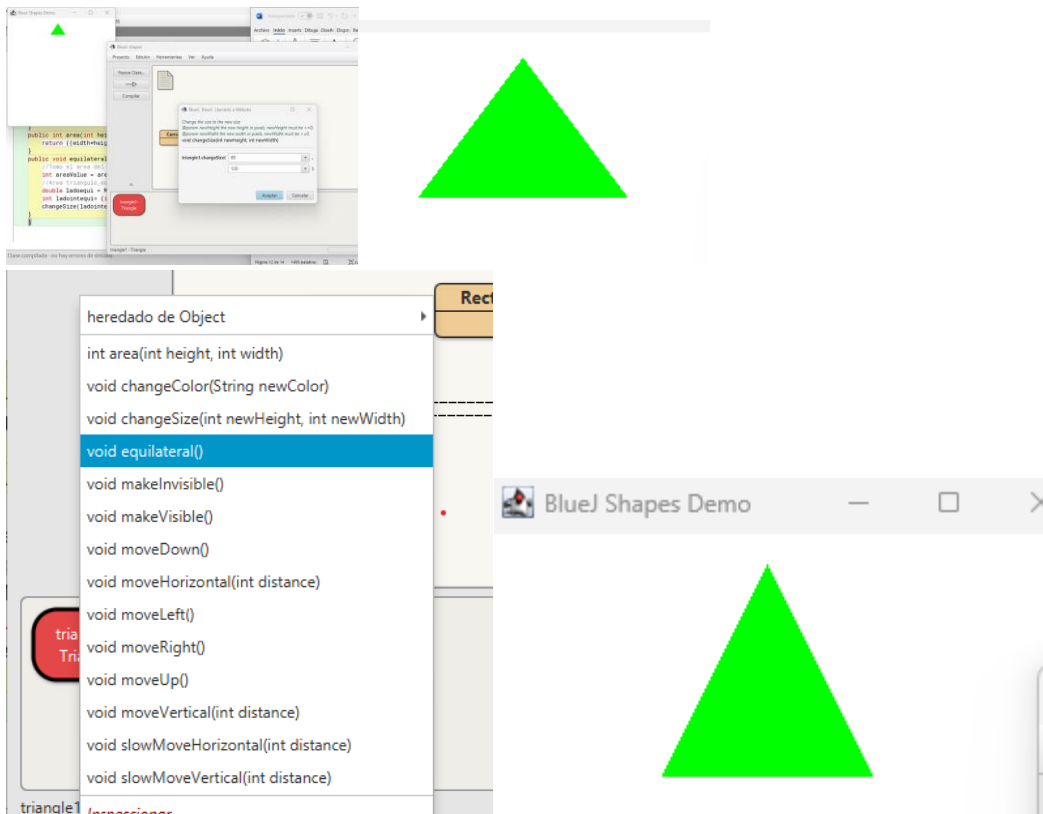
1. Desarrollen en Triangle el método área (). ¡Pruébenlo! Capturen una pantalla.

```
public int area(int height, int width){
    return ((width*height)/2);
}
```



- Desarrollen en Triangle el método equilateral ().
(Transformar el triángulo en un triángulo equilateral de área equivalente)
¡Pruébenlo! Capturen una pantalla.

```
public void equilateral(){
    //Tomo el area del triangulo ya creado y calculado en el metodo area.
    int areaValue = area(height,width);
    //Area triangulo_equilatero=  $\sqrt{4 * \text{área} / \sqrt{3}}$ 
    double ladoequi = Math.sqrt(4*areaValue/Math.sqrt(3));
    int ladointequi= (int) Math.round(ladoequi); //redondeo el numero ded ser necesario
    changeSize(ladointequi,ladointequi);
}
```

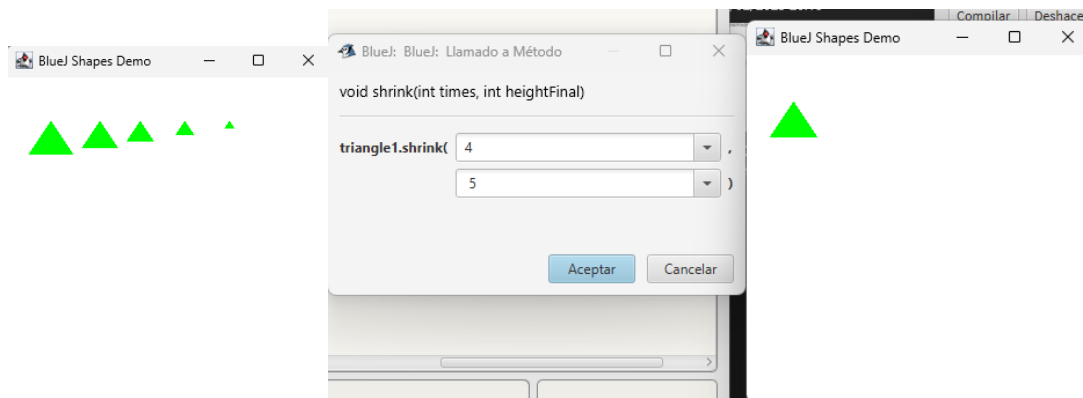


- Desarrollen en Triangle el método shrink (times: int, Height:int)
(Disminuye su tamaño times veces. Hasta llegar a una altura de height) ¡Pruébenlo!
Capturen una pantalla.

Este método fue desarrollado entre las parejas de Avella-Fajardo y Cepeda-Rubiano, en la cual se utilizó una inteligencia artificial para conocer como implementar la impresión de los múltiples triángulos al iterar, la línea de código obtenida de esta fue:

```
"public void createTrianglesRight(int n, int spacing) {
    for (int i = 1; i <= n; i++) {
        Triangle newTriangle = new Triangle(xPosition + (i * (width + spacing)), yPosition);
        newTriangle.makeVisible();
    }
}"
```

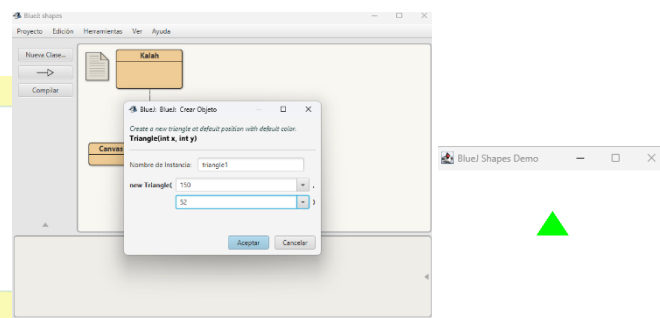
Aunque al momento de implementarla fue adecuada para el problema. ¹



```
public void shrink(int times, int heightFinal) {
    Triangle base = this;
    int dif = height - heightFinal;
    int partes = dif / times;
    int widthFinal = (width * heightFinal) / height;
    int difWidth = width - widthFinal;
    int widthPartes = difWidth / partes;
    Triangle lastTriangle = new Triangle(xPosition + (width + 5), yPosition);
    lastTriangle.changeSize(height - partes, width - widthPartes);
    lastTriangle.makeVisible();
    this.triangles.add(lastTriangle);
    for (int j = 2; j <= times; j++) {
        Triangle newTriangle = new Triangle(xPosition + (j * (width + 1)), yPosition);
        this.triangles.add(newTriangle);
        newTriangle.changeSize(lastTriangle.height - partes, lastTriangle.width - widthPartes);
        lastTriangle = newTriangle;
        newTriangle.makeVisible();
    }
}
```

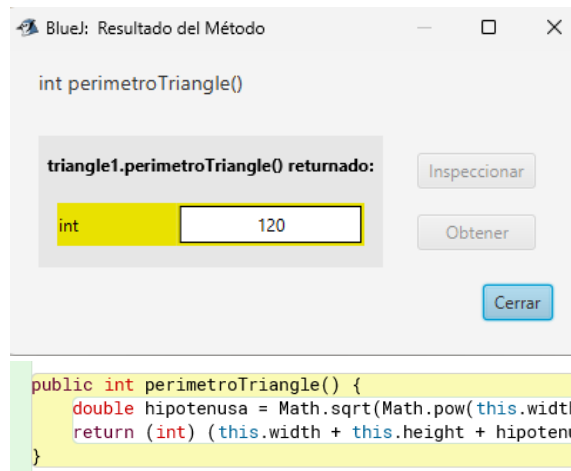
4. Desarrollen en Triangle un nuevo constructor que permita crear un triángulo en una posición específica. ¡Pruébenlo! Capturen una pantalla.

```
public Triangle(int x, int y){
    height = 30;
    width = 40;
    xPosition = x;
    yPosition = y;
    color = "green";
    isVisible = false;
}
```



5. Proponga un nuevo método para esta clase. Desarrollen y prueben el método

Se propone un método para calcular el perímetro del triángulo que se llama PerimetroTriangle.



E. Usando un paquete. Shapes

En este punto vamos a crear huecos con fondos cuadrados para guardar semillas. El diseño gráfico lo definen ustedes.

Los huecos deben ofrecer los siguientes métodos:

Pit	
<pre>+ _(big : boolean) : Pit + putSeeds(seeds : int) : void + removeSeeds(seeds : int) : void + seeds() : int + changeColor(background : String, seeds : String) : void + moveTo(x : int, y : int) : void + makeVisible() : void + makeInvisible() : void</pre>	
<p>El marco de los huecos debe ser cuadrado. El hueco big es el doble de grande que el no big.</p>	<p>Mini-ciclo: 1 <code>_(big) : Pit</code> <code>putSeeds(seeds)</code> <code>removeSeeds(seeds)</code> <code>seeds()</code></p> <p>Mini-ciclo: 2 <code>makeVisible()</code> <code>makeInvisible()</code></p> <p>Mini-ciclo: 3 <code>changeColors(pit, seeds)</code> <code>moveTo(x, y)</code></p>

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

```

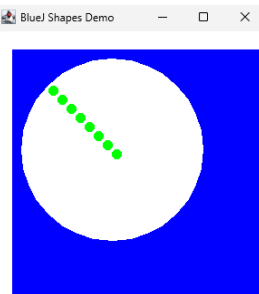
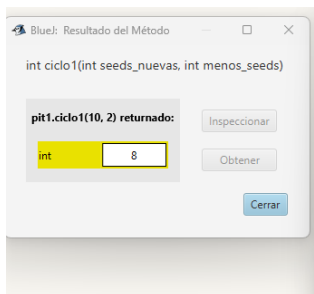
13 public class Pit {
14     private boolean big;
15     private int semillas;
16     private String bg_color;
17     private String s_color;
18     private boolean isVisible;
19     private int xPosition;
20     private int yPosition;
21     private final int width;
22     private final int height;
23     private final int tamaño_s;
24     private int tamaño_h;
25     private Circle semilla;
26     private ArrayList<String> referencias_semillas = new ArrayList<>();
27

```

Se hace la elección de los presentes atributos, los cuales se dividen en varias categorías como la visibilidad, el tamaño, la posición, y color de varios aspectos del pit. En este, hay aspectos tanto para el marco, los huecos y las semillas que se colocan en estos.

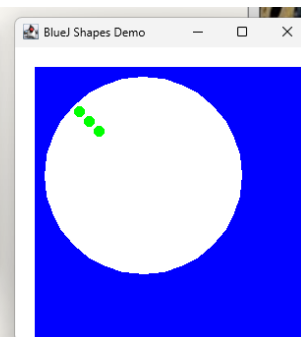
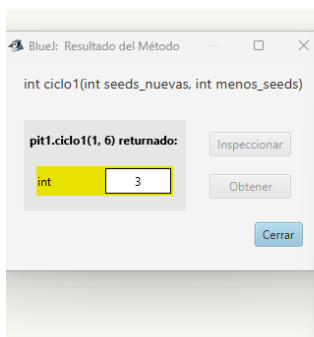
2. Desarrollen la clase considerando los 3 mini-ciclos. Al final de cada mini- ciclo realicen dos pruebas indicando su propósito. Capturen las pantallas relevantes.

- Mini ciclo 1: Aumento y disminución de semillas.



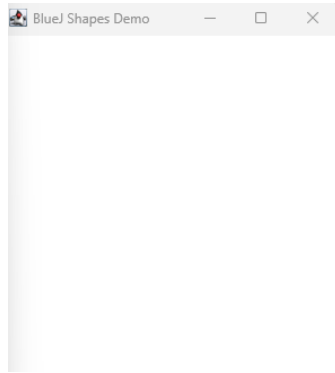
llamando al método cuando hay 0 semillas en el pit.

volviendo a llamar el método después del primer caso prueba.

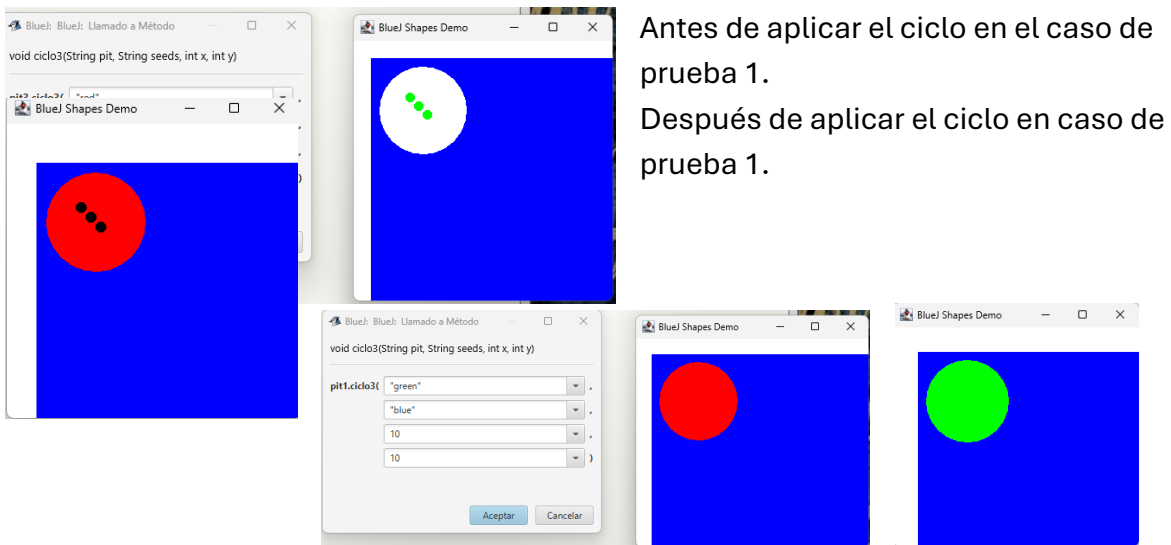


- Mini- ciclo 2:

Al aplicar el método se abre la ventana de canvas en blanco ya que el ciclo hace visible y vuelve a hacer invisible al pit.



- Mini – ciclo 3: Cambiar el color y posición de los elementos del pit.



Antes de aplicar el ciclo en el caso de prueba 1.

Después de aplicar el ciclo en caso de prueba 1.

Antes y después de la segunda prueba.

F. De Python a Java - Se realizaron las encuestas.

BIBLIOGRAFÍA

1. Este método fue desarrollado entre las parejas de Avella-Fajardo y Cepeda-Rubiano, en la cual se utilizó una inteligencia artificial para conocer como implementar la impresión de los múltiples triángulos al iterar, la línea de código obtenida de esta fue:

```
"public void createTrianglesRight(int n, int spacing) {  
    for (int i = 1; i <= n; i++) {  
        Triangle newTriangle = new Triangle(xPosition + (i * (width + spacing)),  
yPosition);  
        newTriangle.makeVisible();  
    }  
}"
```

Aunque al momento de implementarla fue adecuada para el problema.

2. campusMVP. (s. f.). *Variables y tipos de datos en Java: tipos simples, clases y tipos envoltorio o wrapper - campusMVP.es*. campusMVP.es.
<https://www.campusmvp.es/recursos/post/variables-y-tipos-de-datos-en-java-tipos-simples-clases-y-tipos-envoltorio-o-wrapper.aspx>

RETROSPECTIVA

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

* Rubiano – 18 horas.

* Cepeda – 19 horas.

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

Incompleto, faltó la realización del punto F, el punto del (KALAH), porque nos costó adaptarnos a bluej y tuvimos muchos inconvenientes con la función erase de canvas al ser referenciada de la clase Pit a la clase Canvas.

También se tiene un fallo en la clase Pit, pues al hacer makeVisible, se necesita llamar dos veces la función para que se ejecute.

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

La práctica de programación a par, porque nos apoyamos la una en la otra, de este modo solucionar dudas y lograr entender el desarrollo del laboratorio en la misma medida.

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

Lograr que todos los métodos de Triangle fueran funcionales y la construcción de la clase Pit.

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

La función erase de canvas cuando la referenciábamos en la clase Pit y la manera en cómo se enfrentó esta dificultad, fue instanciando un arreglo de objetos, para que la referencia de cada elemento dentro de la clase fuera diferente y se lograra borrar.

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

Consideramos que somos un buen equipo, pero igual manera debemos trabajar en mejorar el modo de trabajo, ser más productivas y tener comunicación asertiva.

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

Se utilizó IAGEN y una referencia de tipos de datos, la más útil fue la IA.