

Nicolás Felipe Bernal Gallo

Juan Daniel Bogotá Fuetes

Desarrollo Orientada a objetos

DOPO LAB

Laboratorio #1 Construcción clases y objetos

21/08/2025

PROFESOR: María Irma Diaz Rozo

# DESARROLLO ORIENTADO A OBJETOS

**Construcción. Clases y objetos. 2025-2** — **Laboratorio 1/6**

# OBJETIVOS

Desarrollar competencias básicas para:

1. Apropiar un paquete revisando el diagrama de clases, la documentación y el código.
2. Crear y manipular un objeto. Extender y crear una clase.
3. Entender el comportamiento básico de memoria en la programación OO.
4. Investigar clases y métodos en el [API](http://docs.oracle.com/javase/8/docs/api/) de Java.
5. Utilizar el entorno de desarrollo de BlueJ.
6. Vivenciar las prácticas XP:

*Planning* — el proyecto se divide en iteraciones; *Coding* — todo el código de producción se programa en parejas.

# ENTREGA

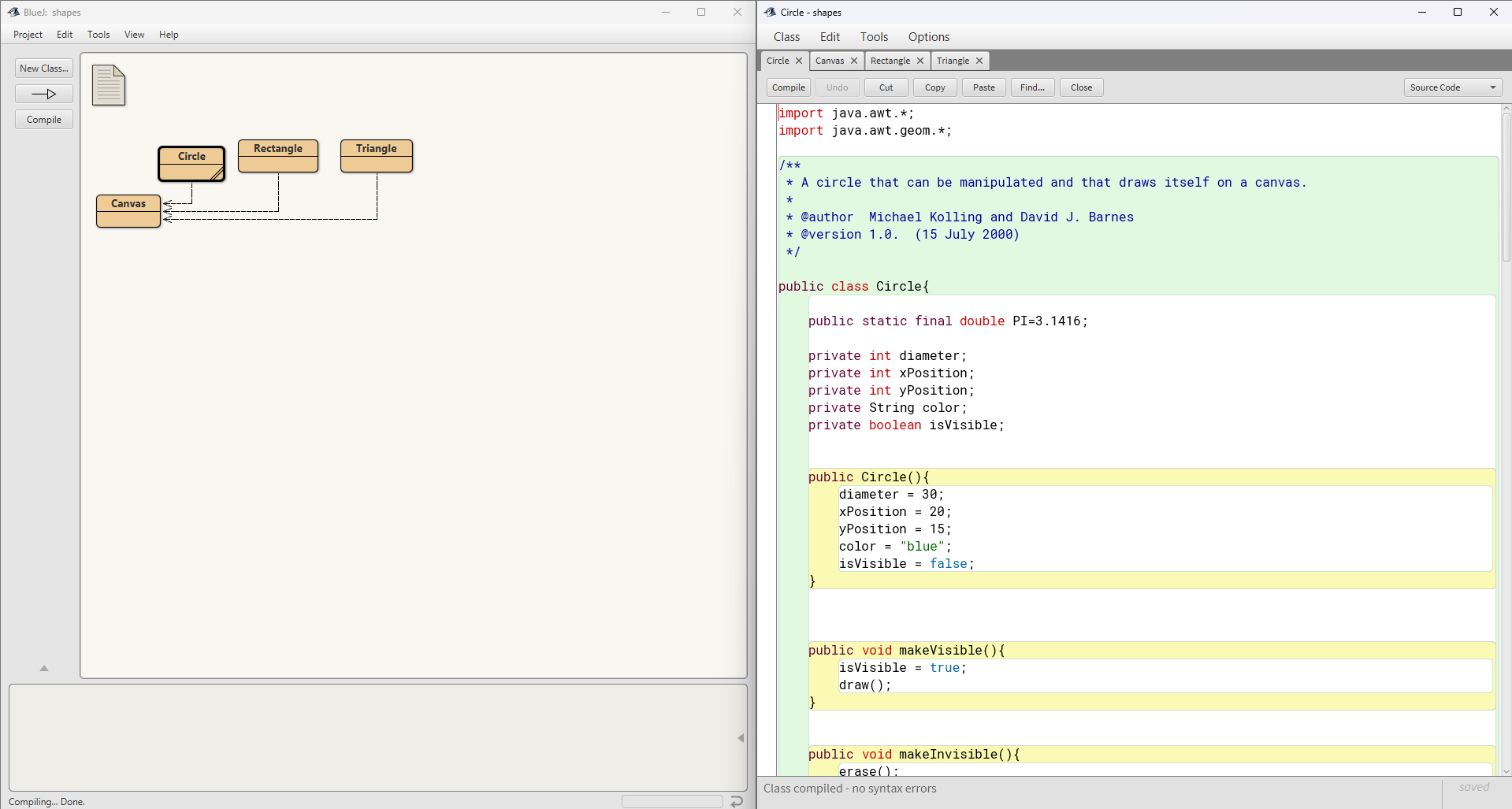
 Incluyan en un archivo .zip los archivos correspondientes al laboratorio. El nombre debe ser los dos apellidos de los miembros del equipo ordenados alfabéticamente.

 Deben publicar el avance al final de la sesión y la versión definitiva en la fecha indicada en los espacios correspondientes.

# SHAPES

**A Conociendo el proyecto shapes.** [**En lab01.doc]**

* 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[1](#_bookmark0). Capturen la pantalla.**



* 1. **El diagrama de clases permite visualizar las clases de un artefacto software y las relaciones entre ellas. Considerando el diagrama de clases de shapes:**

1. **¿Qué clases ofrece?**

Tenemos 4 clases, Canvas, Circle, Rectangle y Triangle.

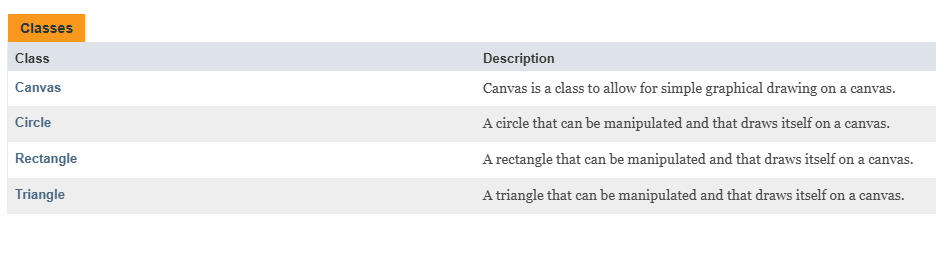
1. **¿Qué relaciones existen entre ellas?**

Circle, Rectangle y Triangle dependen de Canvas.

* 1. **La documentación**[**2**](#_bookmark1) **presenta las clases del proyecto y, en este caso, la especificación de sus componentes públicos. De acuerdo con la documentación generada:**

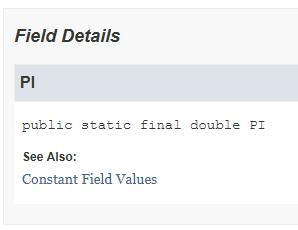
1. **¿Qué clases tiene el paquete shapes?**

Tenemos 4 clases, Canvas, Circle, Rectangle y Triangle.

****

1. **¿Qué atributos tiene la clase Circle?**

Según la documentación de la clase “Circle”, solo tiene un atributo, “PI”.

****

1. **¿Cuántos métodos ofrece la clase Circle?**

La clase “Circle” tiene 12 métodos.

1. **¿Qué atributos determinan el tamaño de un Circle?**

Según la documentación no se encuentra ningún atributo para determinar el tamaño de la clase “Circle”.

1. **¿Cuáles métodos ofrece la clase Circle para cambiar su tamaño?**

Los métodos que ofrece la clase “Circle” para que cambie su tamaño es “changeSize”.

* 1. **En el código de cada clase está el detalle de la implementación. Revisen el código de la clase Circle. Con respecto a los atributos:**

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

Según el código, se tienen 6 atributos: PI, diameter, xPosition, yPosition, color e isVisible.

****

1. **¿Quiénes pueden usar los atributos públicos?**

Los atributos públicos pueden ser utilizados por cualquier clase dentro del mismo paquete shapes.

**Con respecto a los métodos:**

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

Tiene 14 métodos, 12 públicos y 2 privados.

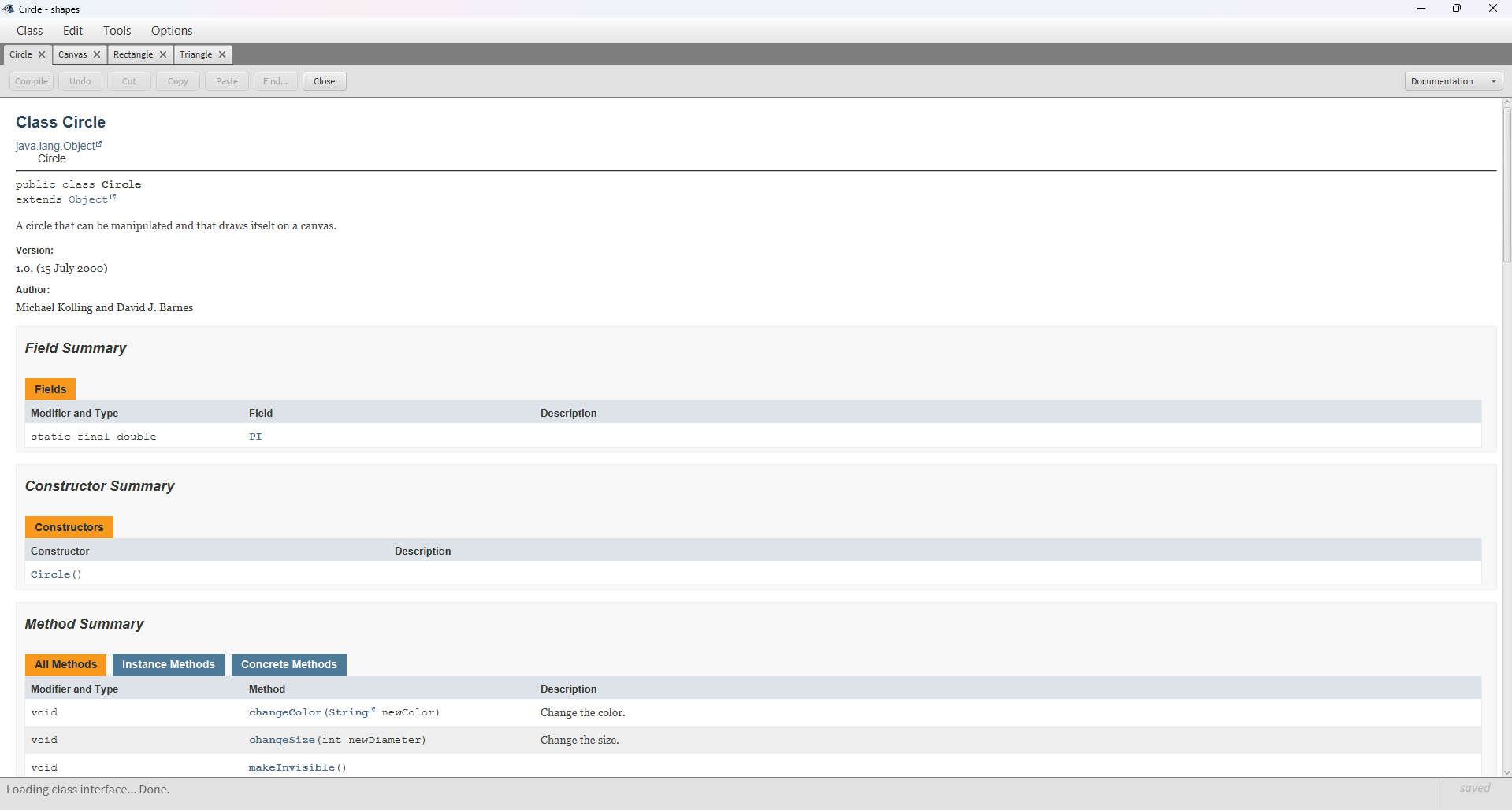
1. **¿Quiénes usan los métodos privados?**

Los métodos privados solo pueden ser utilizados por la misma clase donde están definidos.

Por ejemplo: draw() y erase() en la clase “Circle”

* 1. **Desde el editor, consulte la documentación.**

1. **Capture la pantalla.**

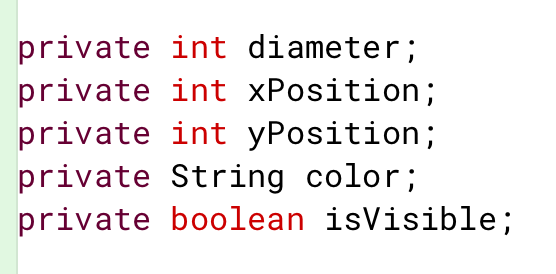


**Comparando la documentación con el código:**

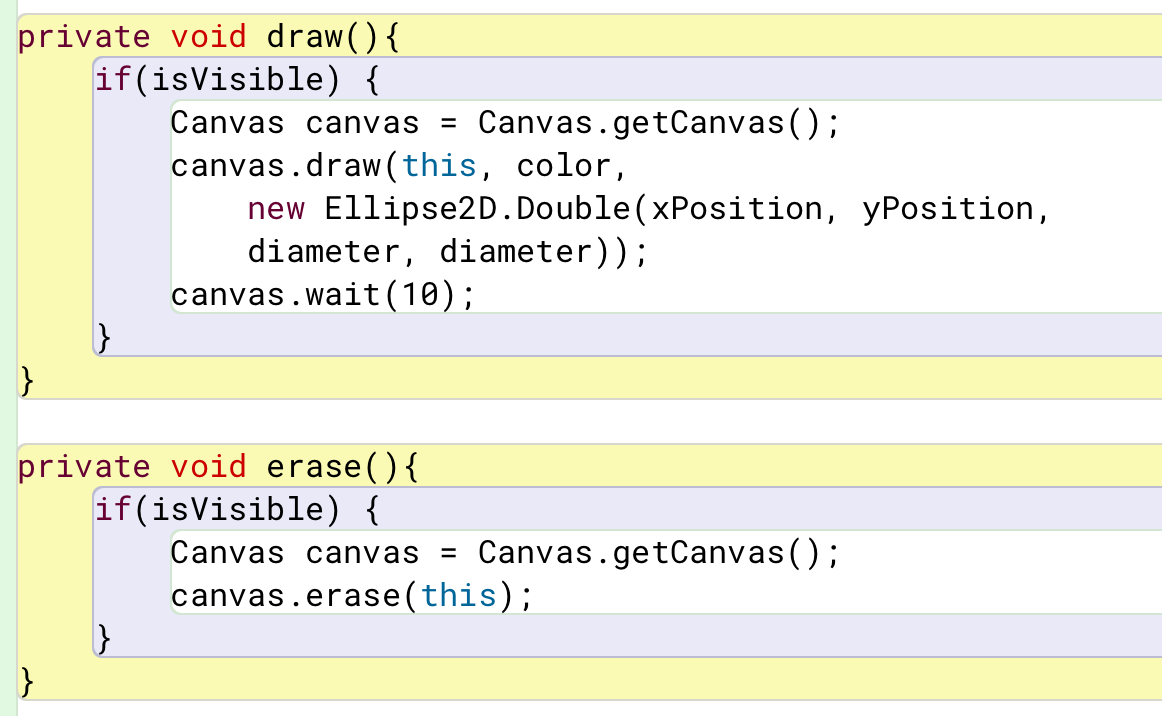
**(b) ¿Qué no se ve en la documentación?**

No se ven todos los métodos y atributos con el modificador de acceso prívate.

En el caso de los atributos no se ven estos:



**En el caso de los métodos no se ven estos:**

****

1. **¿por qué debe ser así?**

Porque se deben proteger los datos y además para que así se evite que otras clases

lleguen a modificar los valores. También se busca asegurar que los datos sean

válidos dentro del código.

* 1. **En el código de la clase Circle, revise el atributo PI.**

1. **¿Qué significa que sea public?**

Esto significa que es accesible desde cualquier clase dentro del programa.

**(b)¿Qué significa que sea static?**

Esto significa que pertenece a la clase en lugar de a una instancia.

1. **¿Qué significa que sea final?**

Esto significaría que no se podría llegar a modificar el atributo después de inicializado. Si como es

un triángulo siempre debe tener tres vértices.

* 1. **En el código de la clase Circle revisen el detalle del tipo del atributo diameter.**

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

Que se trata de que el atributo diameter usa un número entero, no puede usar números decimales.

1. **Si fuera byte, ¿cuál sería el área más grande posible?**

Byte en java significa que puede usar valores entre -128 y 127, entonces como el área de un

Circulo se calcula como:

1. **y ¿si fuera long?**

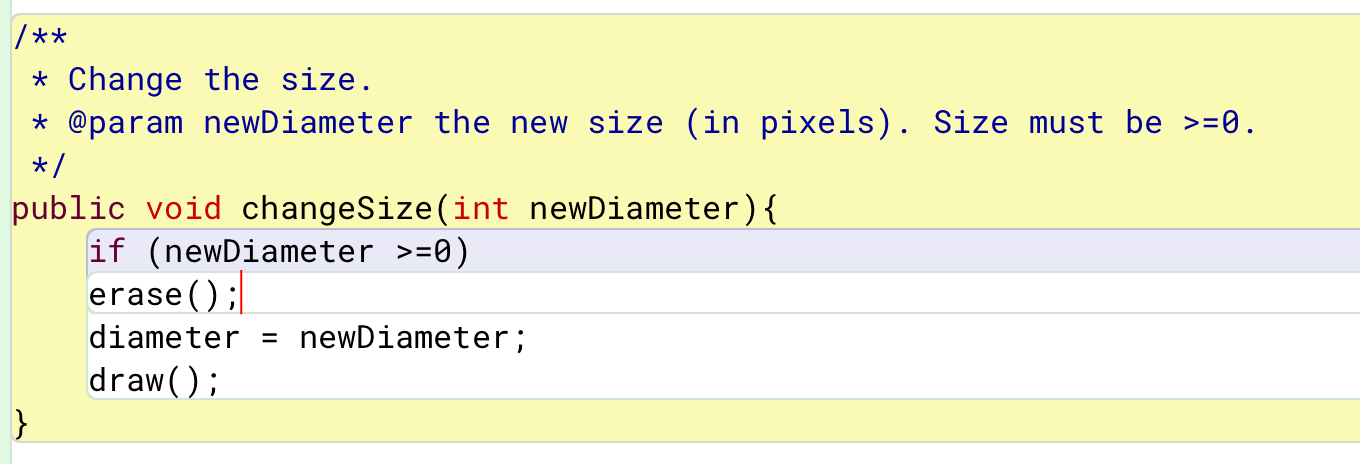
Long en java significa que puede usar valores entre −263 y 263 en el caso de Circulo se calcula como:

1. **¿qué restricción adicional debería tener este atributo?**

Debería tener una restricción de números negativos, ya que el diámetro no puede ser

negativo.

1. **Refactoricen el código considerando**



* 1. **Si creamos 100 círculos,**

1. **¿cuántos PI y cuántos diameter tendríamos?**

1 PI y 100 diameter.

1. **Justifique.**

PI es una constante estática (static final) de la clase: existe una sola copia compartida por todos los objetos, sin importar cuántos círculos creemos.

Diameter es un atributo de instancia: cada círculo tiene su propio diámetro, por eso con 100 círculos hay 100 diameter.

* 1. **¿Cuál dirían es el propósito del proyecto shapes?**

El propósito es dar una introducción practica a la programación orientada a objetos usando BlueJ.

Modelar figuras geométricas para practicar POO: definir clases y objetos, atributos y métodos, diferenciar miembros estáticos vs deinstancia, usar constantes, validación de datos.

1. **Manipulando objetos. Usando un objeto.**
   1. **Creen un objeto de cada una de las clases que lo permitan**[**3**](#_bookmark2)**.**

**(a) ¿Cuántas clases hay?**

Hay 4 clases: “Circle”, “Rectangle”, “Triangle” y “Canvas”

**(b)¿Cuántos objetos crearon?**

Se crearon 4 objetos: triangulo, rectángulo, circulo y canvas.

1. **¿Quién se crea de forma diferente? ¿Por qué?**

El canvas se crea de forma diferente, porque canvas es como la ventana donde se dibujan los

demás objetos. Los demás objetos se crean de manera explícita.

* 1. **Inspeccionen los creadores de cada una de las clases.**

1. **¿Cuál es la principal diferencia entre ellos?**

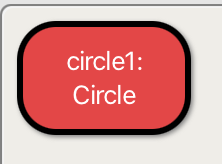
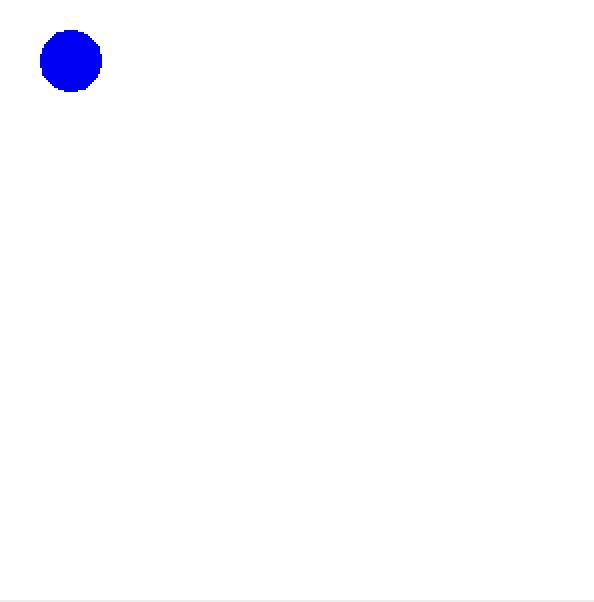
Los objetos triangulo, círculo y rectángulo tienen las propiedades básicas de cada figura, en cambio en el canva no representa una figura, representa un lienzo donde se pueden observar las demás figuras.

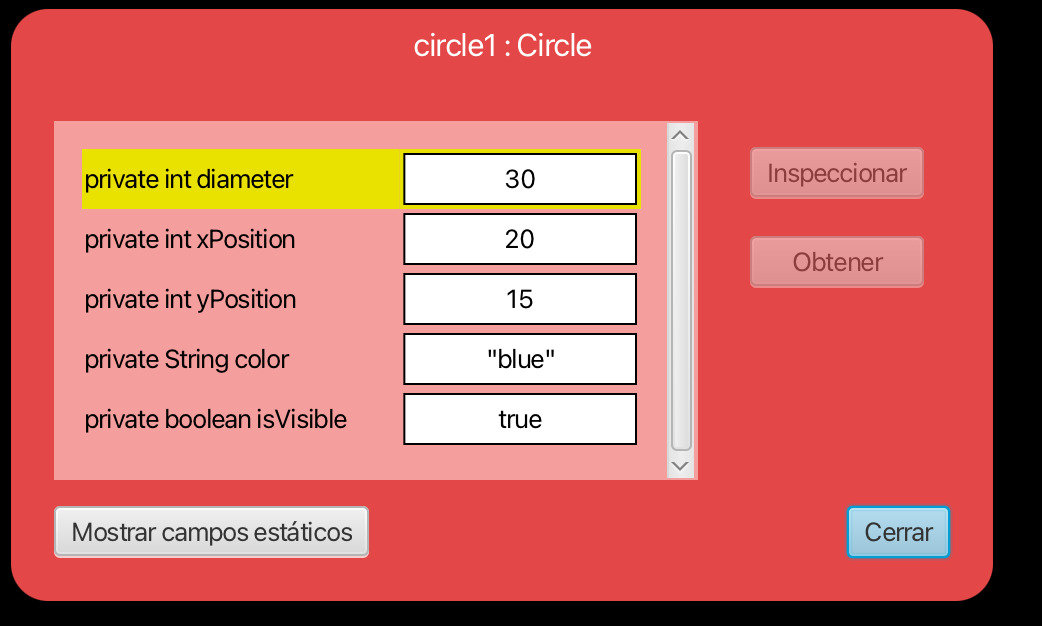
1. **¿Qué se busca con la clase que tiene el creador diferente?**

El canvas busca ser un gestor de dibujo donde se pueden ver los demás objetos.

* 1. **Inspeccionen el estado del objeto :Circle[4](#_bookmark3).**

1. **Capturen la pantalla.**

****

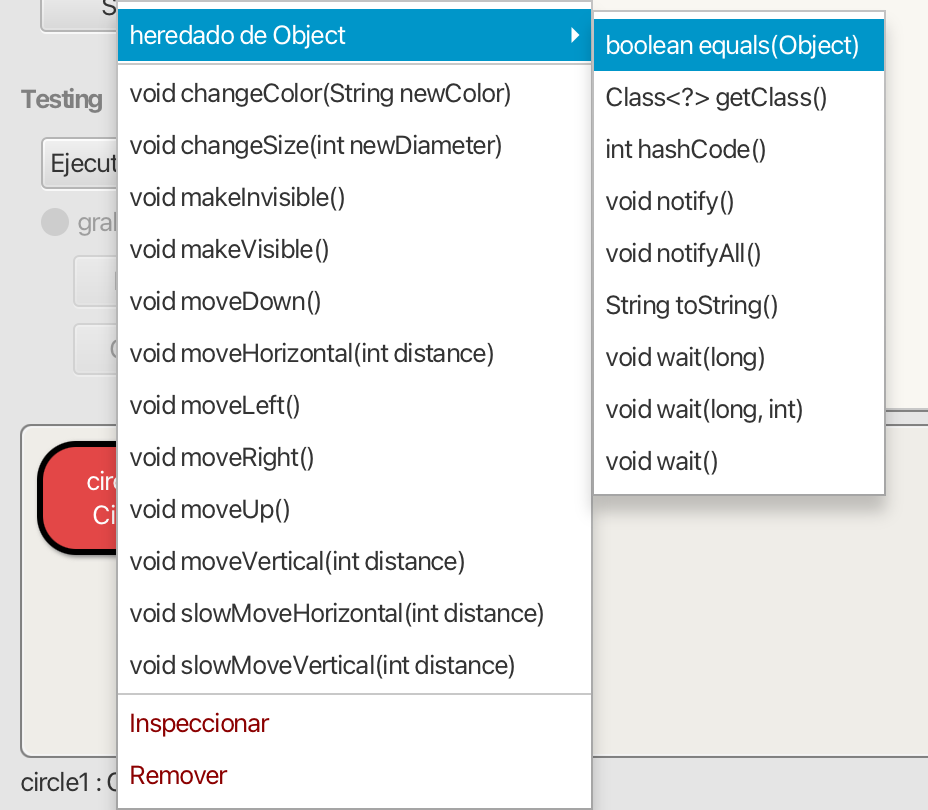
****

1. **¿Cuál es el color inicial?**

El color inicial para el objeto “Circle” es el azul.

* 1. **Inspeccionen el comportamiento que ofrece el objeto: Circle[5](#_bookmark4).**

1. **Capturen la pantalla.**

****

1. **¿Por qué no aparecen todos los que están en el código?**

Porque hay métodos que están con el modificador de acceso private.

* 1. **Construyan, con shapes sin escribir código, una propuesta de la imagen del logo de su domiciliario favorito.**

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

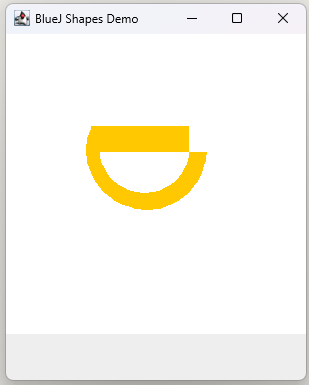
Necesitamos dos clases, sin contar Canvas, Circle y Rectangle.

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

Usamos 5 objetos.

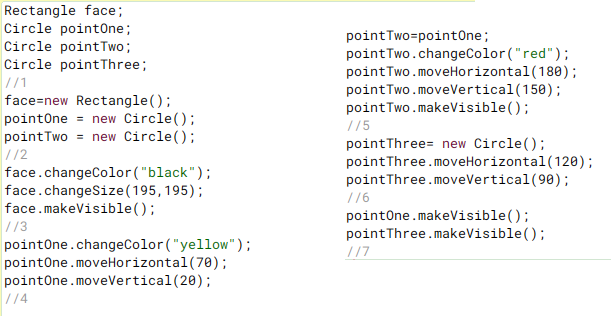


1. **Capturen la pantalla.**



**(d) Incluyan el logo original.**



1. **Manipulando objetos. Analizando y escribiendo código.** [**En lab01.doc]**
   1. **Lean el código anterior.**

**(a) ¿Cuál creen que es la figura resultante?**

Viendo el código por encima, creemos que son dos círculos, uno rojo y otro azul, y un rectángulo negro.

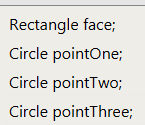
**(b) Píntenla.**



* 1. **Para cada punto señalado:**

**(a) ¿cuántas variables existen?**

Existen 4 variables:



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

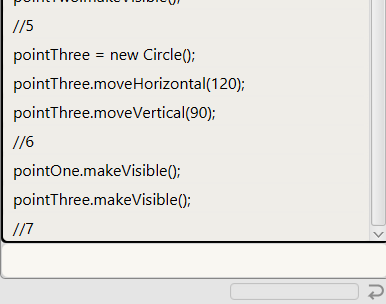
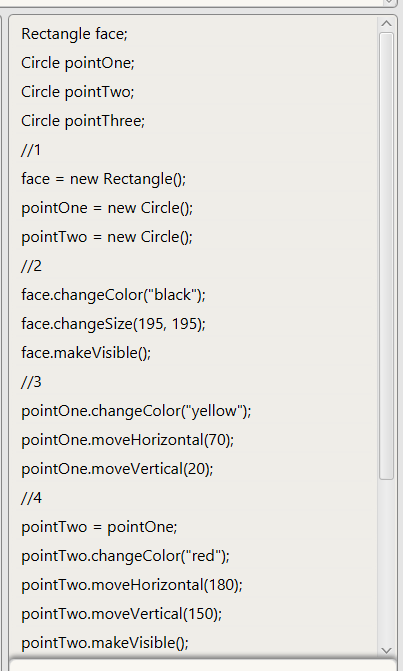
Existen 3 objetos, dos círculos y un rectángulo.

**(c) ¿cuántos objetos se ven?**

Se ven 3 objetos, dos círculos, uno azul y uno rojo y un rectángulo.

* 1. Habiliten la ventana de código en línea[6](#_bookmark5), escriban el código.

1. **Capturen la pantalla.**



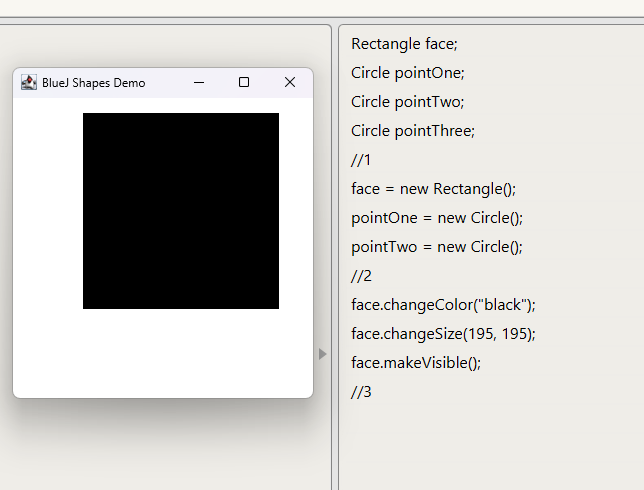
* 1. Compare la figura pintada en 1. con la figura capturada en 3.:

1. **¿son iguales?**

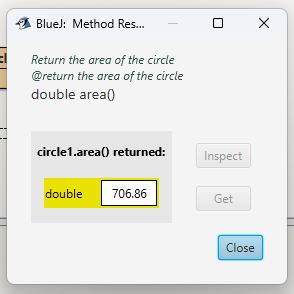
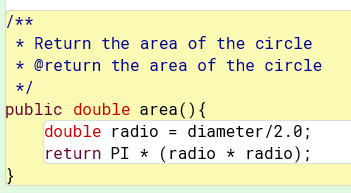
Las figuras capturadas no son iguales.

1. **¿por** **qué?**

Porque en la figura 1 no se pinta nada, ni siquiera se abre el Canvas, solo se declaran variables, en cambio en la figura capturada en 3 se ve un rectángulo negro de tamaño (195,195).

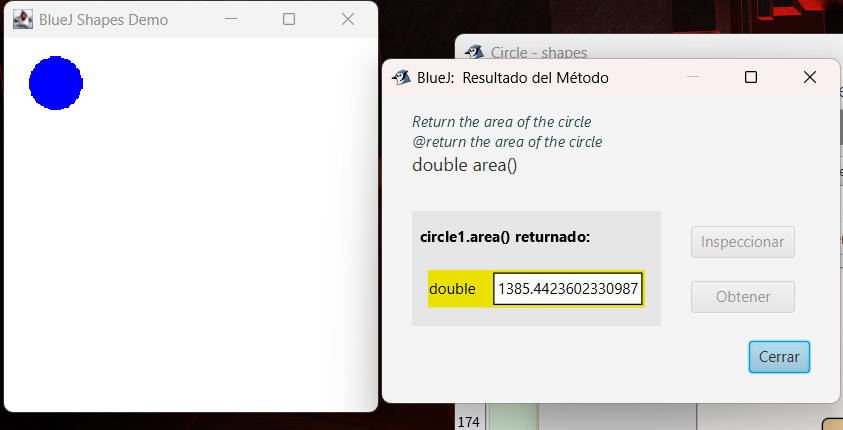
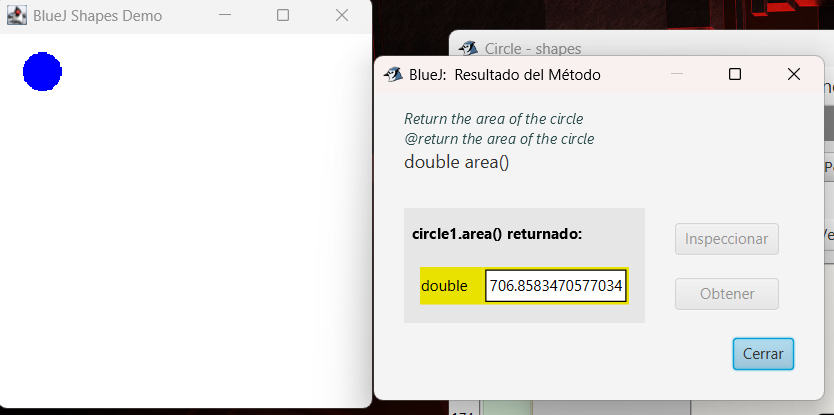
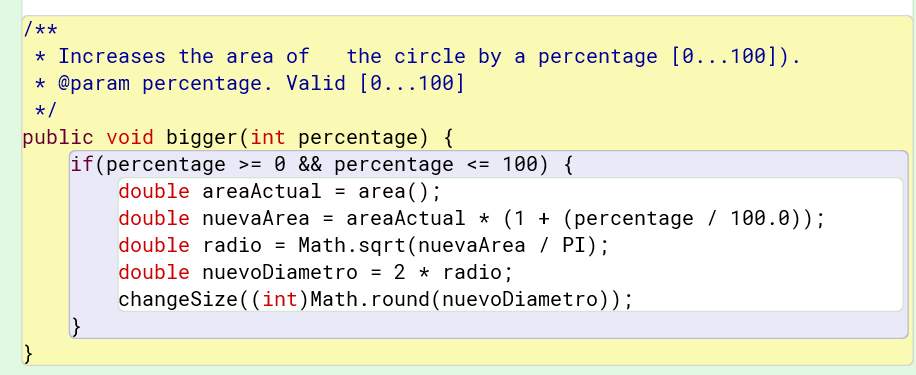


1. **Extendiendo una clase.** Circle**.** [**En lab01.doc]**[**En .java]**
   1. **Desarrollen en Circle el método area() (retorna el área del círculo). ¡Pruébenlo! Capturen una pantalla.**

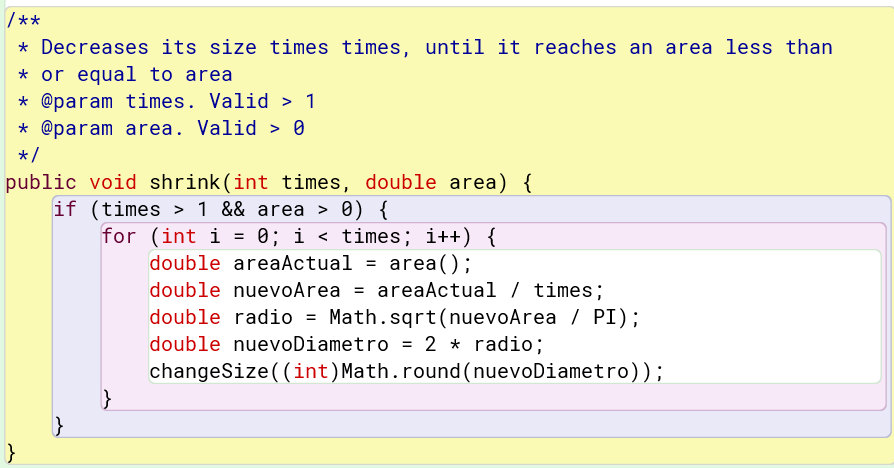


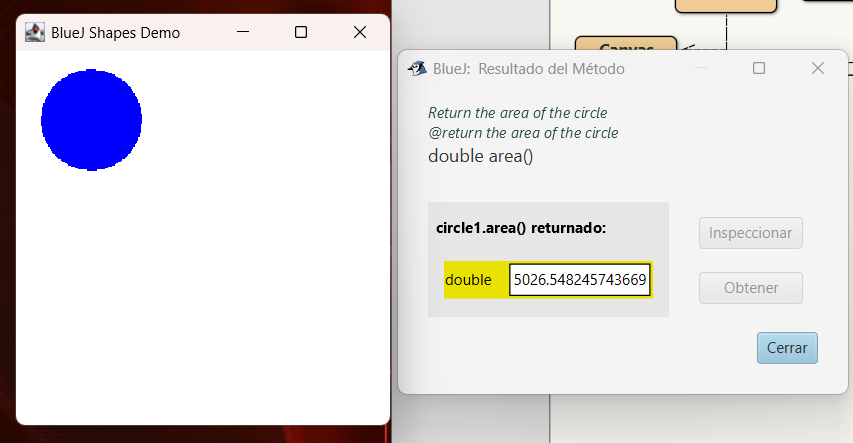
* 1. **Desarrollen en Circle el método bigger(percentage: int) (aumenta el área del círculo un porcentaje [0...100]). ¡Pruébenlo! Capturen dos pantallas.**

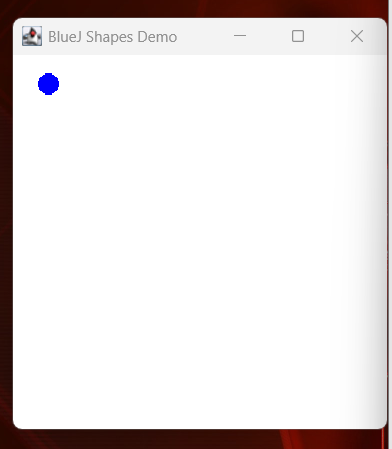
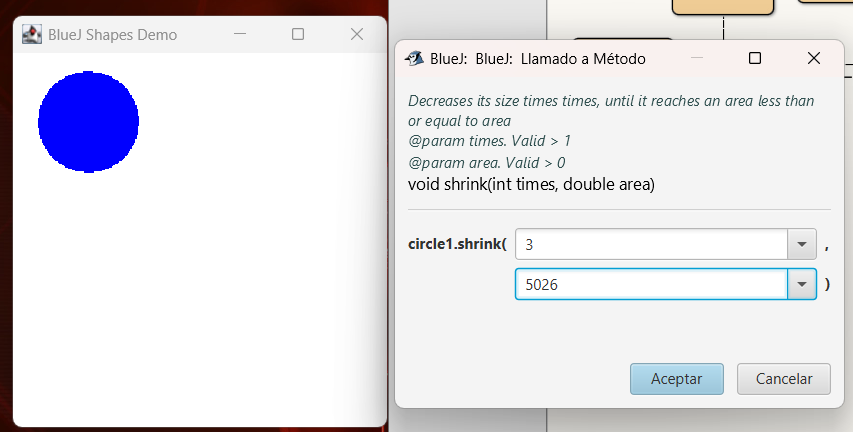
Se aumentó 100%, no es exacto ya que el tamaño se aumenta con el diámetro y el diámetro es tipo int, al no ser tipo double se pierde exactitud.



* 1. Desarrollen en Circle el método shrink(times: int, area: int) (disminuye su tamaño times veces, hasta llegar a un area menor o igual a area). ¡Pruébenlo! Capturen tres pantallas.

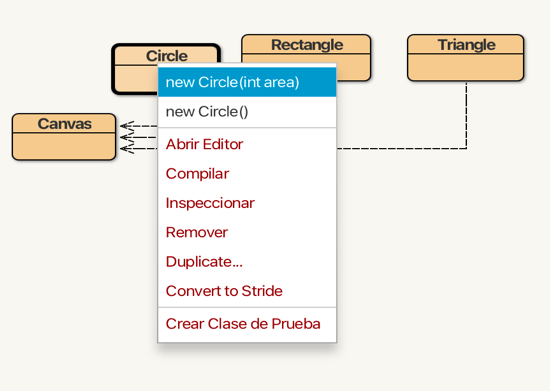
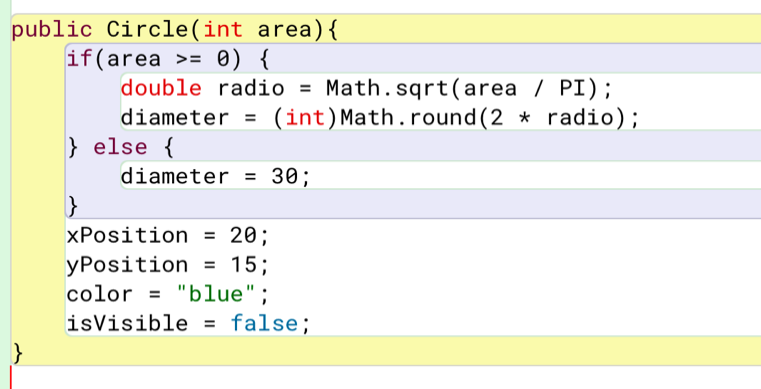


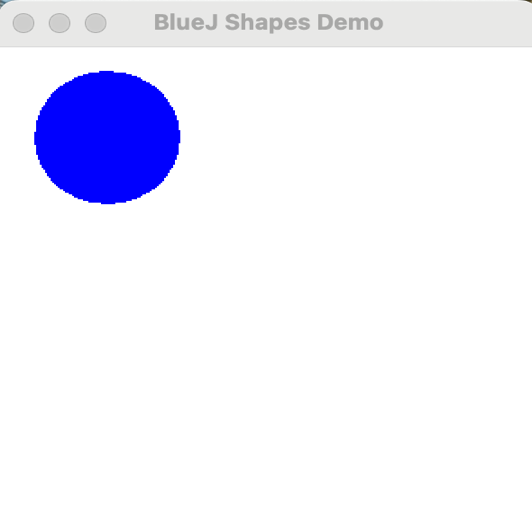
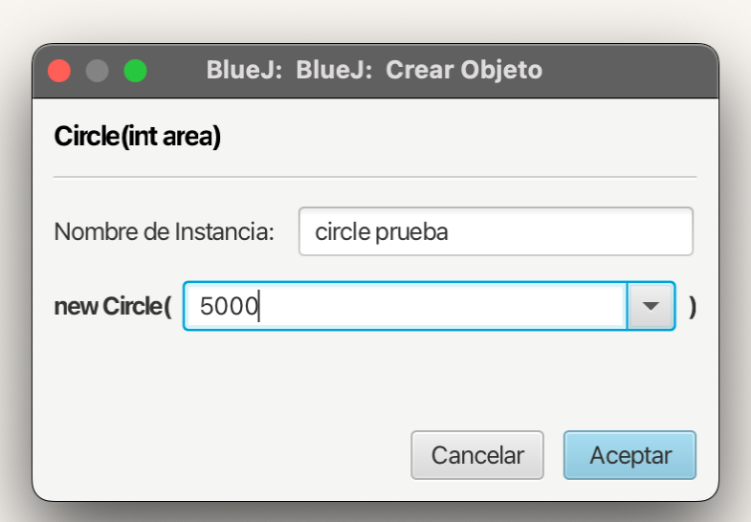




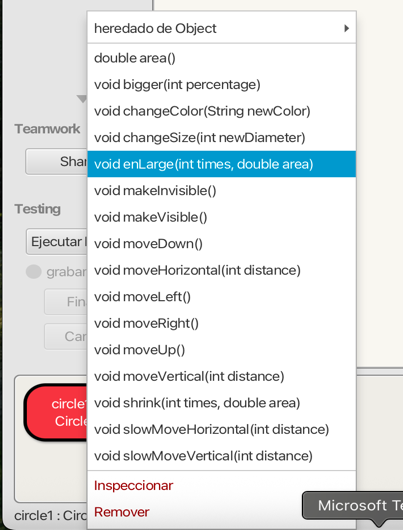
* 1. Desarrollen en Circle un nuevo creador que permita crear un círculo con un área específica.

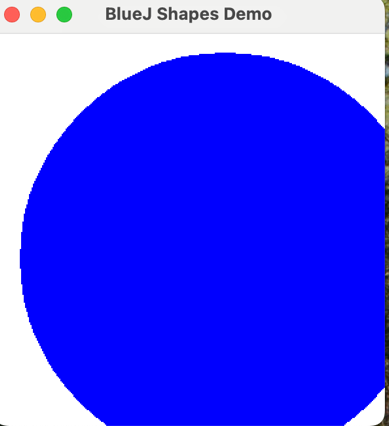
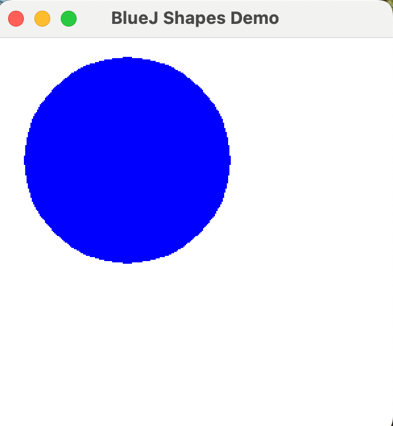
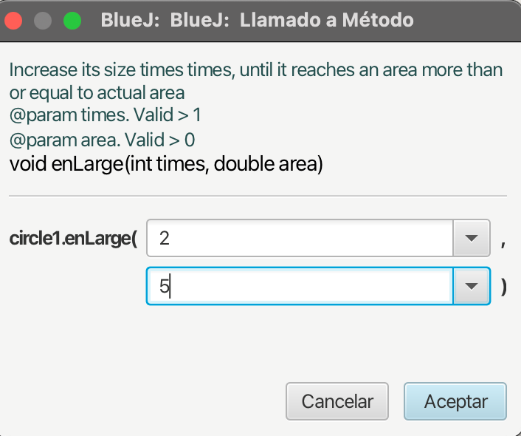
¡Pruébenlo! Capturen una pantalla.



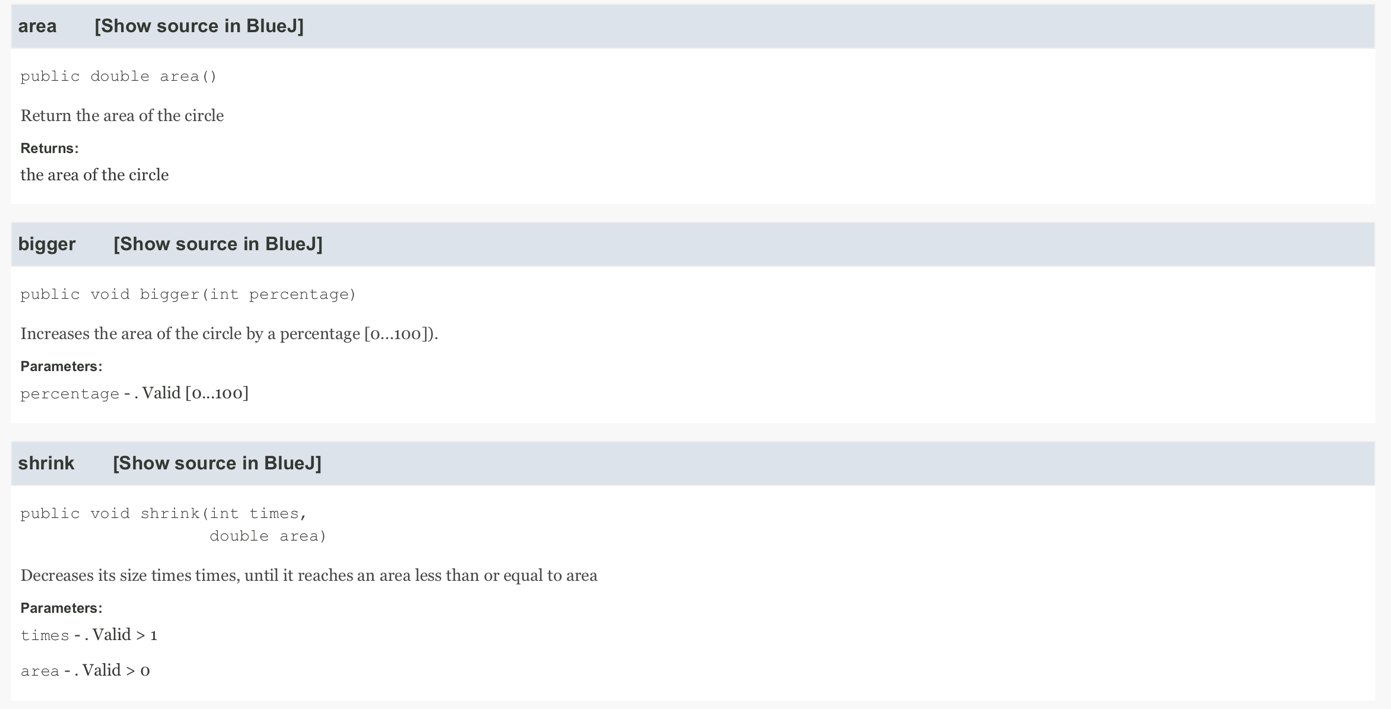


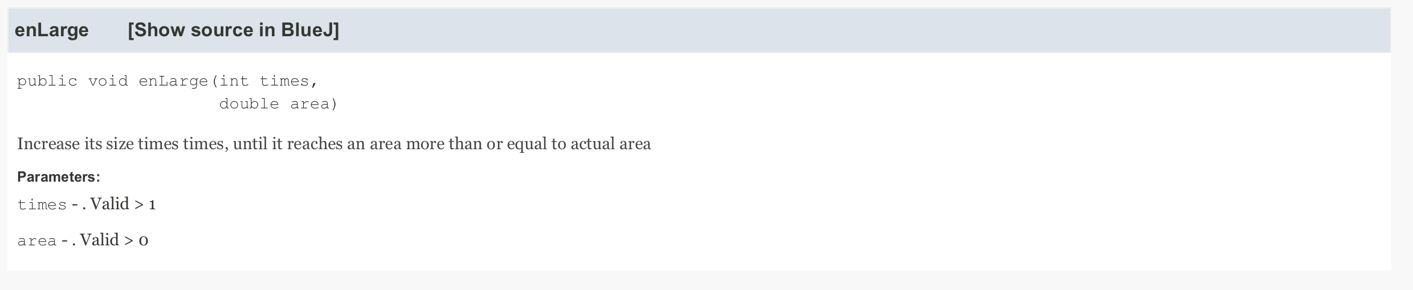
* 1. Propongan un nuevo método para esta clase. Desarrollen y prueben el método.

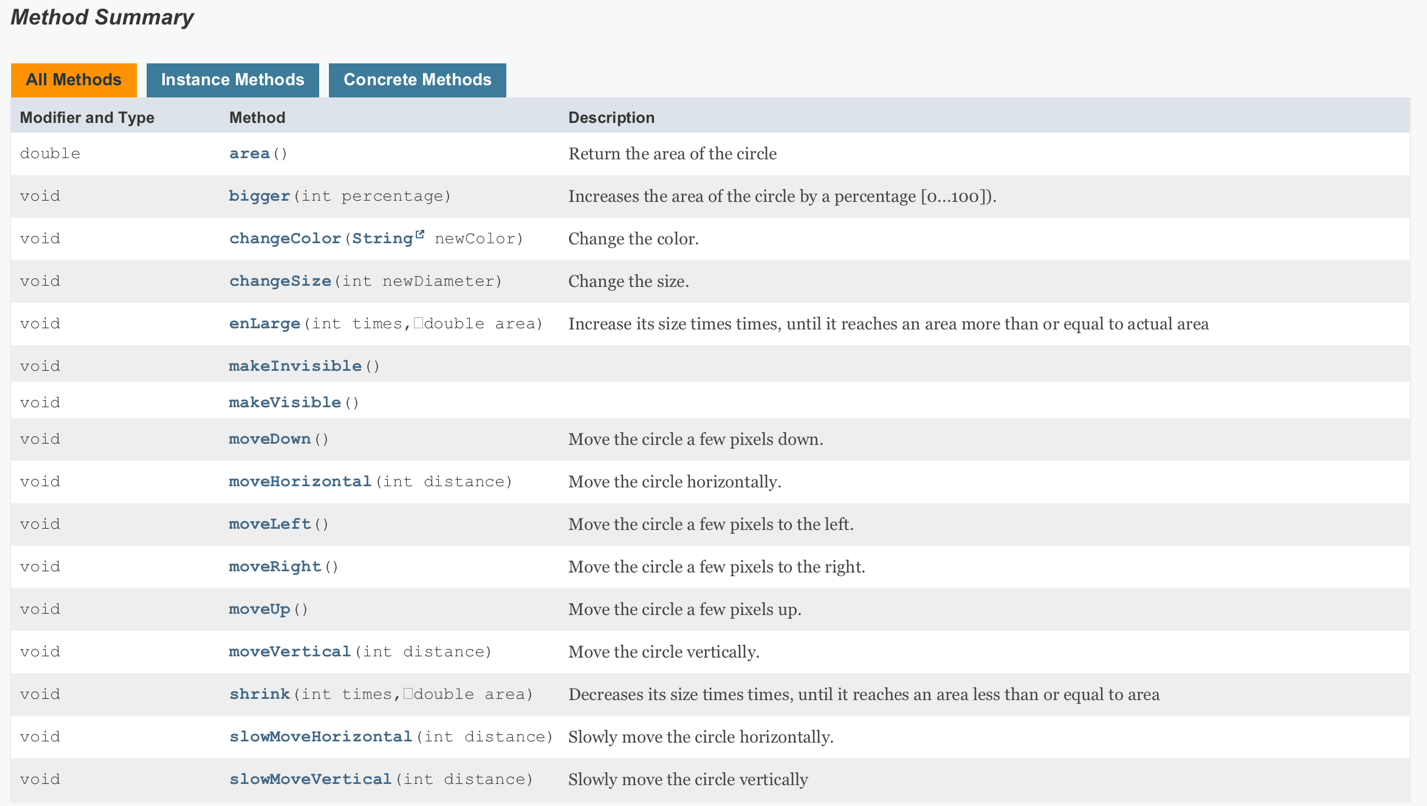




* 1. Generen nuevamente la documentación y revisen la información de estos nuevos métodos. Capturen la pantalla.







1. **De Python a Java** [**En lab01.doc]**

En este punto vamos a usar y evaluar dos recursos de apoyo para la transición de Python a Java. Realicen la evaluación en las encuestas preparadas con ese objetivo:

* 1. El video.
  2. Los prompts.

Nicolas Bernal:



Juan Bogotá:



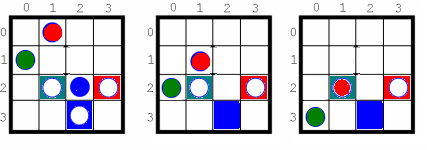
# MINI MARBEL GAME

El objetivo de este punto es desarrollar una mini-aplicación para una versión simplificada del juego Marbel Game. Inspirado en un problema **F** de la maratón internacional 2007. [VINCULO](https://icpc.global/worldfinals/problems/2007-ICPC-World-Finals/2007WorldFinalProblemSet.pdf)

El tablero de este juego es cuadrado (NXN) e inicia con igual número de canicas y de huecos (M), cada par de color diferente. El objetivo de este juego es hacer que todas las canicas caigan en el hueco de su mismo color. Los movimientos consisten en levantar un lado del tablero para hacer que las canicas se deslicen una posición. Las canicas caen si pasan por un agujero. Si una canica cae en un agujero que es de su color, el agujero desaparece. Si una canica cae en un agujero que no es de su color, el juego termina.

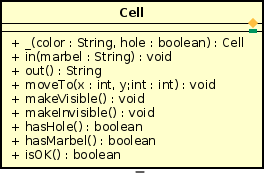
En el ejemplo se muestra un tablero de 4 x 4, con tres canicas. Los movimientos realizados fueron levantar el norte dos veces.

6Menú: View *→* Show Code Pad



* 1. **Creando una nueva clase. Usando un paquete** shapes [**En lab01.doc]**[**En .java]**

En este punto vamos a crear las **casillas** del tablero. El diseño gráfico lo definen ustedes. Las casillas deben ofrecer los siguientes métodos:



**Ciclo 1:**

- \_()

* hasHole

**Ciclo 2:**

* in
* out
* hasMarbel
* isOK

**Ciclo 3:**

* moveTo
* makeVisible
* makeInvisible

**Notas:**

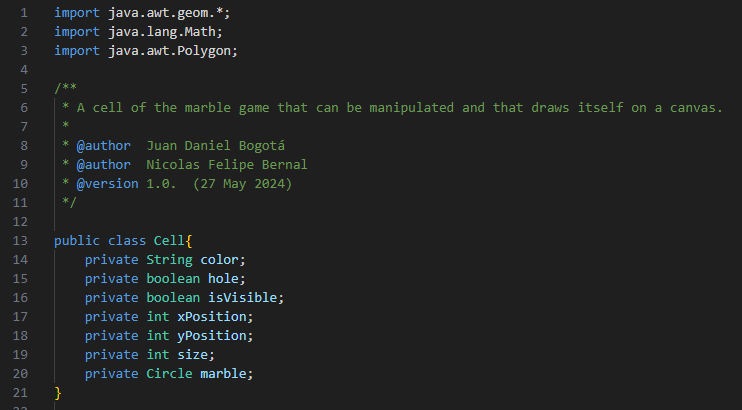
* En in entra una canica del color indicado.
* En out sale la canica que está en la casilla y se re-

torna su color, si es posible. Si no es posible, retorna la cadena vacía.

* isOK retorna falso en el agujero está una canica de

color diferente

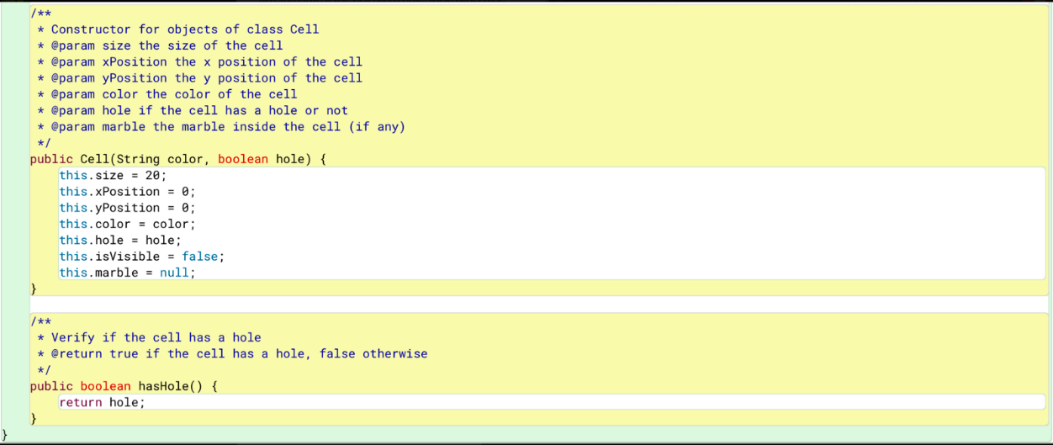
* 1. Inicie la construcción únicamente con los **atributos**. Justifique su selección. Agregue pantallazo con los atributos.

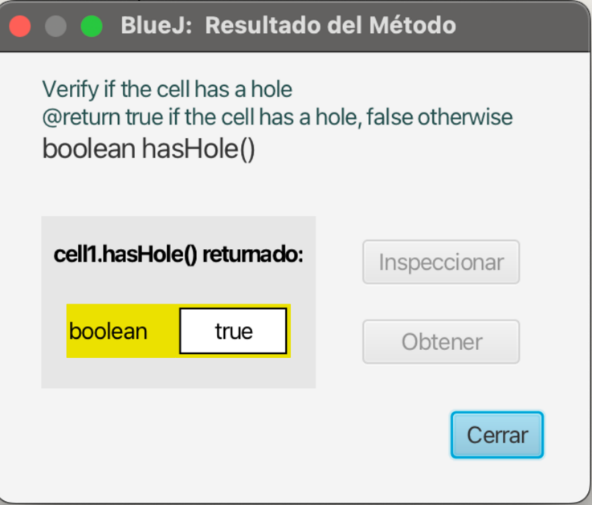
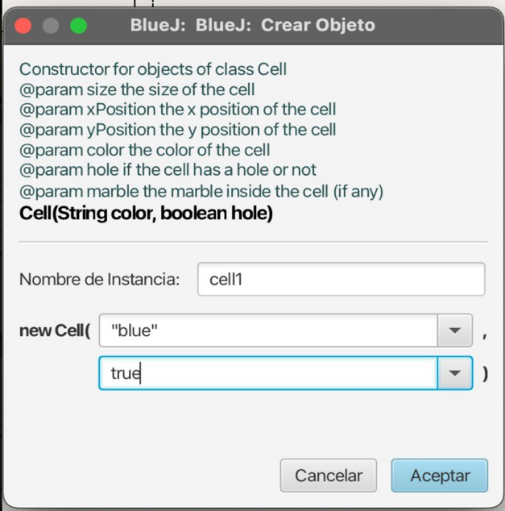


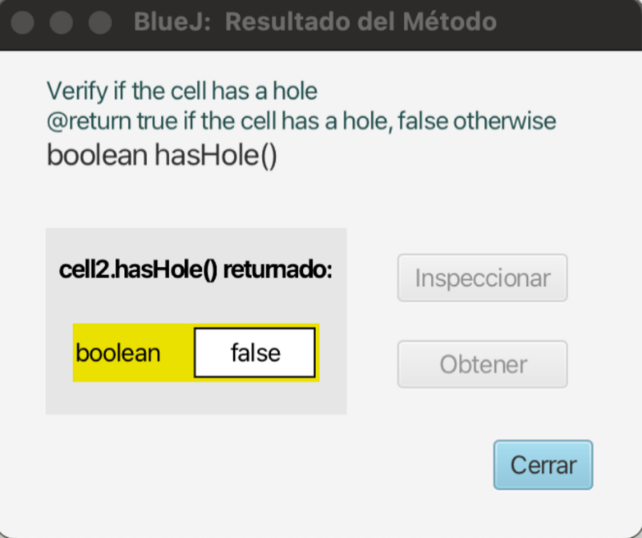
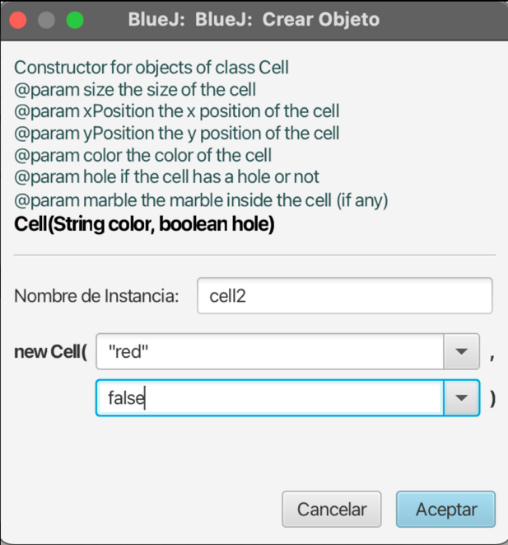
* 1. Desarrollen la clase considerando los 3 mini-ciclos. Al final de cada mini-ciclo

realicen dos pruebas indicando su propósito. Capturen las pantallas relevantes.

Ciclo 1:

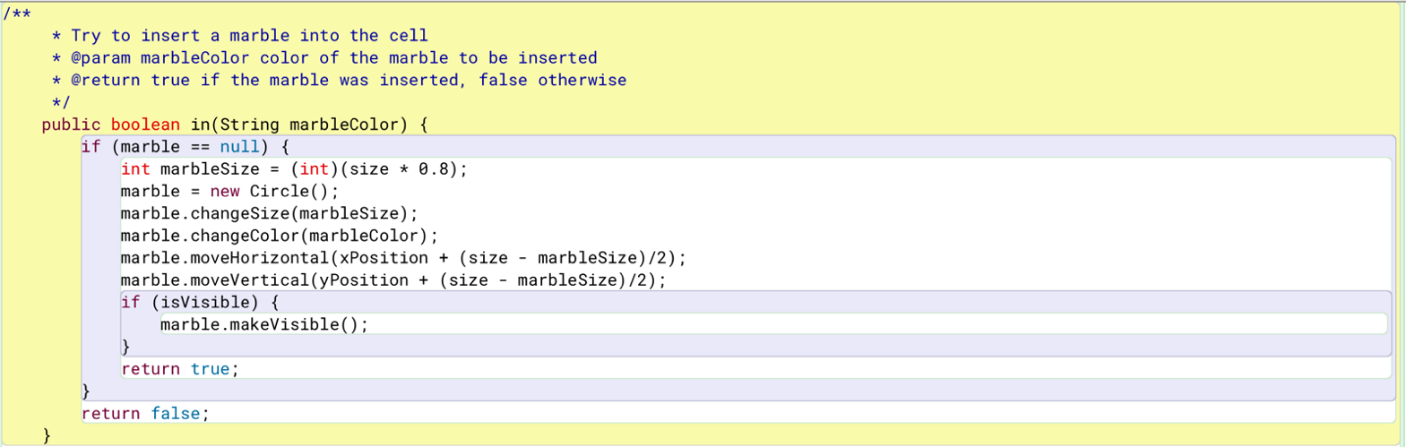


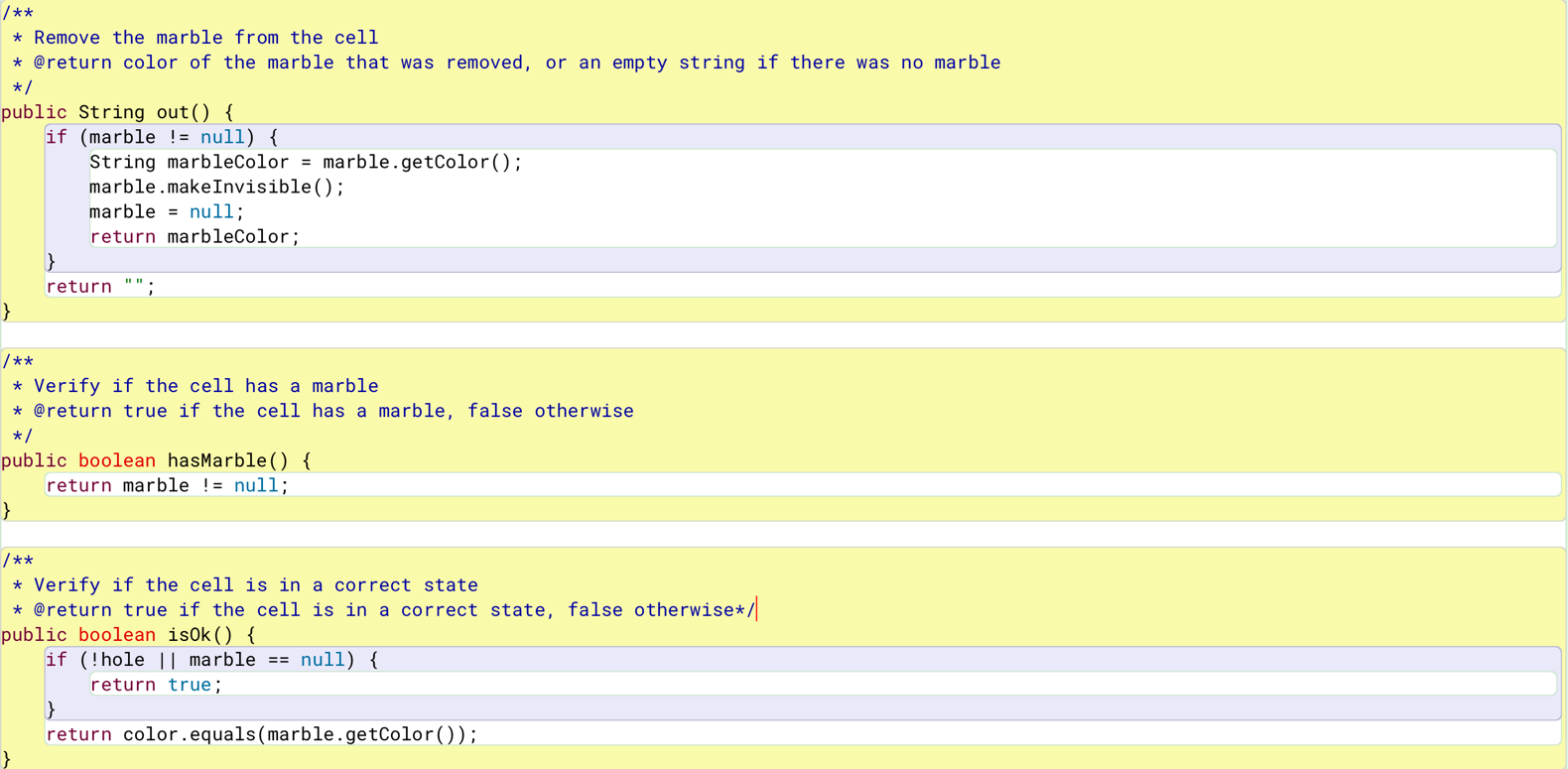


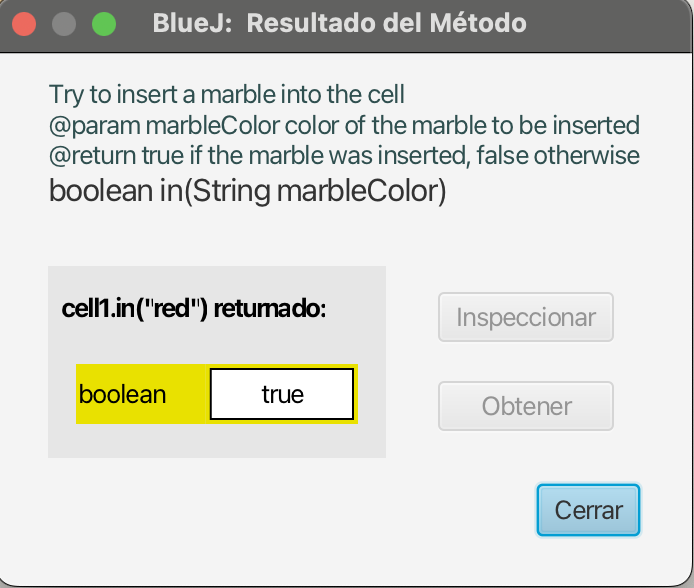
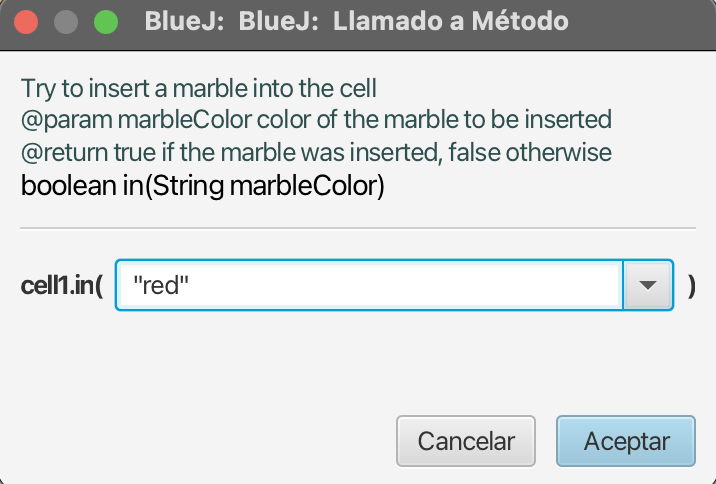
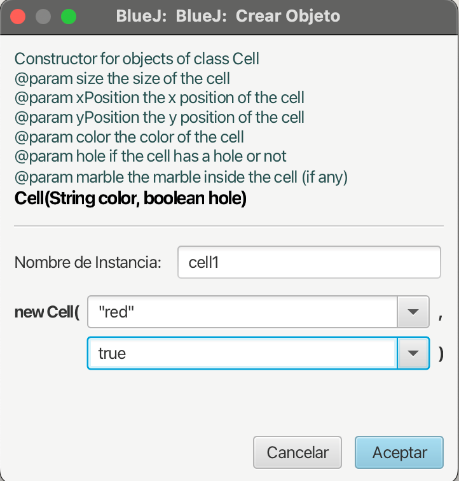


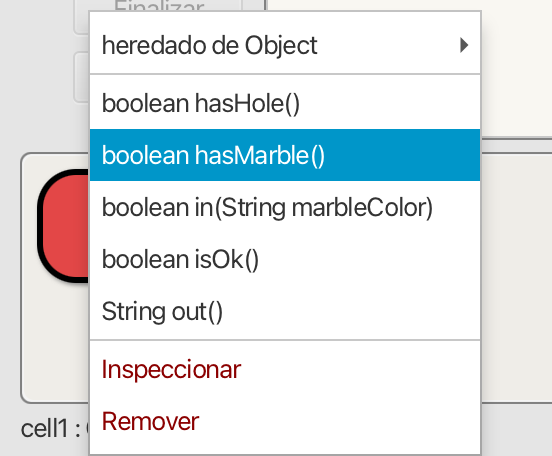
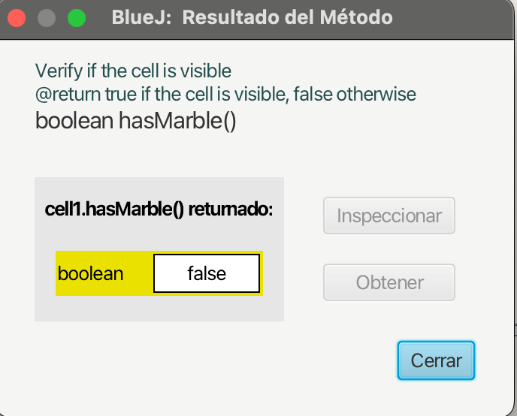
Se probó el creador con los dos parámetros que se señalan en el diagrama de clase de la clase Cell, además se probó el método hasHole() con ambas posibilidades de respuesta.

Ciclo 2:



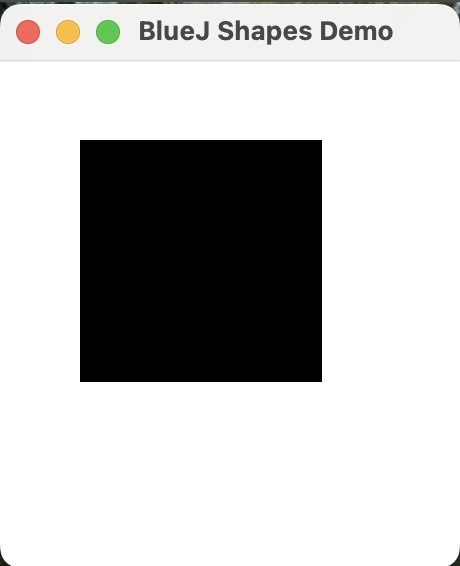
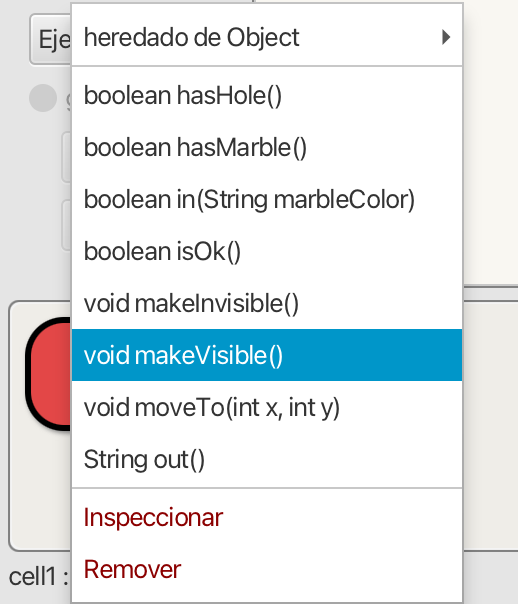


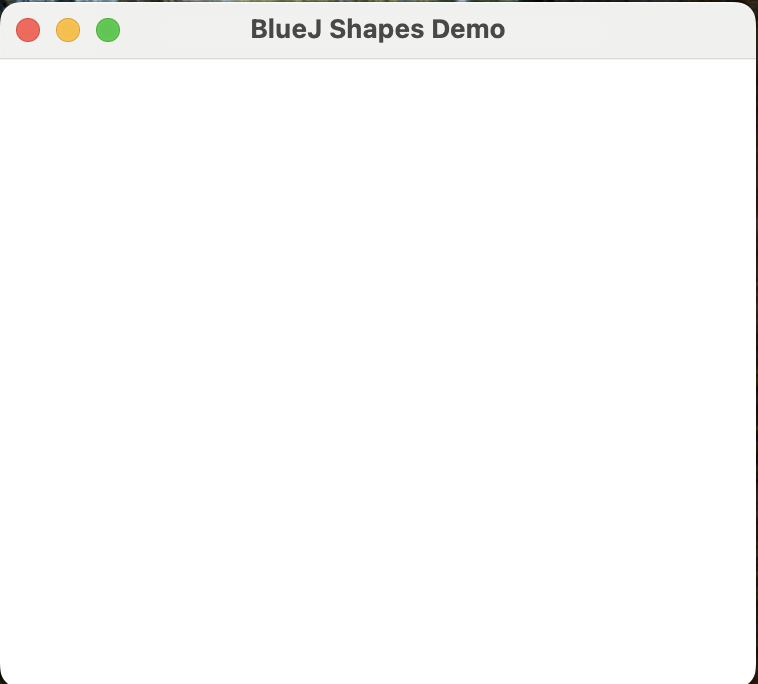
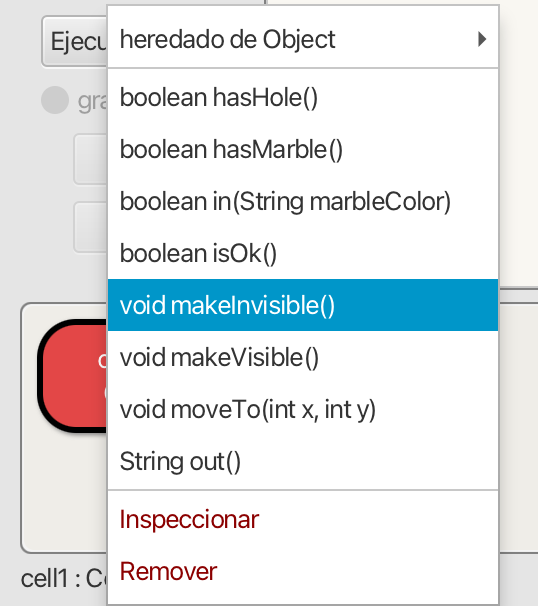




Se hicieron dos pruebas, al método in(String marbleColor) y retorna true, ya que permite insertar la canica si fuera roja y además porque no hay mas canicas en el hoyo, la otra prueba fue a el método hasMarble(), que dio false, ya que aun no existe la canica.

Ciclo 3:





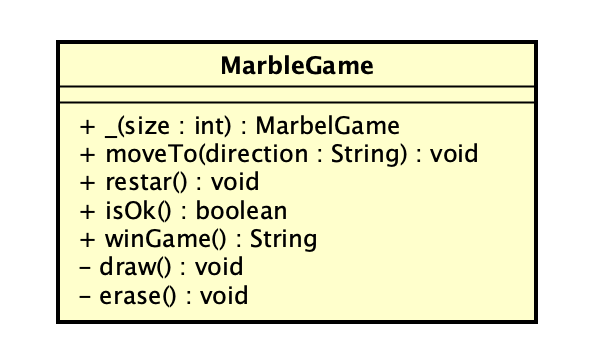
Se hicieron dos pruebas, al método makeVisible() y muestra la figura en el canvas, la otra prueba fue a el método makeInvisible(),que muestra el canvas en blanco ya que vuelve invisible la figura.

**B. Definiendo y creando una nueva clase.** MarbelGame**.** [**En lab01.doc]**[**En .java]**

En este punto vamos a desarrollar la *mini-aplicación* para Marbel Game.

|  |  |
| --- | --- |
| **Requisitos funcionales** | **Requisitos de interfaz** |
| Crear el estado inicial. Realizar los movimientos. Reiniciar el juego.  Consultar el estado del juego (un mensaje con el número  de canicas en sus agujeros).  Informar cuando alguien gana el juego (un mensaje de fe-  licitación). | Los movimientos se identifican por el lado (’North’, ’South’, ’East’, ’West’).  En caso de que no sea posible realizar una de las acciones, debe generar un mensaje de error.  Para los mensajes use JOptionPane. |

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

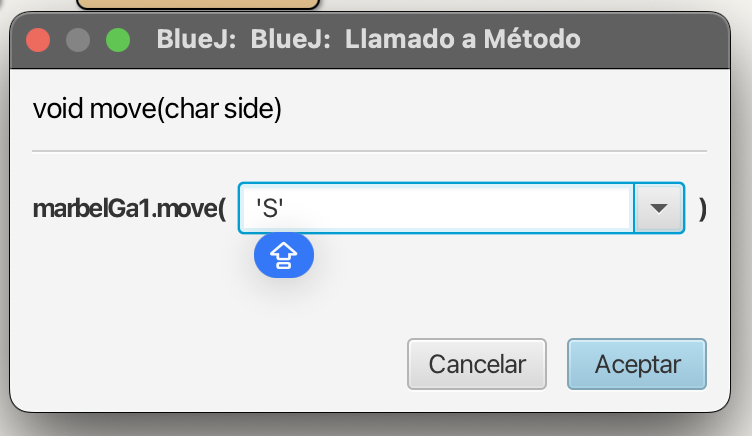
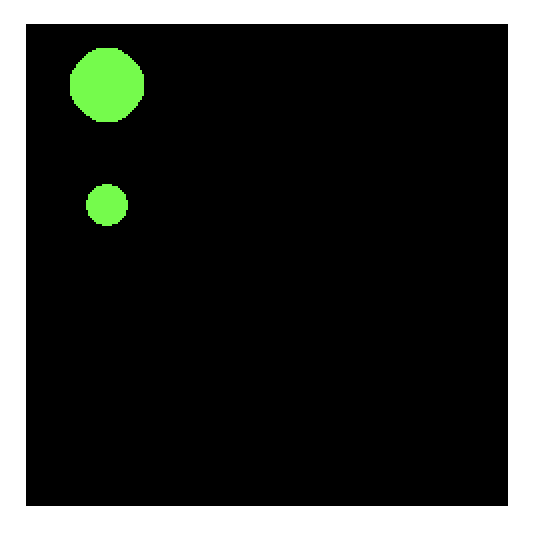
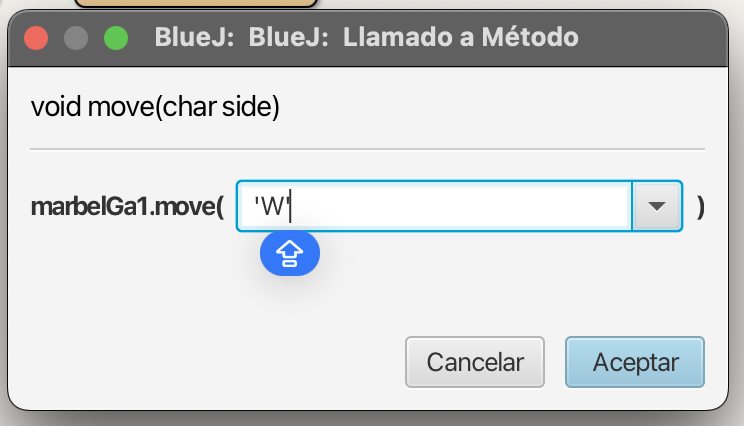
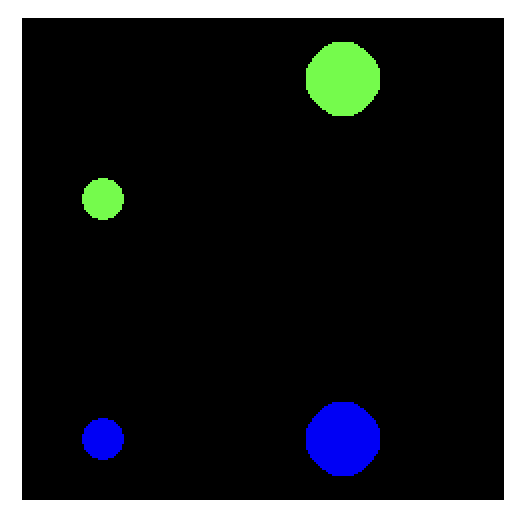
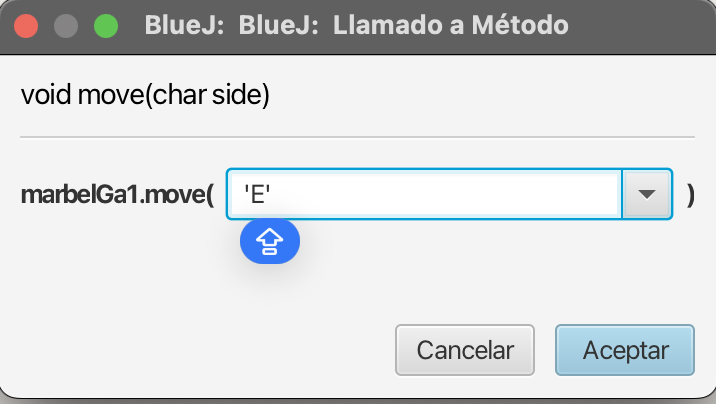


1. Planifiquen la construcción considerando algunos mini-ciclos.

* Ciclo 1:
  + \_()
* Ciclo 2:
  + draw
  + erase
  + moveTo
* Ciclo 3:
  + Restar
  + isOk
* Ciclo 4:
  + winGame

1. Implementen la clase. Al final de cada mini-ciclo realicen una prueba indicando su propósito. Capturen las pantallas relevantes.
2. Indiquen las extensiones necesarias para reutilizar la clase Cell y el paquete shapes.
3. Expliquen.





**C BONO. Nuevos requisitos funcionales.** MarbelGame**.**[**En lab01.doc]**[**En .java]]**

Extiendan la mini-aplicación MarbelGame:

 Hacer un buen movimiento (la máquina decide). Explique la estrategia.  Deshacer el último movimiento. Deshacer se considera un movimiento.

* 1. Diseñen, es decir, definan los métodos que debe ofrecer.
  2. Implementen los nuevos métodos. Al final de cada método realicen una prueba in- dicando su propósito. Capturen las pantallas relevantes.

# RETROSPECTIVA

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

Juan Daniel Bogotá Fuentes 28 horas, más o menos 3 horas por día (jueves, lunes,martes,

miércoles, jueves, viernes, sábado).

Nicolas Felipe Bernal Gallo 28 horas (Jueves[3 horas], lunes[2 horas], martes[3 horas],

miércoles[1 hora], jueves[6 horas], viernes[6 horas], sábado[7 horas])

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

Incompleto, porque no realizamos el bono y tuvimos errores dentro del laboratorio lo que no permite que sea un laboratorio completo.

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

Dos programadores trabajan juntos en el código (Pair Programming). Esta es la prácticaXP que nos fue más útil a la hora de realizar este laboratorio, puesto que nosotros nosintercambiábamos la labor de escribir código y revisar el código de la otra persona.También porque intercambiamos ideas y conocimiento que fuimos adquiriendo en elauto estudio de cada persona.

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

Entender todos los requerimientos que exigía el laboratorio a partir de las preguntas que fueron expuestas para su resolución. No dejando por nuestra parte lugar a la ambigüedad de nuestras respuestas.

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

El mayor problema técnico que presentamos a la hora de realizar este laboratorio fue la no interconectividad que presenta Blue J, donde no podíamos escribir en el mismo archivo al mismo tiempo, lo cual nos llevó a buscar otro entorno virtual que satisficiera esa necesidad. Luego de esto volver a Blue J y generar así el “Shapes” final.

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

Repartir de una buena forma el trabajo para que ambos aprendamos y además tengamos la misma carga de trabajo. Nos comprometemos a repartir mejor las horas por día para hacer el laboratorio

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

*ChatGPT*. (s/f). Chatgpt.com. Recuperado el 30 de Agosto de 2025, de <https://chatgpt.com/>

Barragán, A. (2023, octubre 12). Introducción a Java: Operadores. *Openwebinars.net*.

<https://openwebinars.net/blog/introduccion-a-java-operadores/>

Vypirailenko, A. (2023, julio 21). *División entera Java*. CodeGym.

<https://codegym.cc/es/groups/posts/es.696.division-entera-java>

*Los 11 mejores chatbots de IA para 2025*. (s/f). Getguru.com. Recuperado el 6 de febrero de 2025, de

<https://www.getguru.com/es/reference/best-ai-chatbots>

El mejor fue Getguru.com ya que una de las cosas que más nos tomó tiempo, fue encontrar un logo de envíos adecuado para realizarlo en Bluej.