PROGRAMACIÓN ORIENTADA A OBJETOS

## Construcción. Clases y objetos.

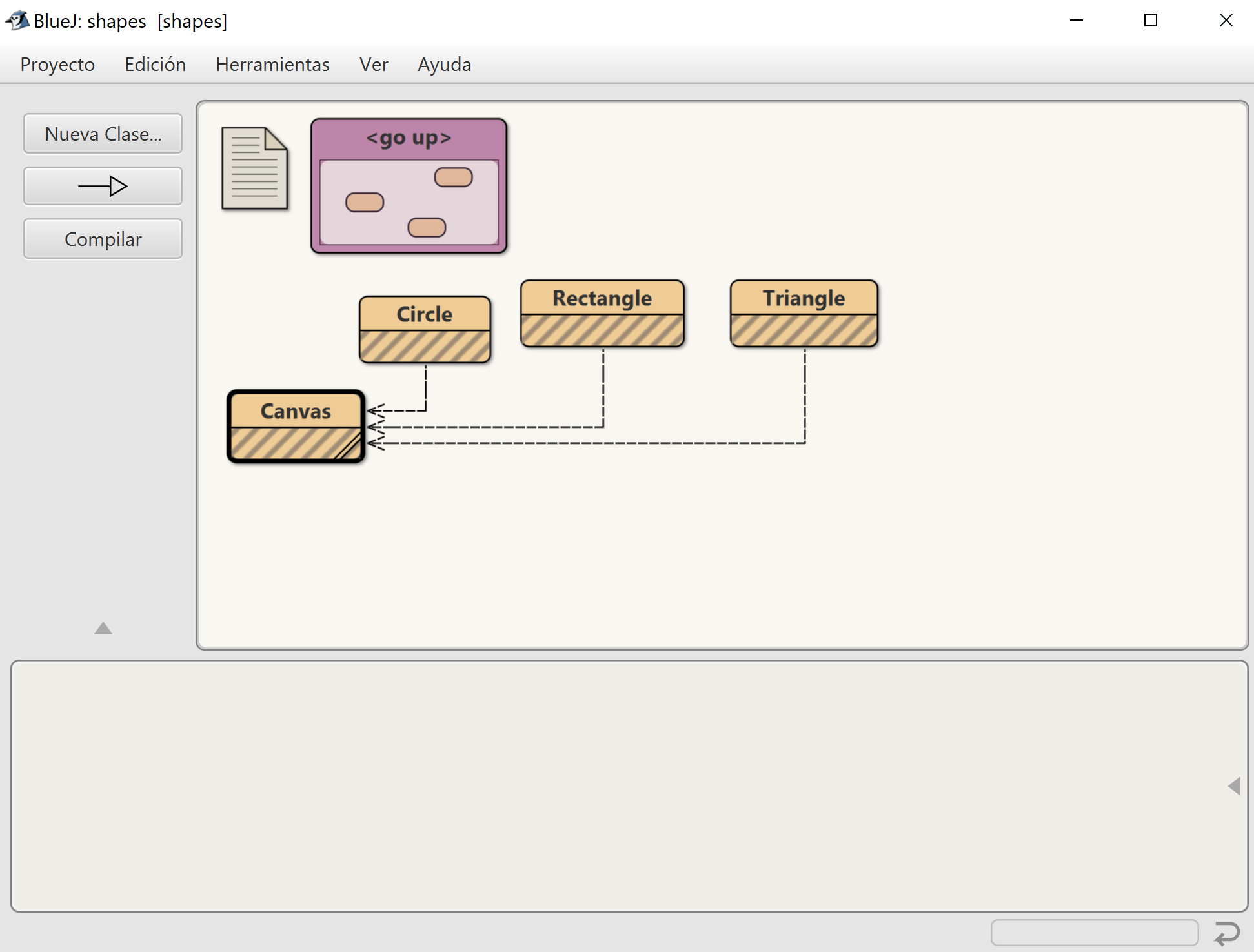
## 2022-1

## Laboratorio 1

## Andrés Camilo Oñate Quimbayo, Nicolas Ariza Barbosa

## SHAPES

## A. Conociendo el proyecto shapes

1. Proyecto
2. Considerando el diagrama de clases de “shapes”, las clases que ofrece son:

* Canvas
* Circle
* Rectangle
* Triangle

Relaciones:

* La clase Canvas se relaciona con todas las demás clases, ya que hace uso de ellas.

1. **Documentación**
2. Las clases que contiene el paquete son

* Circle
* Rectangle
* Triangle

1. **Atributos** de la Clase Circle según la documentación:

* PI

1. Métodos de la Clase Circle ofrece 12 métodos según la documentación.
2. Métodos para que la figura cambie:

- makeVisible

- makeInvisible

- changeSize

- changeColor

1. Código
2. La clase tiene 5 atributos en el código.
3. Solo hay un atributo que describe la forma del círculo y ese es diameter.
4. La clase en total tiene 14 métodos.
5. Los metodos privados son draw y erase y los usan:

* makeVisible
* makeInvisible
* moveHorizontal
* moveVertical
* lowMoveHorizontal
* slowMoveVertical
* changeSize
* hangeColor

1. En la documentación no se aprecian los atributos y métodos de la clase cuya visibilidad está en.
2. Cualquier característica de una clase que sea privado puede ser accedido únicamente por la misma. Es decir, si, por ejemplo, un atributo es privado, solo puede ser accedido por lo métodos o constructores de la misma clase.

Ninguna otra clase sin importar la relación que tengan podrá tener acceso a ellos. Dado que solo pueden ser usados en la clase , no amerita que estén en la documentación.

1. es el tipo de accesibilidad que otras clases tienen respecto a ella, en este caso este atributo es públicopara todas las clases e indica que es accesible a través de una instancia del objeto.
2. le da a este atributo la característica de ser una variable global, es decir, su valor es universal para todas las clases dentro de nuestro programa.
3. establece que el valor contenido dentro de este atributo es inmutable, es decir, ninguna clase puede modificar el valor originalmente contenido en este atributo.
4. Double se utiliza para representar números de punto flotante, usa 64 bits para almacenar un valor variable. Como el atributo Pi que se utiliza en la clase tiene pocos decimales, se hace inadecuado reservar tanto espacio en memoria, los 32 bits del tipo float son suficientes, Pi debería ser de tipo .

Texto

Descripción generada automáticamente

1. El atributo tiene como tipo de dato int, es un entero de 32 bits complemento a dos. Su valor mínimo es -2,147,483,648 y el máximo 2,147,483,647.
2. El atributo debería ser de tipo *byte* ya que en memoria este tipo de dato es capaz de almacenar valores numéricos enteros desde –128 hasta 127.
3. Para diámetros muy grandes, este atributo debería ser de tipo *short* ya que este tipo de dato puede almacenar números que van desde-32768 hasta 32767.
4. Se refactoriza el tipo de dato del atributo en la declaración de la clase y en el método changeSize.

Texto

Descripción generada automáticamente

# 

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteEl propósito del proyecto es busca mostrar la estructura (visual y en código) más básica que pueden tener las diferentes clases en un entorno de trabajo con O.O. Nos invita a analizar los diferentes componentes (atributos y metodos) de una clase en específico y su papel en una buena abstracción de modelos de la vida real hacia el mundo digital.

## B. MANIPULANDO OBJETOS. USANDO UN OBJETO

Después de compilar el material:

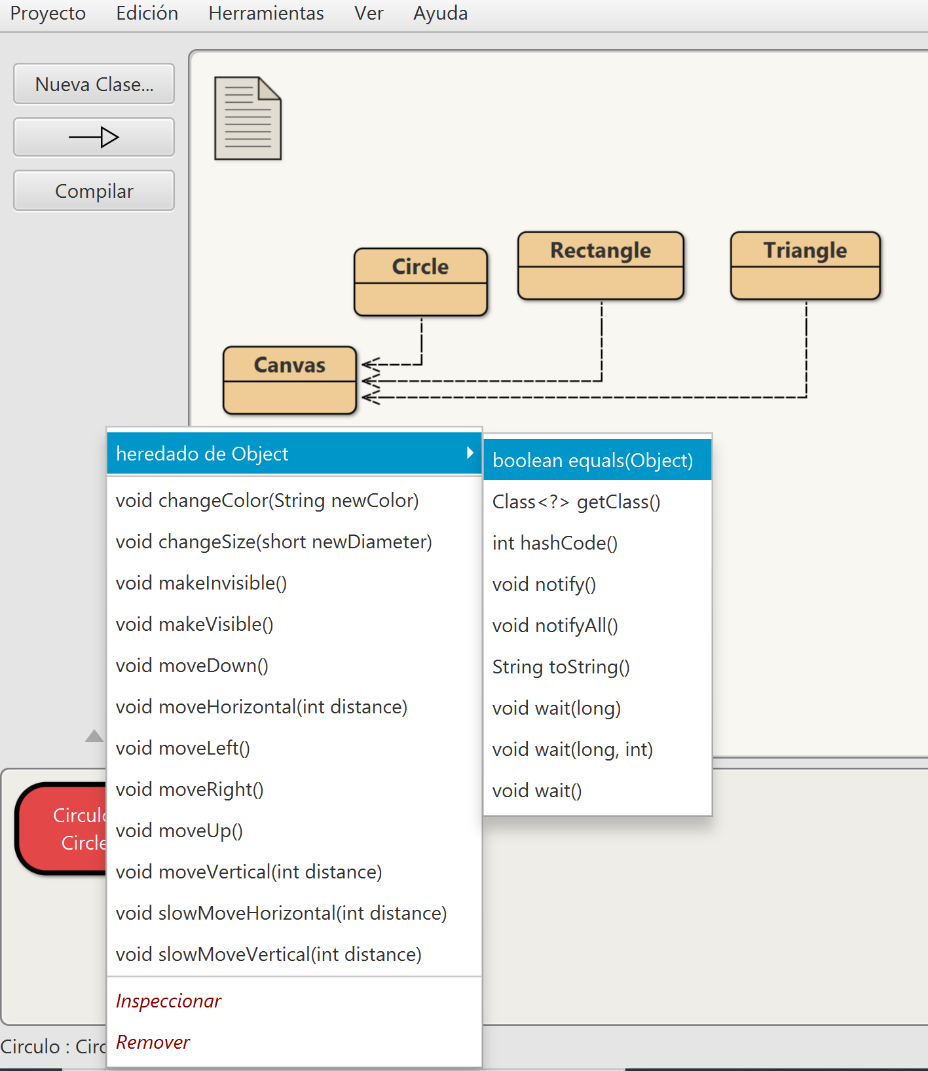
# Interfaz de usuario gráfica Descripción generada automáticamente con confianza baja

1. Hay tres clases: Circle, Rectangle, Triangle y Canvas.
2. Creamos instancias de todas las posibles clases dentro del proyecto , el motivo de generar cuatro objetos y no tres viene dadas a que nuestro entorno visual o también es una clase y por lo tanto es posible generar un objeto a partir de esta clase. Además, sin entorno visual no seriamos capaces de ver los demás objetos que ya se crearon previamente.
3. Interfaz de usuario gráfica

   Descripción generada automáticamente

Los valores de inicio son los que están establecidos en el constructor de la clase.

1. En la documentación no se aprecian los métodos de la clase cuya visibilidad está en prívate:

* erase
* draw

1. Interfaz de usuario gráfica, Aplicación

   Descripción generada automáticamenteRed Social: LinkedIn
2. Se usaron dos clases: Rectangle y Triangle.
3. Se usaron en total 7 objetos:

* Un objeto canvas
* Cuatro instancias de la clase rectangle
* Una instancia de la clase circle

1. Logo original.



## C. MANIPULANDO OBJETOS. ANALIZANDO Y ESCRIBIENDO CÓDIGO

1. La figura resultante esperada es un rostro

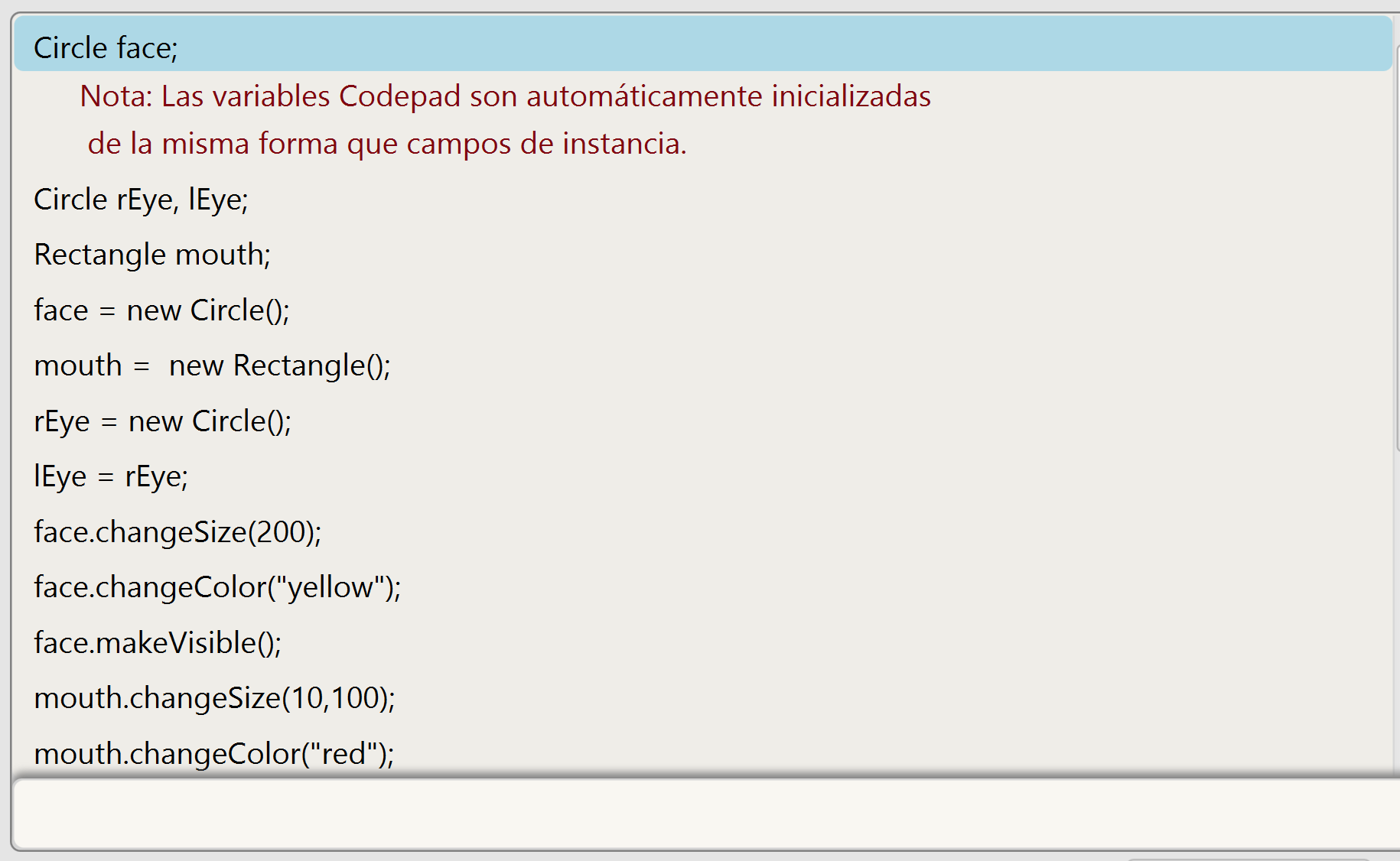
## 

1. Dado que la línea de código con la declaración: no crea una nueva instancia de la clase Circle, sino que los punteros hacen referencia al mismo objeto, la figura resultante es:

Interfaz de usuario gráfica, Gráfico

Descripción generada automáticamente

1. Hay un total de 16 variables únicamente en la representación de este rostro. 5 variables por cada circulo (10) y 6 por el rectángulo.
2. Según la figura esperada en total existen 5 objetos, tres círculos, un rectángulo y un canvas. Al ejecutar el código, es posible ver que solo existen 4 objetos, dos círculos, un rectángulo y el canvas.
3. Un círculo es de color amarillo, otro de color azul y el rectángulo es de color rojo.
4. En total se ven en pantalla 4 objetos, el canvas que es nuestro entorno grafio, un rectángulo y dos círculos.



.

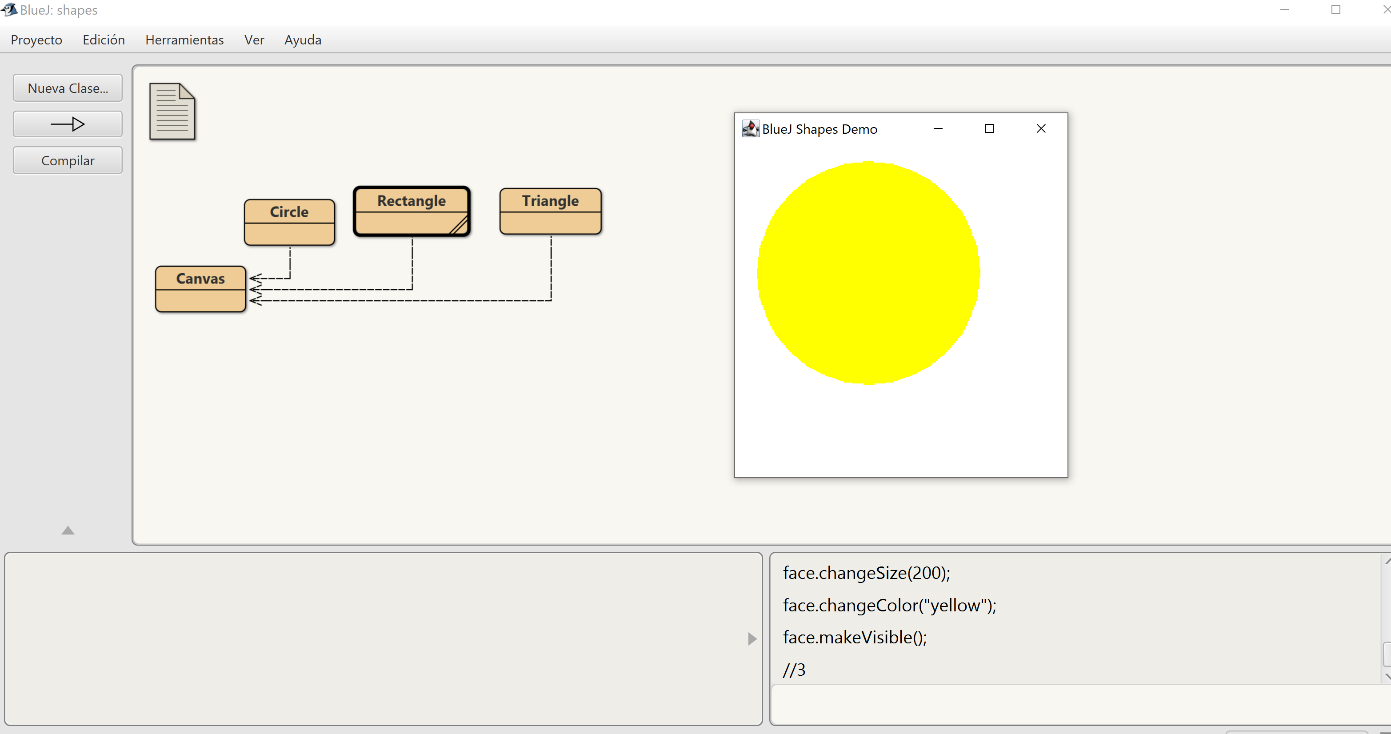
1//

* Variables: 0
* Existen cuatro objetos sin variable definida: Face, rEye, lEye, mouth.
* El constructor pone su color en “blue” a los círculos, magenta al rectángulo.
* No se ve ningún objeto

2//

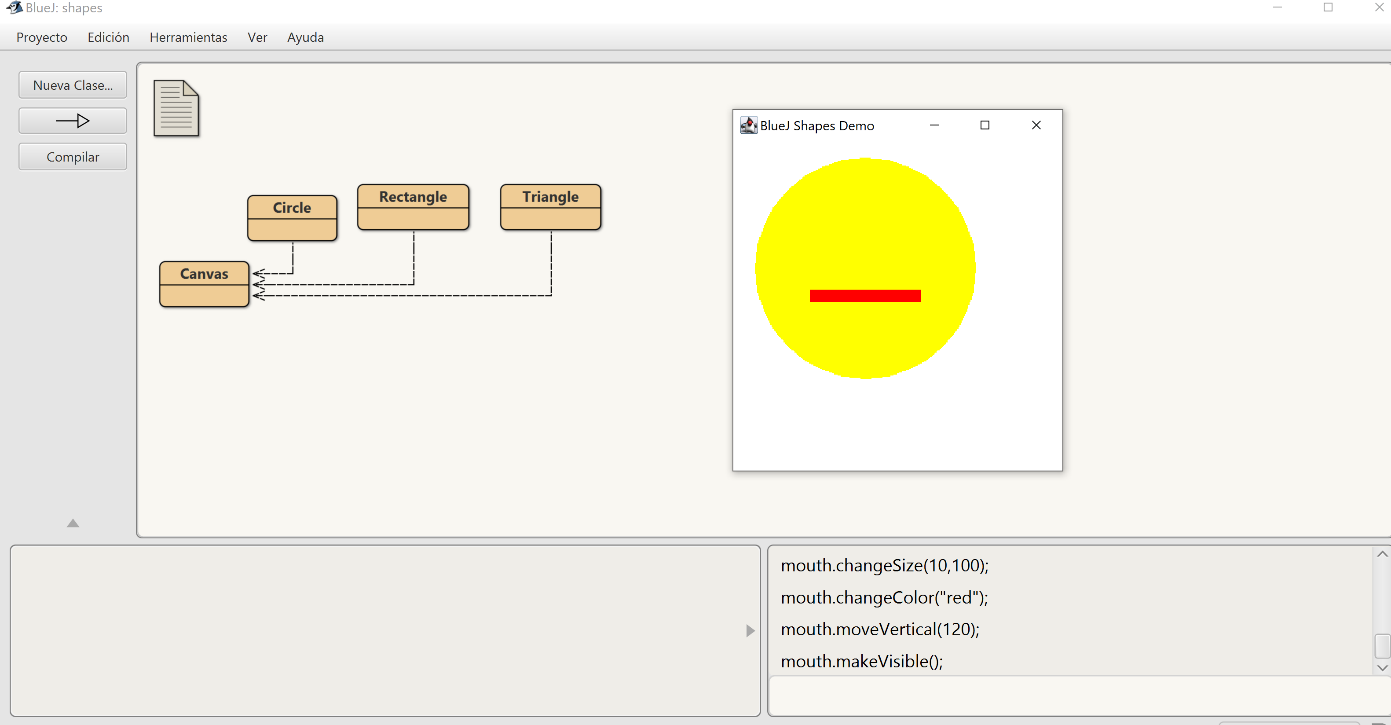
* Variables: 4
* Existen tres instancias de los objetos. (Debido a la línea )
* Blue, magenta.
* Ninguno.

3//

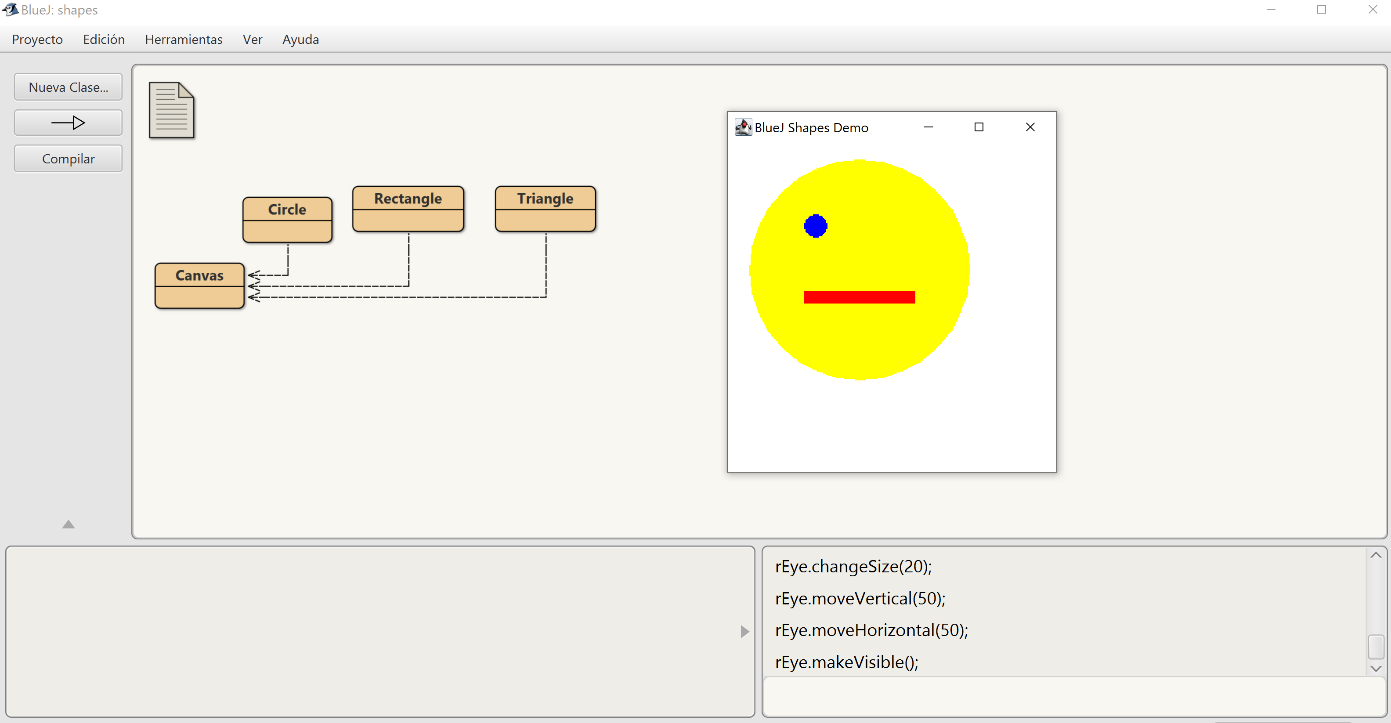
* Variables: 4
* Existen cuatro instancias de los objetos. (Canvas)
* Yellow, magenta.
* Objeto face mostrado en el objeto canvas.
* 

4//

* Variables: 4
* Existen cuatro instancias de los objetos. (Canvas)
* Yellow, rectángulo: “red”.
* Objetos fase y mouth mostrado en el objeto canvas.



5//

* Variables: 4
* Existen cuatro instancias de los objetos. (Canvas)
* Face: Yellow, mouth: “red”, rEye: “Blue”.
* Objetos fase, rEye y mouth son mostrados en el objeto canvas.

6//

* Variables: 4
* Existen cuatro instancias de los objetos. (Canvas)
* Face: Yellow, mouth: “red”, rEye: “Blue”.
* Interfaz de usuario gráfica

  Descripción generada automáticamenteObjetos fase, rEye y mouth son mostrados en el objeto canvas.

Gráfico

Descripción generada automáticamente

1. Las figuras planteadas son semejantes, no son exactamente iguales ya que el diámetro de las circunferencias, y las dimensiones del rectángulo no son iguales para los dos modelos.
2. Como ya se ha mencionado, no se instancias dos objetos para los ojos de la cara, sino que ambas variables señalan al mismo objeto, al momento de realizar la parte 6// del código, el circulo sufre cambios no adecuados.

## 

## D. EXTENDIENDO UNA CLASE. CIRCLE

Imagen que contiene Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteAsignando un diámetro de 2 unidades (el radio es de 1 unidad)

Llamando al método

Interfaz de usuario gráfica, Aplicación

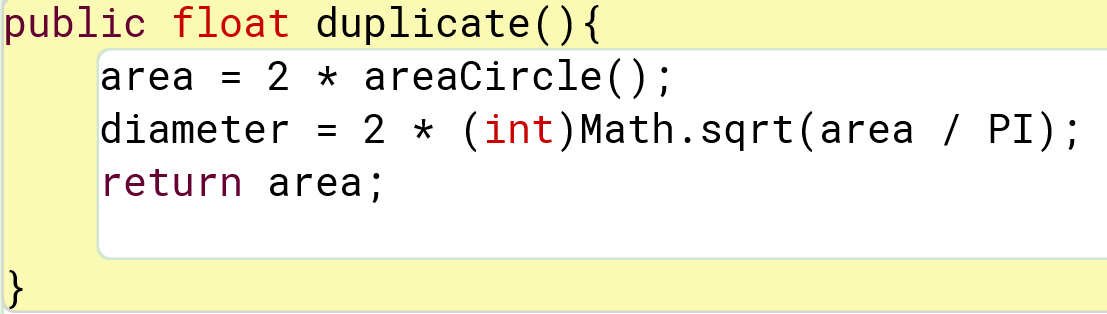
Descripción generada automáticamente

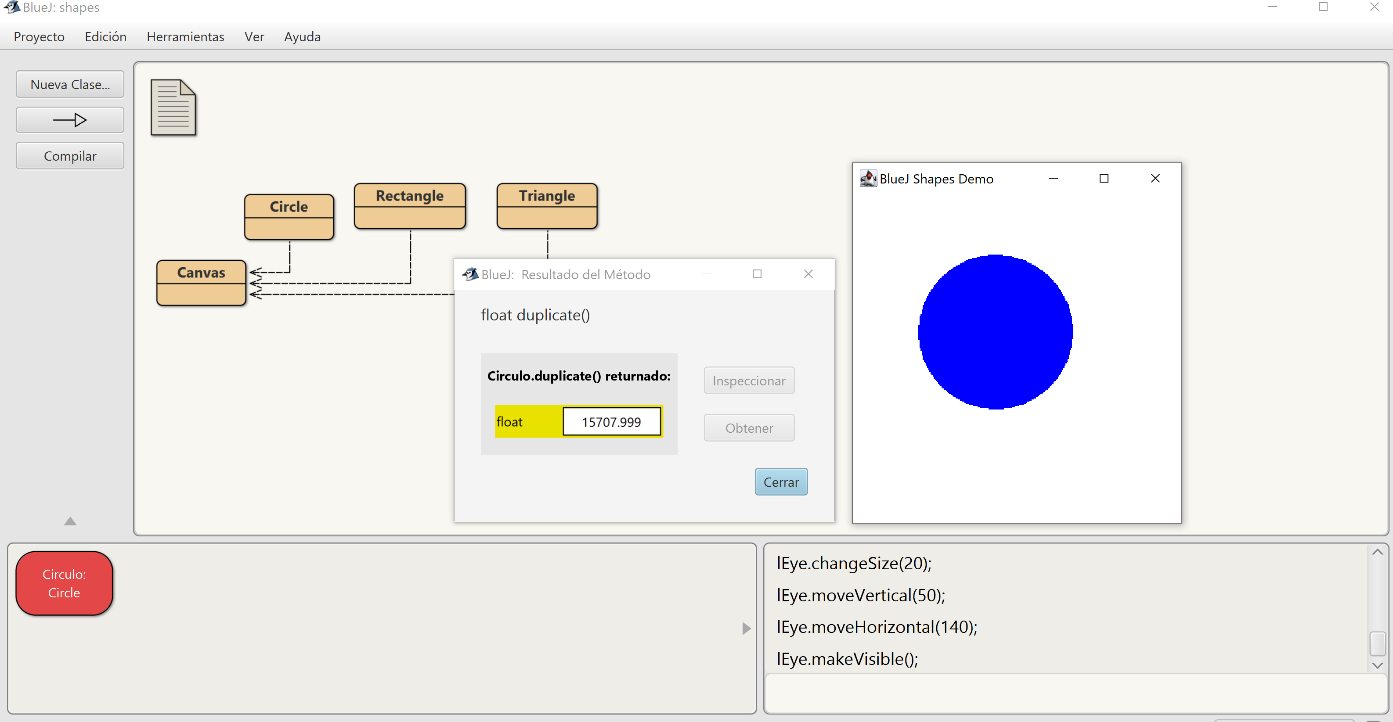
Y retorna el valor del área de dicho circulo

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Método :



Interfaz de usuario gráfica

Descripción generada automáticamenteSe crea el método , se hace el llamado el método , después se corrige el valor del atributo correspondiente al doble del área.

1. Método . Se hace uso de un ciclo for:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

1. Determinar el volumen de un cilindro con una base circular con diámetro .

Texto, Aplicación

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

.

1. Documentación:

Imagen que contiene Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Tabla

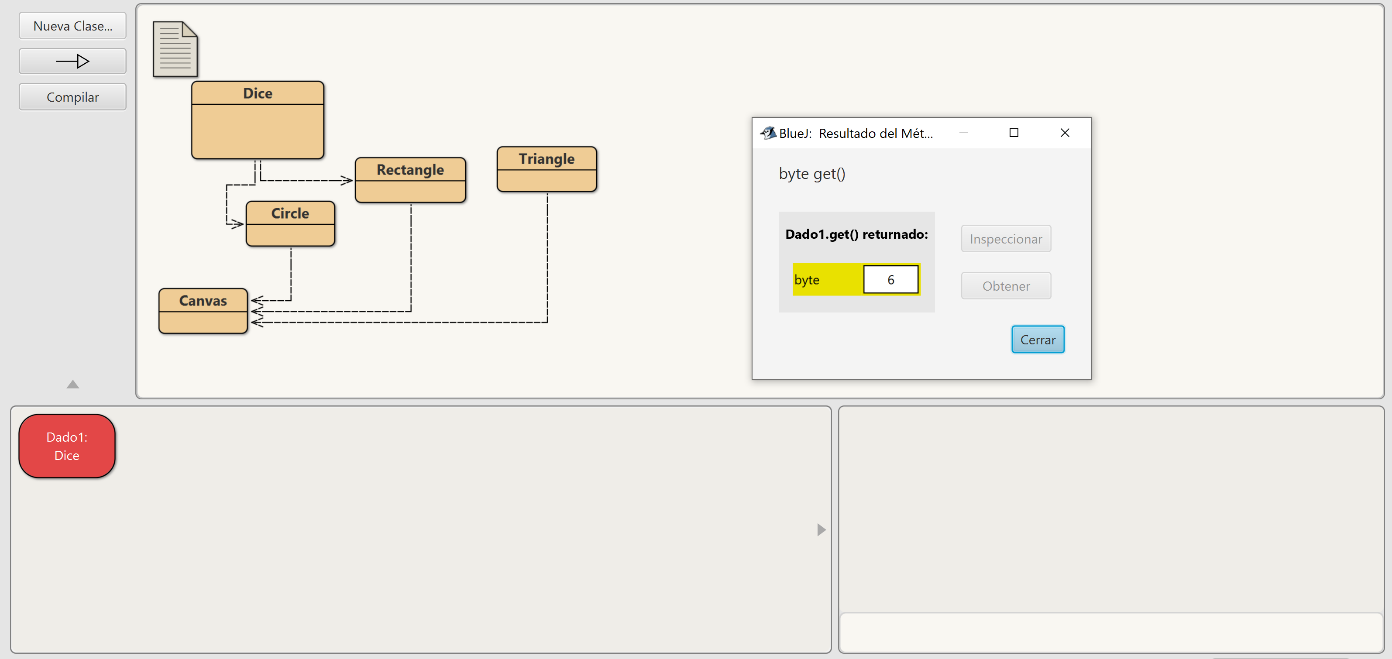
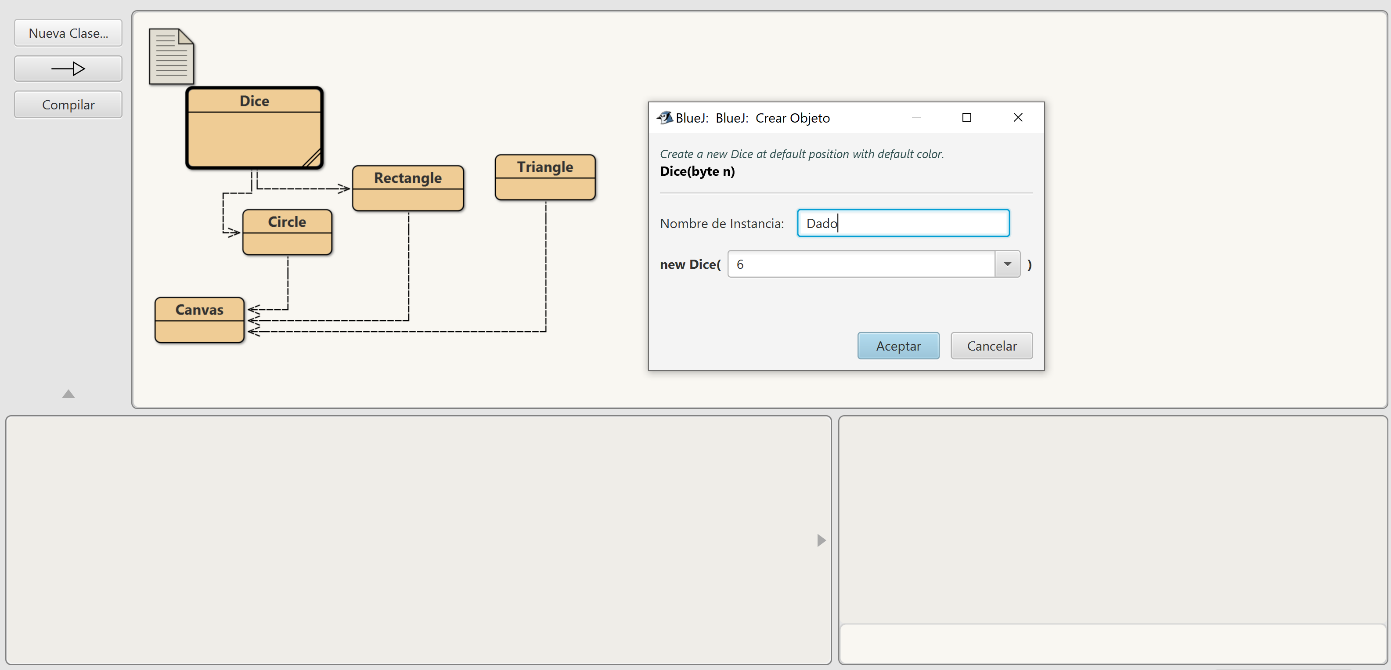
Descripción generada automáticamente

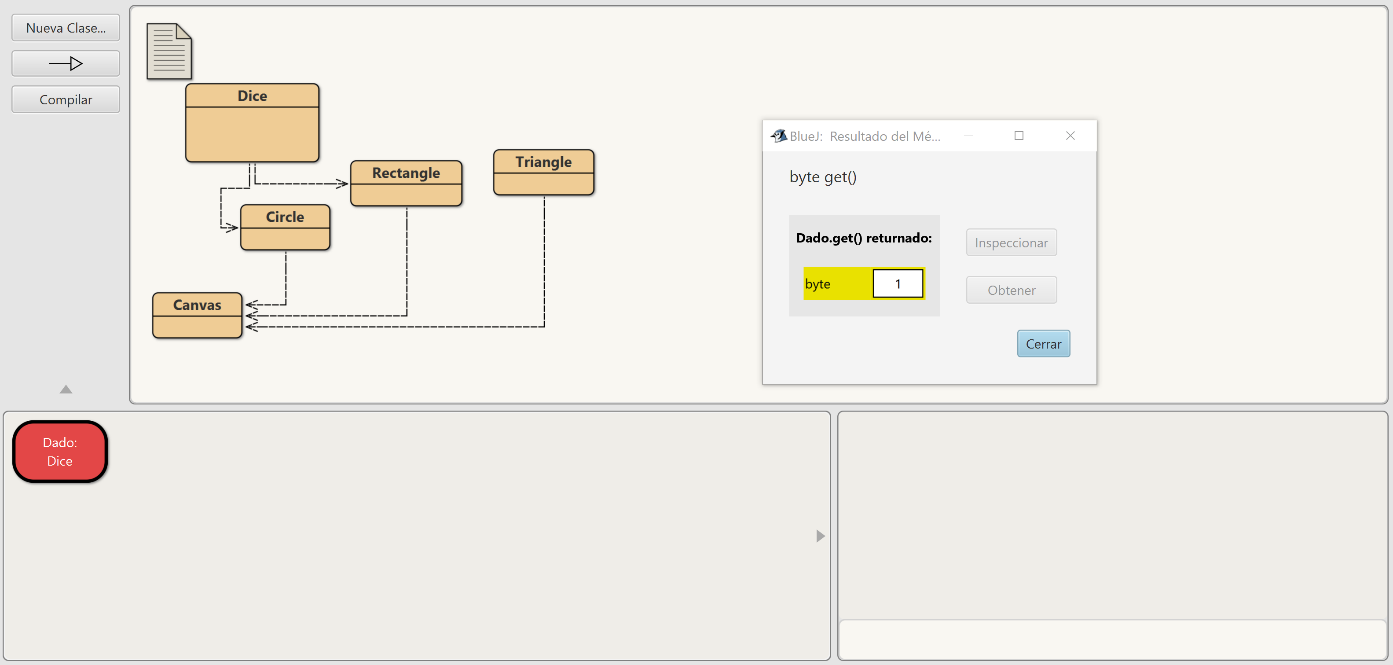
## E. Codificando una nueva clase. Dice

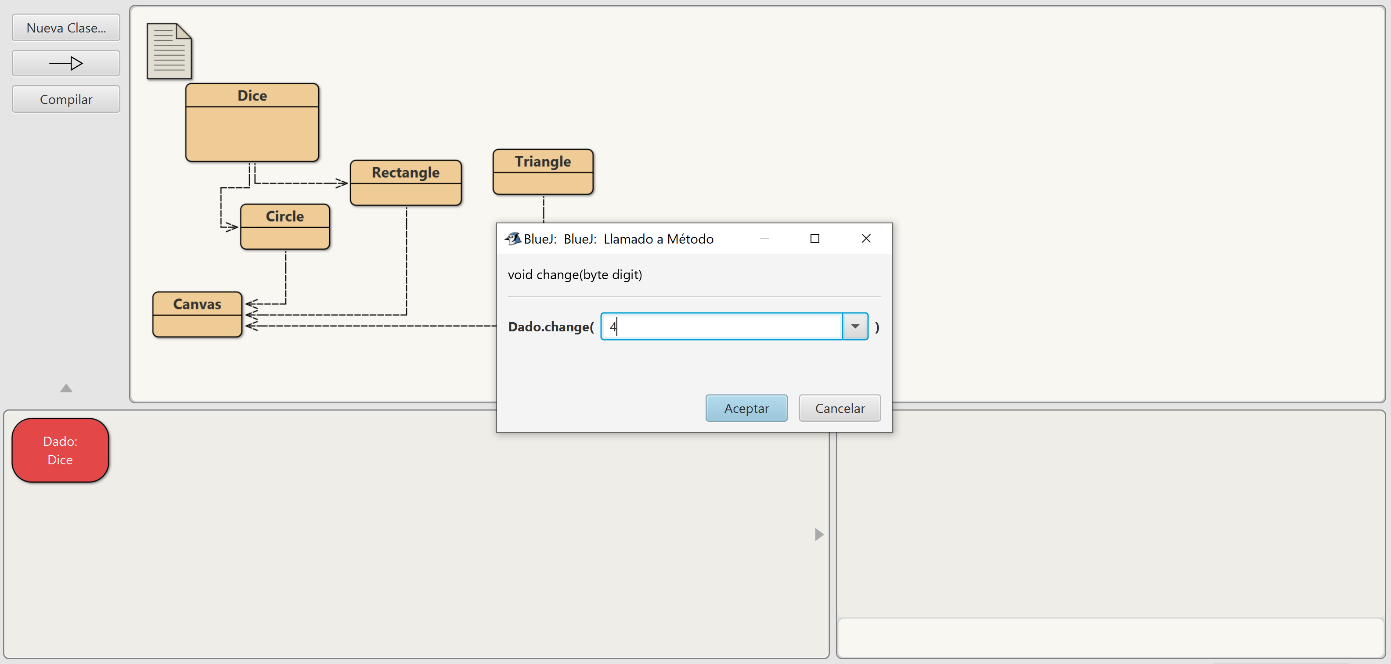
1. Clasificando:

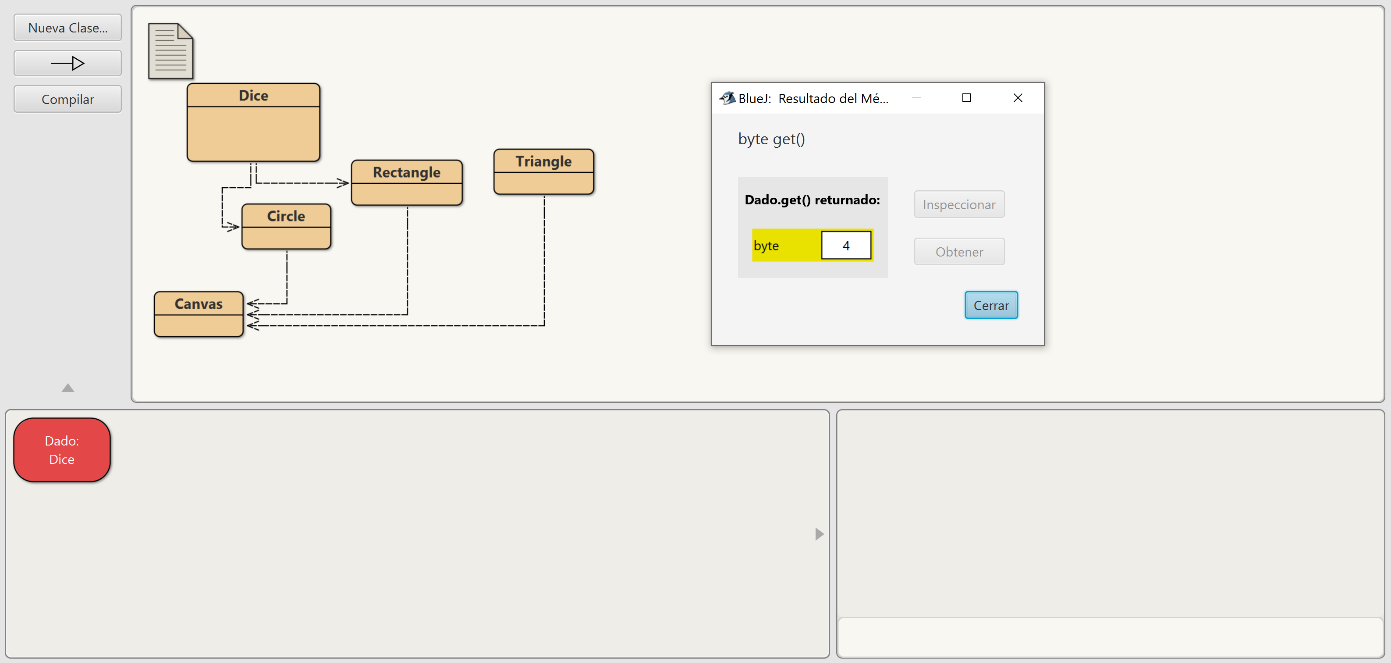
* Constructores:
* Analizadores y modificadores:

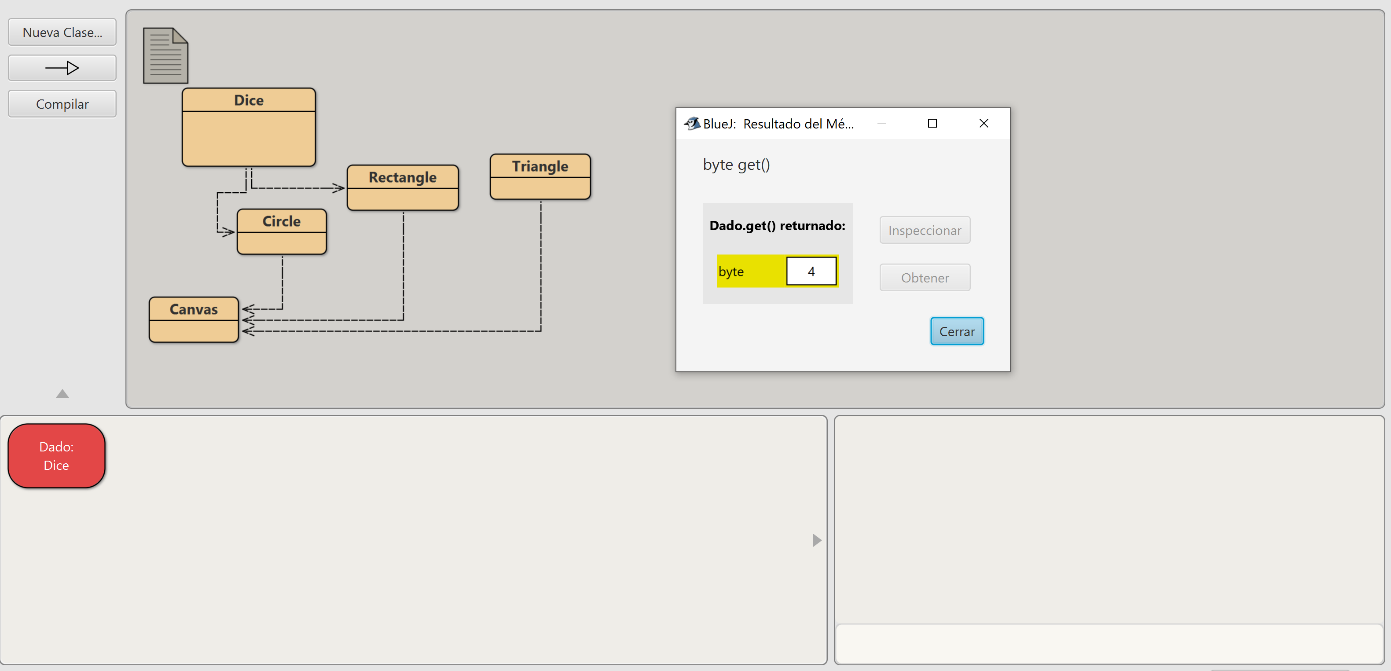
1. Clase **Dice:**

* Primer Mini ciclo:
* Segundo Mini ciclo:

Next ()

Change(digit)

.

Change ()

* Tercer Mini ciclo

Diagrama

Descripción generada automáticamentemakeVisible()

makeInvisible()

Diagrama

Descripción generada automáticamente

.

* Cuarto miniciclo

moveTo(x,y)

Diagrama

Descripción generada automáticamente

changeColor()

Diagrama

Descripción generada automáticamente

## F. Diseñando y codificando una nueva clase. DiceTaken

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

Texto

Descripción generada automáticamente con confianza media

* Para esta clase definimos un constructor.
* Un método para lanzar y mover el dado. ()
* que actualiza los colores de acuerdo con el enunciado.
* Un método que realiza las dinámicas de un juego exitoso.

1. Planifiquen la construcción definiendo algunos mini ciclos. Recuerden que al final de cada mini ciclo deben poder probar.

**Iterativo – Incremental Mini ciclos:**

1. Implementen la clase. Al final de cada mini ciclo realicen una prueba de aceptación. Capturen las pantallas relevantes.

* Primer mini ciclo

\_digit()

Diagrama

Descripción generada automáticamente

lanzar(i,j)

Diagrama

Descripción generada automáticamente

* Segundo mini ciclo.

Patrón de fondo

Descripción generada automáticamentePatrón de fondo

Descripción generada automáticamente

updateTable ()

Diagrama

Descripción generada automáticamente

* Tercer mini ciclo

isWinner()

Forma

Descripción generada automáticamente con confianza media

1. En la siguiente imagen se puede apreciar la clase *Dice* dentro de la interfaz de BlueJ

Diagrama

Descripción generada automáticamente

En total esta clase tiene dos extensiones de las clases *Rectangle* y *Circle.* Esto se debe a que el “cuerpo” de nuestro dado esta compuesto por un (1) rectángulo que y siete (7) círculos que indican el valor de la cara del dado.

Evidentemente no se muestran todos los círculos dentro de los modelos gráficos de dado que presentamos, estos están almacenado implícitamente en nuestro código.

Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

## RETROSPECTIVA

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

Hombre)

14,5 (Horas/Hombre)

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

Completo

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

The project is divided into iterations, nos ayudo a planificar el desarrollo de las clases.

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

El y

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

La matriz de instancias de la clase Dice.

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

Colaboración en el desarrollo del laboratorio. Mejorar las herramientas para trabajo en equipo.