# PROGRAMACIÓN ORIENTADA A OBJETOS

## Introducción. Clases y objetos.

## 2019-2

## Laboratorio 1/6

Integrantes:

Brayan Burgos

Daniel Alfonso

# Conociendo el proyecto shapes

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” ¿qué clases ofrece? ¿qué relaciones existen entre ellas?

* Las clases que ofrece son: Rectangle, Circle, Triangle, Canvas.
* Las clases Rectangle, Circle y Triangle se relacionan con Canvas

2. La documentación presenta las clases del proyecto y, en este caso, la

especificación de sus componentes públicos. De acuerdo con la documentación generada: ¿qué clases tiene el paquete shapes? ¿qué atributos tiene la clase Circle? ¿cuáles métodos ofrece la clase Circle para que la figura cambie (incluya

sólo el nombre)?

* Según la documentación las clases que tiene el Proyecto shapes son: Canvas, Circle, Triangle, Rectangle
* En la documentacion aparece solo un atributo: PI
* Los métodos que hacen que la figura de clase Circle cambie son: changeColor()

changeSize()

makeInvisible()

makeVisible()

3. En el código de cada clase está el detalle de la implementación. Revisen el Código de la clase Circle. Con respecto a los atributos: ¿cuántos atributos realmente tiene? ¿cuáles son privados y cuáles públicos? Con respecto a los métodos: ¿cuántos métodos tiene en total? ¿cuáles son privados? ¿Quienes usan los componentes privados?

* La clase circle tiene 6 atributos en total
* Privados: diameter, xPosition, yPosition, color, isVisible
* Publicos: PI
* La clase circle tiene 14 metodos en total
* Hay dos metodos privados draw() y erase() y estos son usados por algunos metodos públicos como: makeVisible(), makeInvisible(), moveHorizontal(), moveVertical(), , slowMoveHorizontal(), slowMoveVertical(), changeSize(), changeColor()

4. ¿Qué no se ve en la documentación? ¿por qué debe ser así?

En la documentación no se ven los atributos privados y los métodos privados. Debe ser asi porque estos métodos y atributos son solo usados por la clase correspondiente además de la integridad de los datos

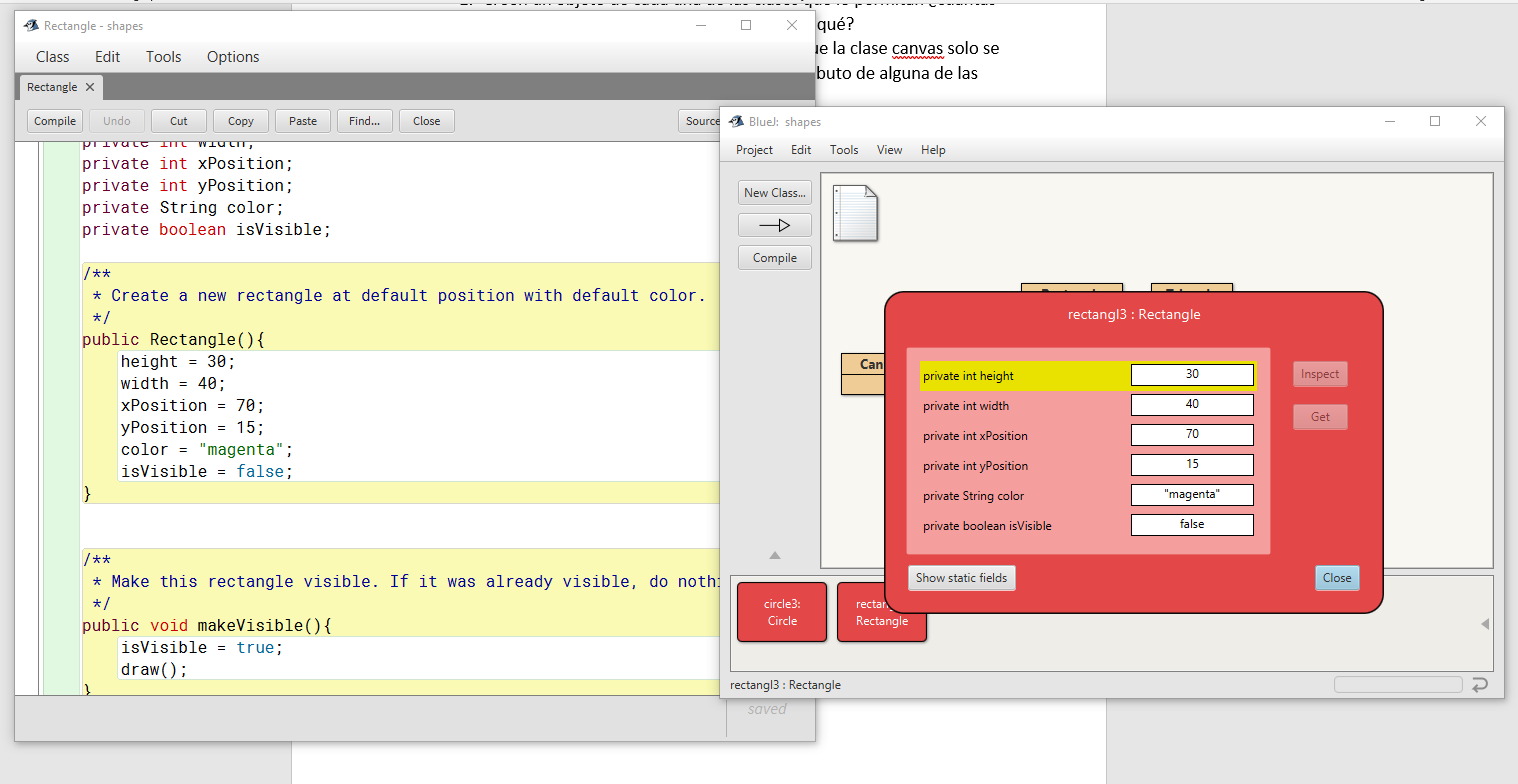
5. En el código de la clase Circle revisen el detalle del atributo PI. ¿qué se está indicando?  
Se indica que el atributo es público y este podrá ser usado por cualquier clase y estático ya que no se puede cambiar el valor de este valor constante

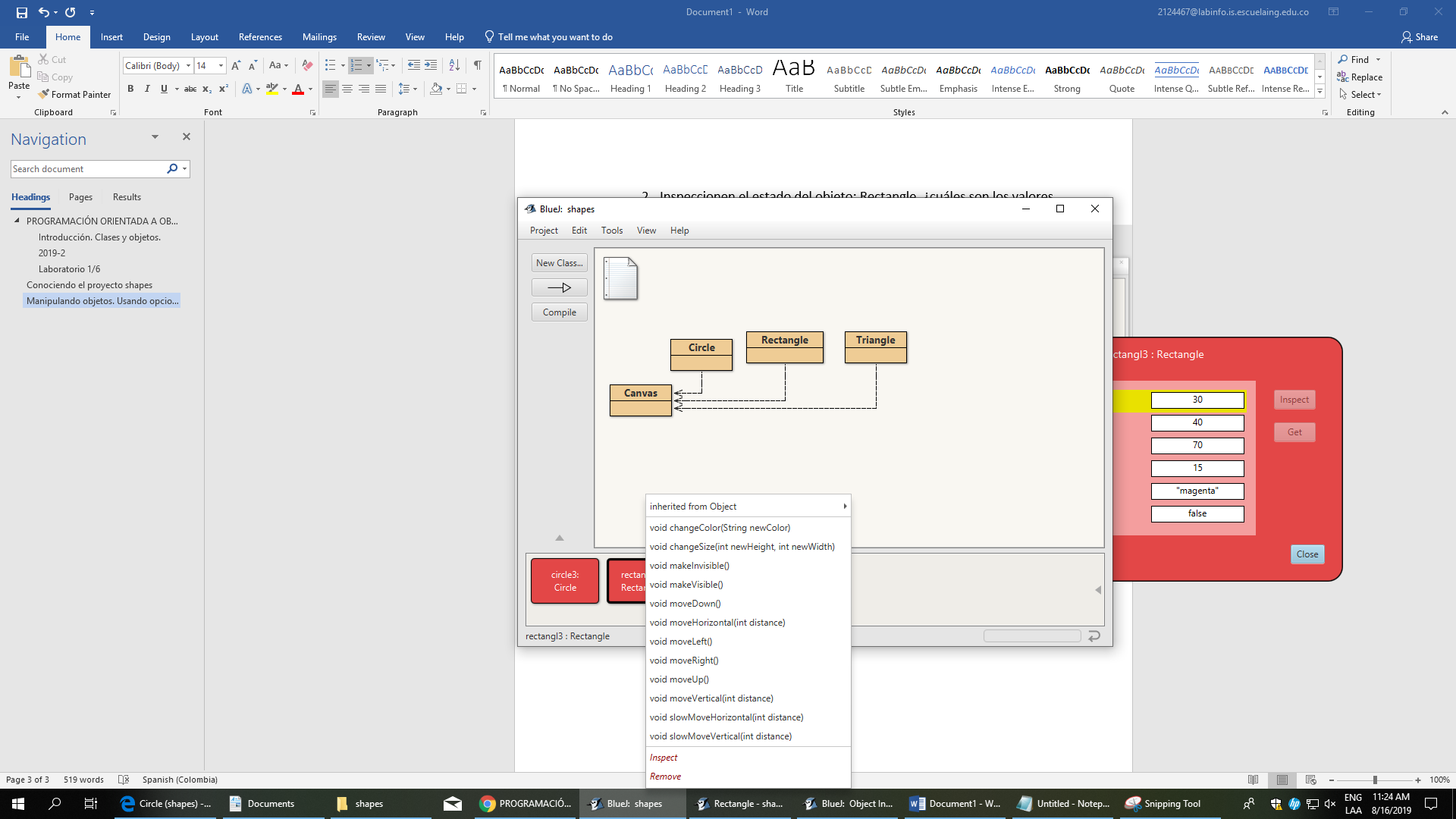
6. ¿Cuál dirían es el propósito del proyecto “shapes”?  
El propósito es construir figuras geométricas como: Circulo, Rectangulo y Triangulo, teniendo la posibilidad de modificar sus propiedades físicas como color, tamaño, posición, etc.

# Manipulando objetos. Usando opciones.

1. Creen un objeto de cada una de las clases que lo permitan ¿cuántas clases hay? ¿cuántos objetos crearon? ¿por qué?

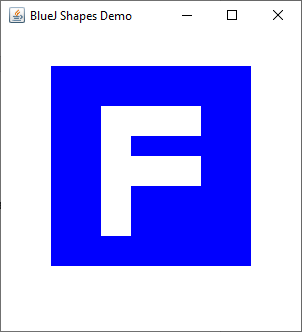
Hay 4 clases, se crearon 3 objetos debido que la clase canvas solo se encarga de dibujar las figuras y no es un atributo de alguna de las clases

1. Inspeccionen el estado del objeto: Rectangle, ¿cuáles son los valores de inicio de todos sus atributos? Capturen las pantallas  
   
2. Inspeccionen el comportamiento que ofrece el objeto: Rectangle. Capturen la pantalla. ¿por qué no aparecen todos los que están en el código?



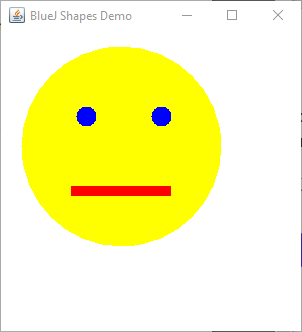
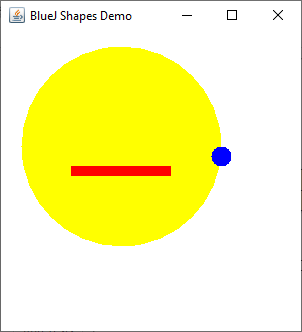
Los metodos que no aparecen en el comportamiento del objeto rectángulo son: draw() y erase() debido a que estos son metodos privados, los demás permiten cambiar los atributos del objeto

1. Construyan, con “shapes” sin escribir código, una propuesta de la imagen de su app favorito. ¿Cuántas y cuáles clases se necesitan? ¿Cuántos objetos se usan en total? Capturen la pantalla.



Se necesitan dos clases Canvas y Rectangle, además se usan 4 objetos(rectangle) en total

# Manipulando objetos. Analizando y escribiendo código.

1. Lean el código anterior ¿cuál es la figura resultante? Píntenla.  
   
2. Habiliten la ventana de código en línea escriban el código y para cada punto señalado indiquen: ¿cuántas variable existen? ¿cuántos objetos existen? ¿qué color tiene cada uno de ellos? ¿cuántos objetos se ven? Expliquen. Capturen la pantalla.  
   

Existen 4 variables, son: face, rEye, lEye y mouth

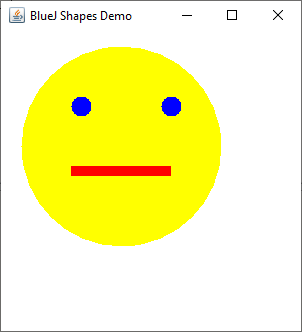
Existen 3 objetos

Colores de los objetos: face = “yellow”, rEye = “blue” y mouth = “red”

Se ven 3 objetos porque lEye es una variable de referencia hacia el objeto rEye, por esto mismo al cambiar la posicion de lEye, se mueve el objeto creado en rEye

1. Es igual la figura pintada en 1. igual a la figura capturada en 2. , ¿por qué?

Las figuras no son iguales debido a lo siguiente:   
Circle face;  
Circle rEye,lEye;  
Rectangle mouth;  
face= new Circle();  
mouth= new Rectangle();  
rEye= new Circle();  
lEye= new Circle(); // Cambiamos IEye = rEye ya que necesitamos un //nuevo objeto en lugar a una referencia   
face.changeSize(200);  
face.changeColor("yellow");  
face.makeVisible();  
mouth.changeSize(10,100);  
mouth.changeColor("red");  
mouth.moveVertical(120);  
mouth.makeVisible();  
rEye.changeSize(20);  
rEye.moveVertical(50);  
rEye.moveHorizontal(50);  
rEye.makeVisible();  
lEye.changeSize(20);  
lEye.moveVertical(50);  
lEye.moveHorizontal(140);  
lEye.makeVisible();

Y como resultado obtenemos lo siguiente.   


# Extendiendo clases (Código anexado en Circle.java)

1. 1. Desarrollen en Circle el método growUp(times) (que hace que crezca lentamente el número dado de veces). ¡Pruébenlo! grow up
2. Desarrollen en Circle el método area(). ¡Pruébenlo!
3. Desarrollen en Circle el método jump() (que hace que salte a la derecha). ¡Pruébenlo!
4. Generen nuevamente la documentación y revise la información de estos nuevos

métodos. Capture la pantalla.



# SLOT MACHINE

1. ¿Cuál es la probabilidad de ganar en esta máquina? Explique su respuesta.  
   La probabilidad de ganar en esta máquina es de
2. Clasifiquen los métodos en: constructores, analizadores y modificadores.

* Métodos Constructores: SlotMachine
* Métodos Analizadores: isWinningState(), percentageOfWinningStates()
* Métodos Modificadores: reset(), pull(), pull(times : int), makeVisible(), makeInvisible(), move(horizontal: int, vertical: int).

1. Desarrollen la clase SlotMachine considerando los miniciclos. Al final de cada miniciclo realicen una prueba. Capturen las pantallas relevantes.
2. ¿Cuál es el porcentaje de estados ganadores después de hacer 1, 10, 100 y 1000 jugadas? Presente un análisis de los datos.

* El porcentaje de estados ganadores después de hacer 1 intento es de 0%
* El porcentaje de estados ganadores después de hacer 10 intentos es de 10%
* El porcentaje de estados ganadores después de hacer 100 intentos es de 7%
* El porcentaje de estados ganadores después de hacer 1000 intentos es de 5.3%

El porcentaje de victoria en el Slot Machine es variado ya que es aleatorio el resultado de las casillas que salgan, no es proporcional a los intentos cada vez se pueden obtener resultados de porcentajes de victoria muy diferentes

# 