**Laboratorio #1 – Programación orientada a objetos (SHAPES)**

1. **Conociendo el Proyecto Shapes:**

**¿Qué clases ofrece?** Ofrece 4 clases las cuales son Circle, Triangle, Rectangle, Canvas.

**¿Qué relaciones existen entre ellas?** Todas estan relacionadas con el Canvas. Pero entre Triangle, Rectangle y Circle no existe alguna relación.

**¿Qué clases tiene el paquete Shapes?** Canvas, Circle, Rectangle, Triangle

**¿Qué atributos tiene la clase Circle?** PI

**¿Cuántos métodos ofrece la clase Cricle?** 12

**¿Cuáles métodos ofrece la clase Circle para que la figura cambie?** changeColor, changeSize, makeInvisible, makeVisible, moveDown, moveHorizontzal, moveLeft, moveRight, moveUp, moveVertical, slowMoveHorizontal, slowMoveVertical

**¿Cuántos atributos realmente tiene?** 6 – diameter, xPosition, yPosition, color, isVisible, PI

**¿Cuáles atributos describlen la forma del circulo?**

Privados: diameter, color, isVisible

**¿Cuántos métodos tiene en total?** 14

**¿Quiénes usan los metodos privados?** Canvas

**¿Qué no se ve en la documentación?** Los atributos y métodos privados.

**¿Por qué debe ser así?** Por la protección a los datos.

**¿Qué se está indicando al decir que es static (PI)?** Declara que el atributo no pertenece a la clase, y se puede acceder a el sin necesidad de instanciar el objeto.

**¿Cómo decimos que PI es una constante?** Declarando el atributo con la palabra reservada final

**¿Qué se esta indicando al decir que el atributo “diameter” es int?** Al declarar un atributo del tipo int, se dice que el valor asignado a la variable debe ser un entero.

**¿De qué tipo podría ser “diameter” en caso de que los diametros siempre fueran menor a 100?** Podria ser de tipo byte (-128, 127) >> Rango

**¿De qué tipo podría ser “diameter” en caso de que los diametros fueran mayores a 220000000?** El tipo de dato int almacena un valor de por lo tanto, alcancaria. Pero si se requiere otra podría ser de tipo long.

**¿Qué restricción tendría?** Un valor menor a

**¿Cuál es el propósito del proyecto?** El proyecto permite el acercamiento a los objetos y poder programarlos para una función especifica.

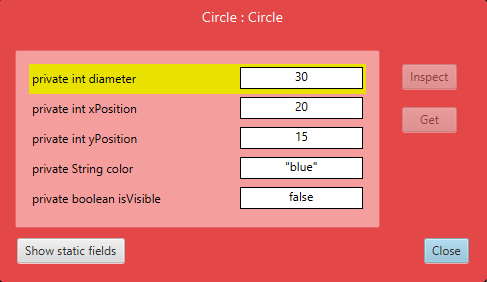
1. **Manipulando Objetos**

**¿Cuántas clases hay?** 4

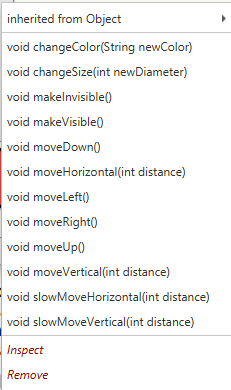
**¿Cuántos objetos crearon?** 3

**¿Por qué?** Porque el constructor del Canvas es privado.

**¿Cuáles son los valores de inicio de todos sus atributos?**



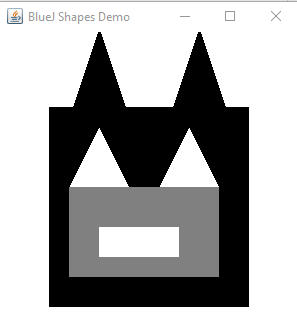
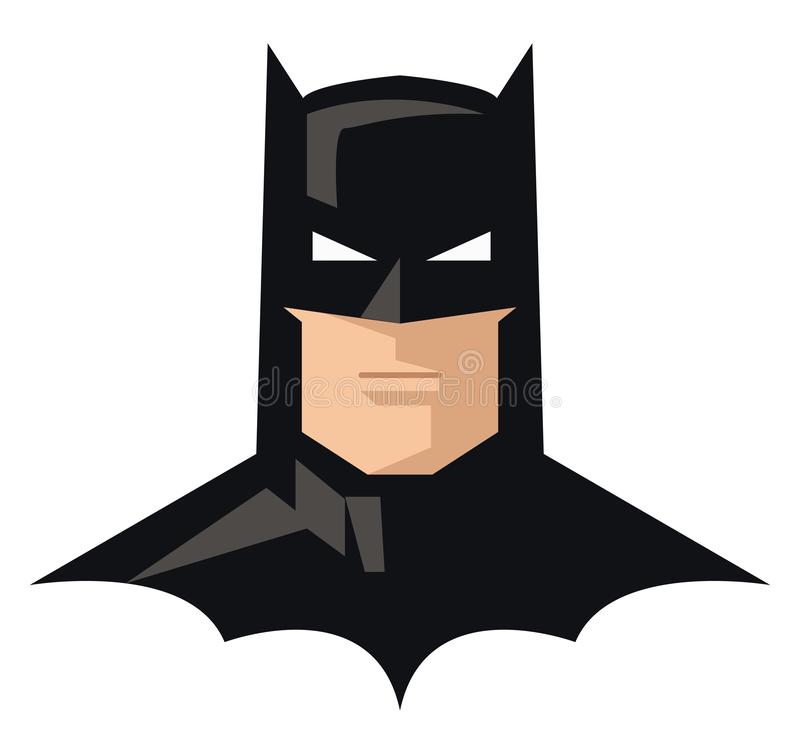
**¿Por qué no aparecen todos los que están en el código?** Porque son privados



**¿Cuántas y cuales clases se necesitan?** 2 – Triangle y Rectangle

**¿Cuántos objetos se usan en total?** 7

**Realidad en lego** **Expectativa**

**¿Cuál es la figura resultante?**



**¿Cuántas variables existen?** 3

**¿Cuántos objetos existen?** 2

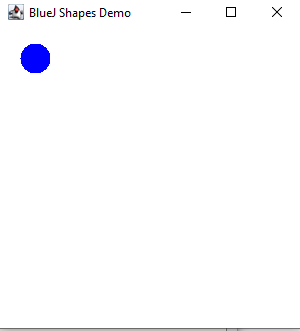
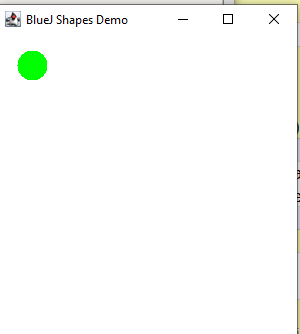
**¿Qué** **color tiene cada uno de ellos?** Todos son negros

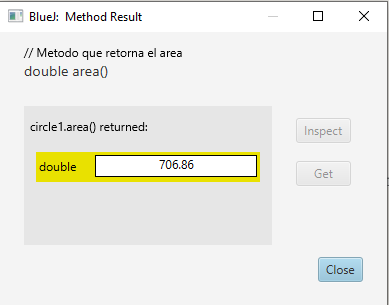
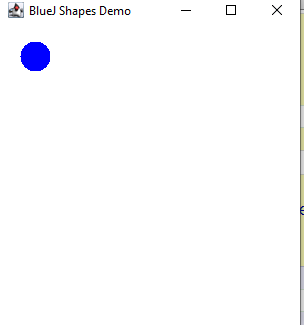
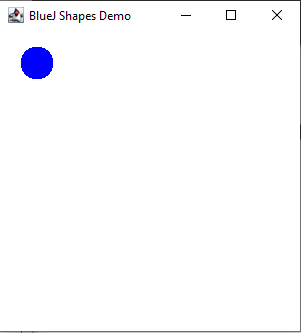
**¿Cuántos objetos se ven?** 2

Se ven solamente dos objetos, pero uno de ellos este sobrepuesto en la cara con el mismo color, por ende, no se visualiza correctamente.

**Extendiendo Clases**

**1.**



1. 
2.  **<<<**Antes – Despues>>>
3. 

**SELF – ASSEMBLY. Pieces**

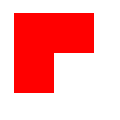
1. **Implementando una nueva clase**
   1. **Analizadores:** take(), put()
   2. **Modificadores:** reflect (), rotate(), resize(), translate()
2. **Miniciclos**
   1. **Constructor**



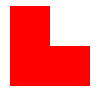
* 1. **Take()**



**Put()**



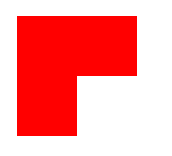
* 1. **Reflect()**



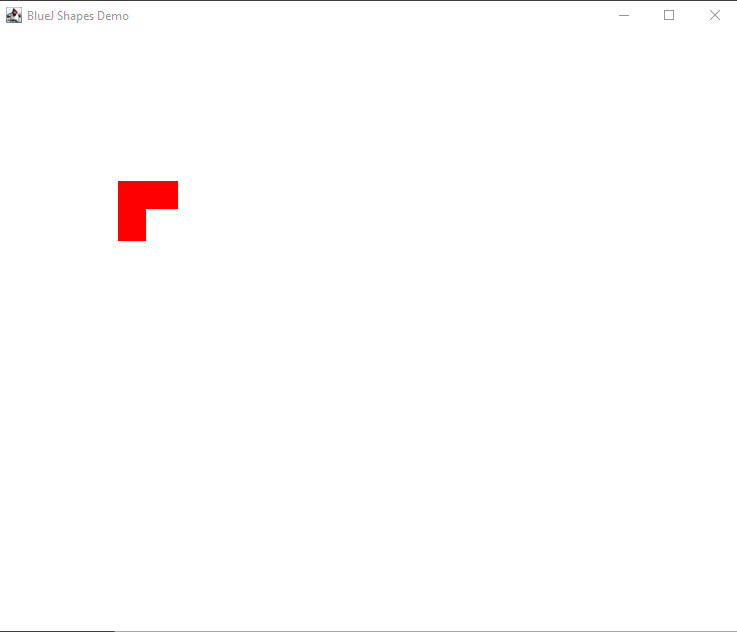
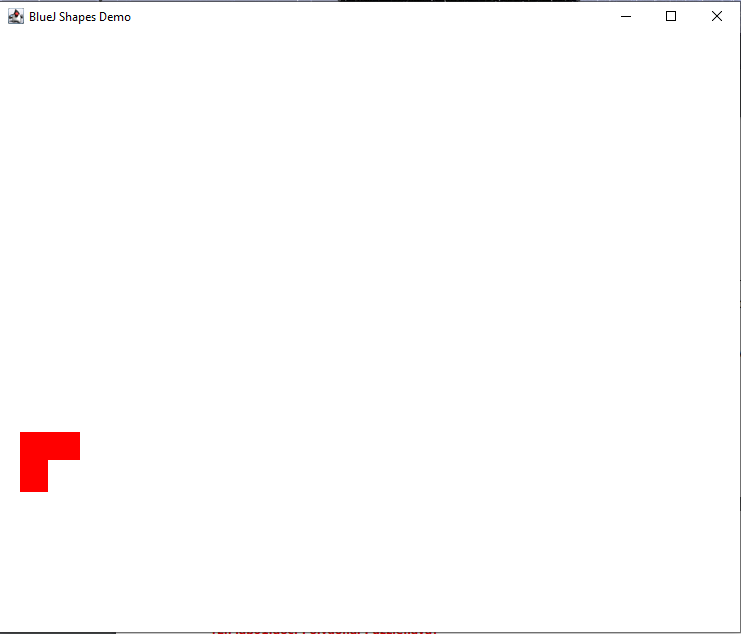
**Rotate()**



* 1. **Resize()**



**Translate()**



**Polygonal Puzzle**

1. **Astah**
2. **Metodos**:
   1. \_PoligonalPuzzle(width: int, height: int)
   2. createPiece(color: String, points: int[ ][ ])
   3. createPiece()
   4. deletePiece()
   5. take(color: String)
   6. untake()
   7. rotate()
   8. reflect(d: char)
   9. put()
   10. translate(dx: int, dy: int)
   11. resize(percentage: int)
   12. showMessage(msg: String)
   13. sort()
   14. initializeColors()
3. **Mini ciclos**
   1. \_PoligonalPuzzle, initializeColors, showMessage
   2. createPiece, createPiece, sort
   3. take, untake, put, deletePiece
   4. rotate, reflect, translate, resize
4. **Explicación**:
   1. Se crea un Puzzle indicándole las medidas que va llevar, luego se crea un método encargado de inicializar la lista de colores que se van a poder usar para las fichas, y finalmente un método que permite la comunicación directa con el usuario, indicándole cada una de sus acciones y guiándolo en todo el proceso.
   2. En el segundo mini ciclo se definen los métodos encargados de crear las piezas visualmente, uno de ellos lo hace con las medidas dadas por el usuario, mientras que el otro lo hace totalmente aleatorio, tomando en cuenta el tamaño del puzle para evitar que se generen fichas mas grandes que éste. Adicionalmente un método que organiza las fichas visualmente, por si el usuario desea crear mas de una ficha.
   3. En el tercer mini ciclo se definen los métodos que permiten tomar, dejar, poner y eliminar una pieza antes creada. Borrar la pieza restaura el color a disponible.
   4. Luego se definen los modificadores, que permiten rotar, trasladar o modificar el tamaño de una de las piezas. Cabe recalcar que para tomar estas acciones debe existir una pieza seleccionada.
   5. Y por último se crea un método informa al usuario si el rompecabezas se encuentra armado. Esta comprueba cada vez que una pieza es colocada.
5. **Metodo nuevo:**
   1. deletePiece();
      1. Permite eliminar una pieza y restaurar el color.

**RETROSPECTIVA**

1. 25 / Johann – 25 / Espinosa
2. Completo al 90%, la parte donde la maquina decide que ficha crear dada su área es complicada
3. Pair programming – Ya que nos permite apoyarnos en conocimientos mutuos.
4. Hacer rotar o reflejar la figura, porque requiere de lógica para entender la posición que deben tomar los rectángulos, ya que cada pieza contiene 1 o más rectángulos.
5. Rotar la figura y reflejarla, tomar individualmente cada uno de los rectángulos y especificarle las coordenadas cartesianas que debian tomar.
6. Entender y lograr hacer el laboratorio de la mejor manera posible. Nos comprometemos a mejorar como equipo.