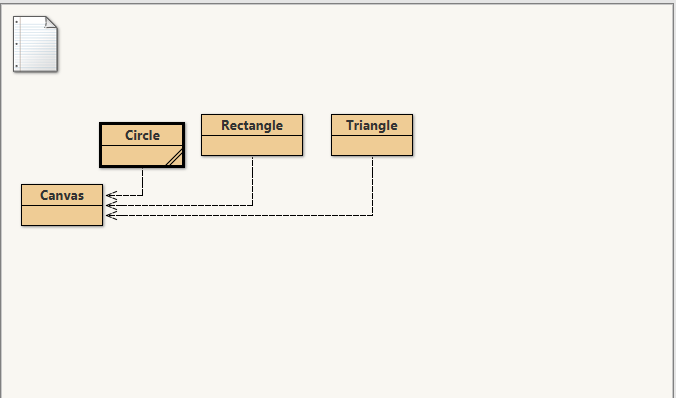
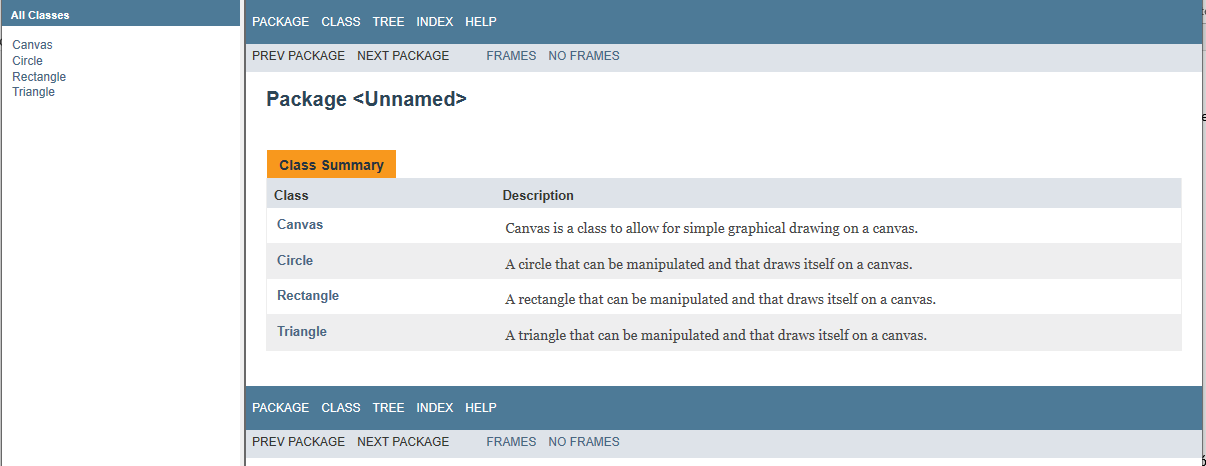
**LABORATORIO 1**

* **EDUARD ARIAS**

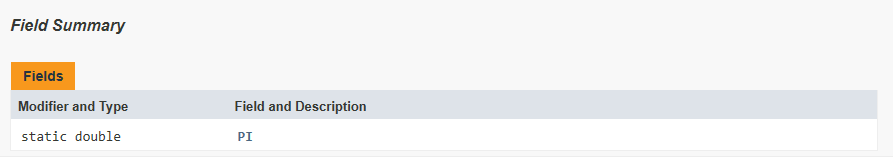
1. **CONOCIENDO EL PROYECTO SHAPES**
   1. El diagrama de clases permite visualizar las clases de un artefacto software y las relaciones entre ellas. Considerando el diagrama de clases de “shapes”, ¿qué clases ofrece? ¿qué relaciones existen entre ellas?



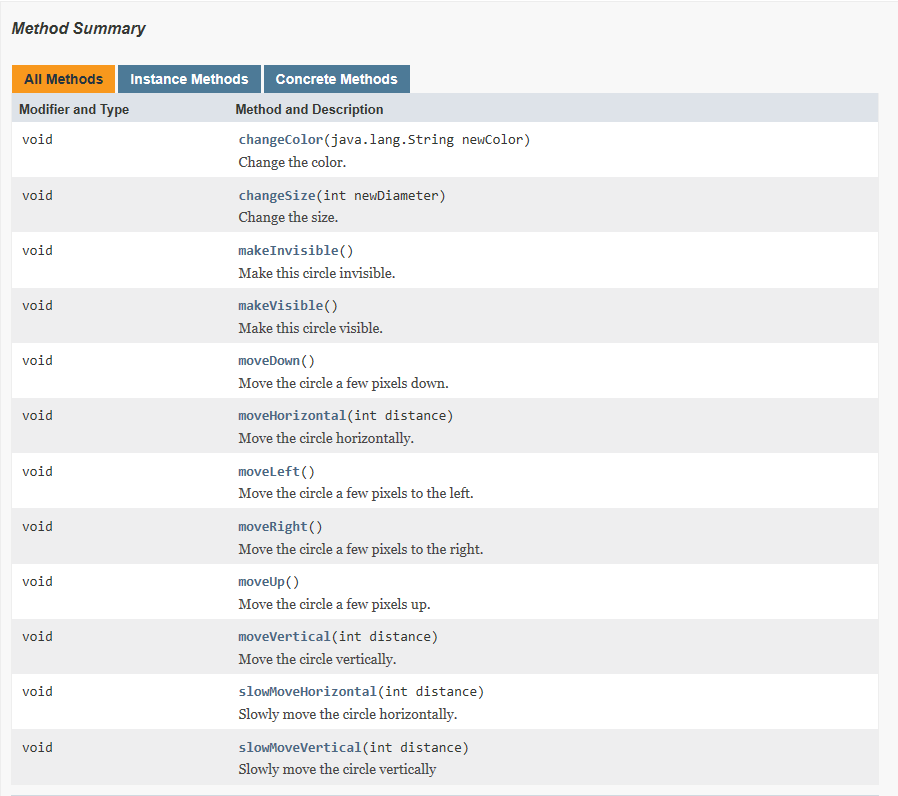
* + 1. El proyecto ofrece las clases Circle, Rectangle, Triangle y Canvas.
    2. Las clases Circle, Rectangle y Triangle utilizan el Canvas para dibujarse en pantalla.
  1. La documentación presenta las clases del proyecto y, en este caso, la especificación de sus componentes públicos. De acuerdo con 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)?
     1. La documentación muestra las cuatro clases: Circle, Rectangle, Triangle y Canvas.



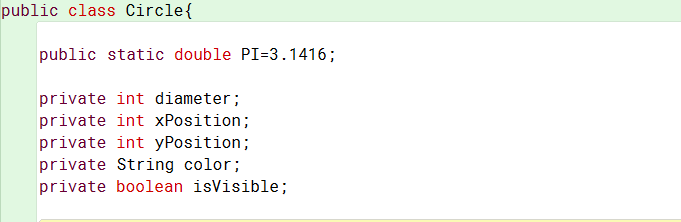
* + 1. **Atributos de Circle:**



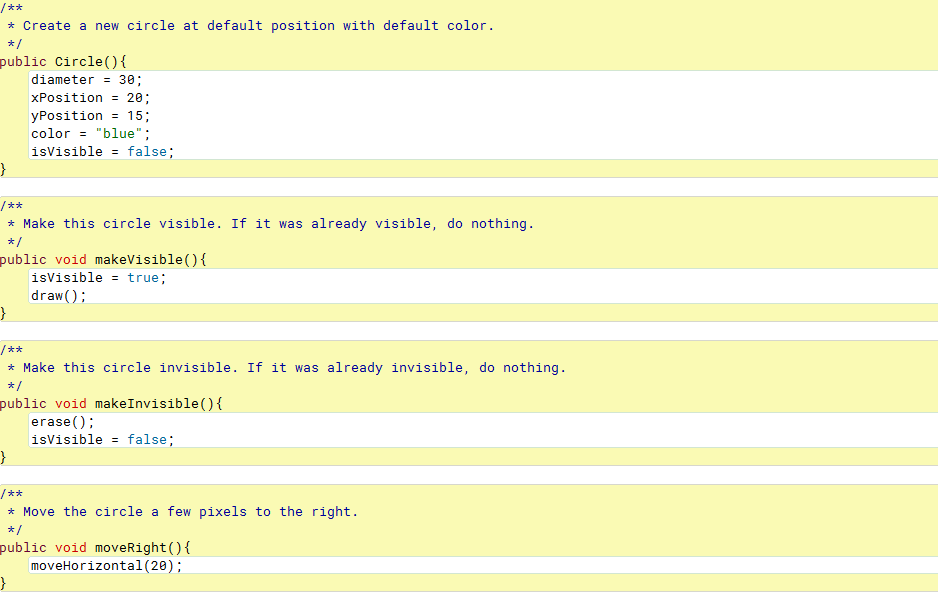
* + 1. **Métodos de Circle:**

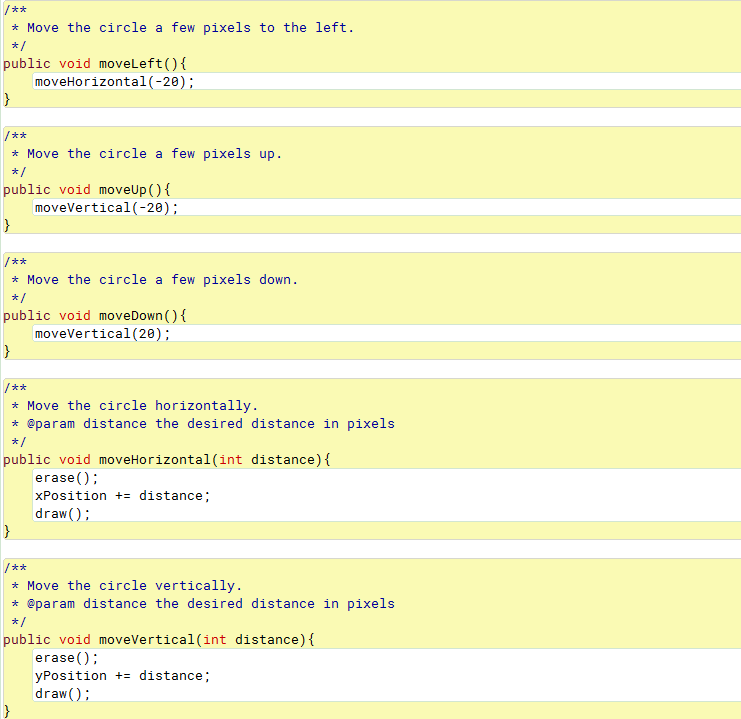


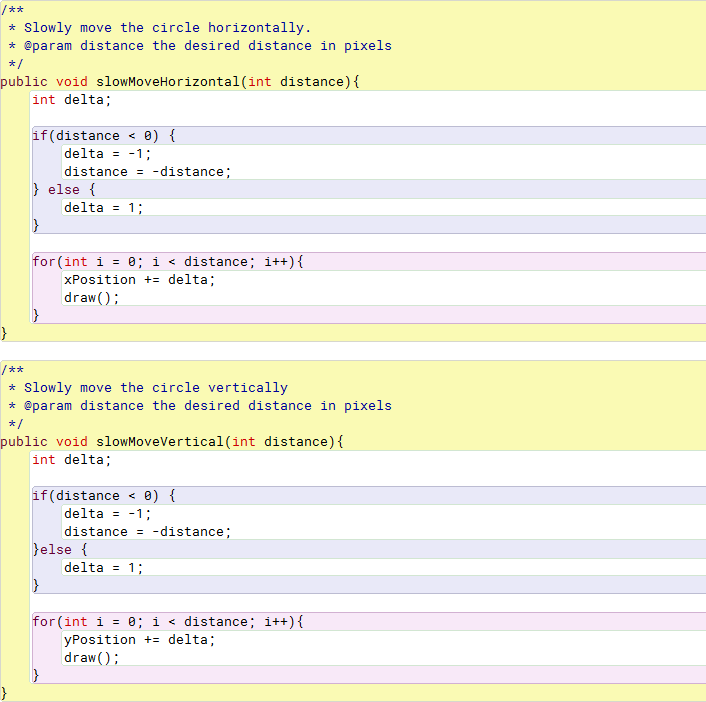
* 1. En el código de cada clase está el detalle de la implementación. Revisen el código de la clase Circle. Con respecto a los atributos: ¿cuántos atributos realmente tiene? ¿cuáles son privados y cuáles públicos? Con respecto a los métodos: ¿cuántos métodos tiene en total? ¿cuáles son privados? ¿Quienes usan los componentes privados?
     1. **Atributos del Código:** tiene en total 6 atributos



* + 1. Los atributos diameter, xPosition, yPosition, color y isVisible son privados. El único que es público es la constante PI.
    2. **Métodos del Código:** tiene en total 15 métodos



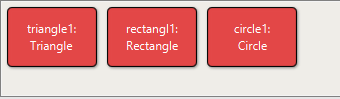




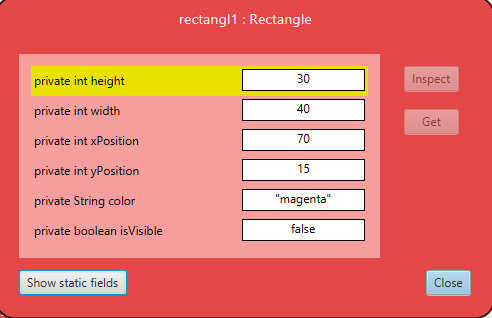


* + 1. Los métodos privados son: draw y erase.
    2. Los métodos internos de la clase son los únicos que utilizan draw y erase para dibujar la figura en pantalla.
  1. ¿Qué no se ve en la documentación? ¿por qué debe ser así?
     1. La documentación no muestra las características privadas de la clase debido a que no es lo que se espera que un agente externo vea de la clase.
  2. En el código de la clase Circle revisen el detalle del atributo PI. ¿qué se está indicándo?
     1. Se está indicando que es un atributo público, de tipo double y que le pertenece a la clase.
  3. ¿Cuál dirían es el propósito del proyecto “shapes”?
     1. Aprender a utilizar objetos dibujando figuras en la pantalla.

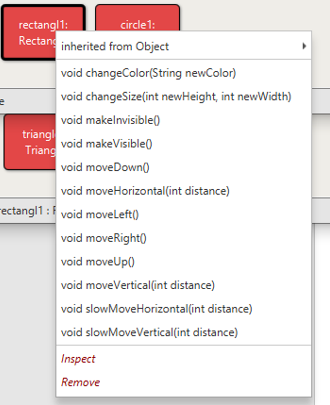
1. **MANIPULANDO OBJETOS. USANDO OPCIONES.**
   1. Creen un objeto de cada una de las clases que lo permitan. ¿cuántas clases hay? ¿cuántos objetos crearon? ¿por qué?



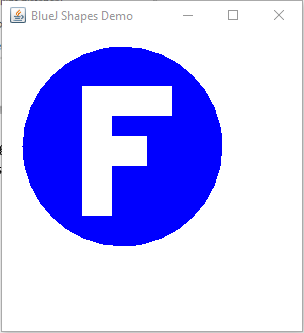
* + 1. Hay 4 clases, pero solo se pueden crear 3 objetos. Esto debido a que la clase Canvas no es instanciable como los demás; lo que hizo fue retornar una referencia.
  1. Inspeccionen el estado del objeto: Rectangle, ¿cuáles son los valores de inicio de todos sus atributos? Capturen las pantallas



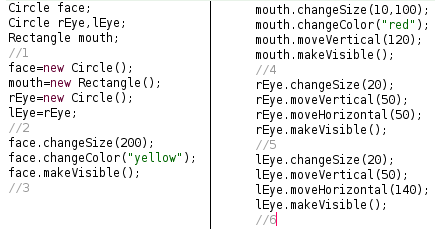
* 1. Inspeccionen el comportamiento que ofrece el objeto: Rectangle. Capturen la pantalla. ¿por qué no aparecen todos los que están en el código?



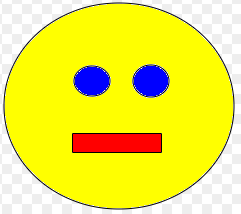
* 1. Construyan, con “shapes” sin escribir código, una propuesta de la imagen de su app favorito. ¿Cuántas y cuáles clases se necesitan? ¿Cuántos objetos se usan en total? Capturen la pantalla.
     1. Se necesitaron las clases Circle y Rectangle**.**
     2. Se utilizaron 4 objetos en total.



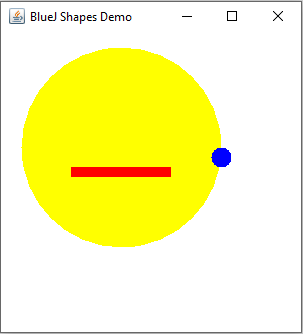
1. **MANIPULANDO OBJETOS. ANALIZANDO Y ESCRIBIENDO CÓDIGO.**



* 1. Lean el código anterior ¿cuál es la figura resultante? Píntenla.

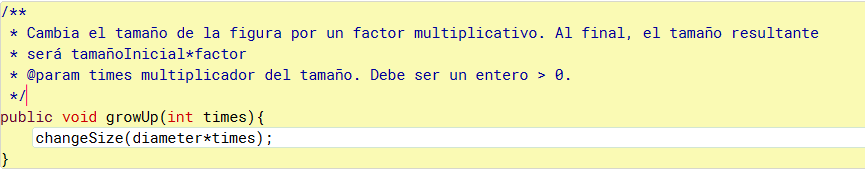


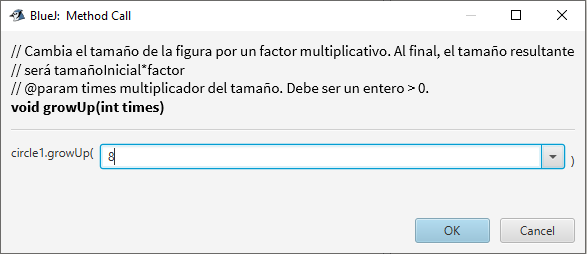
* 1. Habiliten la ventana de código en línea, escriban el código y para cada punto señalado indiquen: ¿cuántas variable existen? ¿cuántos objetos existen? ¿qué color tiene cada uno de ellos? ¿cuántos objetos se ven? Expliquen. Capturen la pantalla.
     1. + Hay tres variables de tipo Circle y una de tipo Rectangle.
        + No existen objetos hasta el momento.
     2. + Hay 3 objetos en memoria: dos Circle y un Rectangle.
        + Tienen el color por defecto, pero no se ven.
     3. + El objeto face cambia su color a amarillo y su tamaño a 200.
        + Solo se puede ver el círculo de face.
     4. + El objeto mouth es ahora de 10 de alto por 100 de largo y su color es rojo.
        + El objeto mouth se mueve 120 pixeles hacia abajo.
        + Son visibles mouth y face.
     5. + El objeto rEye cambia su tamaño a 20.
        + El objeto rEye se mueve 50 pixeles hacia abajo y 50 hacia la derecha.
        + Son visibles face, mouth y rEye (y lEye).
     6. + El objeto lEye cambia su tamaño a 20.
        + El objeto lEye se mueve 50 pixeles hacia abajo y 140 hacia la derecha.
        + Son visibles face, mouth y rEye (y lEye).

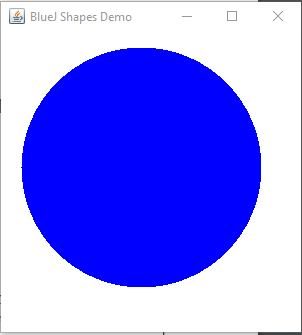
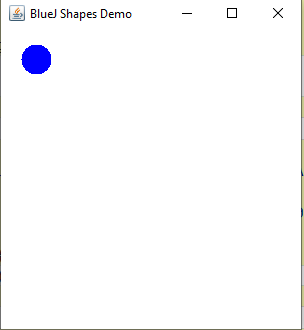


* 1. ¿Es igual la figura pintada en 1 que a la figura capturada en 2? ¿por qué?
     1. No, debido a que en el punto 2 del código no se está creando un nuevo círculo, sino que se asigna la referencia a uno ya creado. Esto hace parecer que las variables rEye y lEye son objetos distintos, pero en realidad apuntan al mismo objeto en memoria. Al llegar al punto 6 el código mueve el objeto de rEye, estropeando la figura.

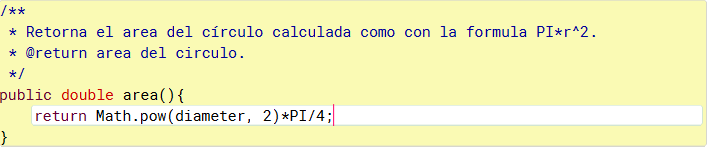
1. **EXTENDIENDO CLASES.**
   1. Desarrollen en Circle el método growUp(times) (que hace que crezca lentamente el número dado de veces). ¡Pruébenlo! grow up

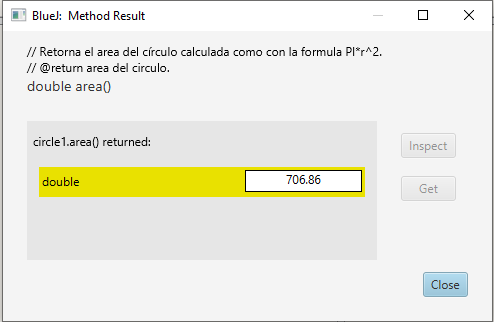




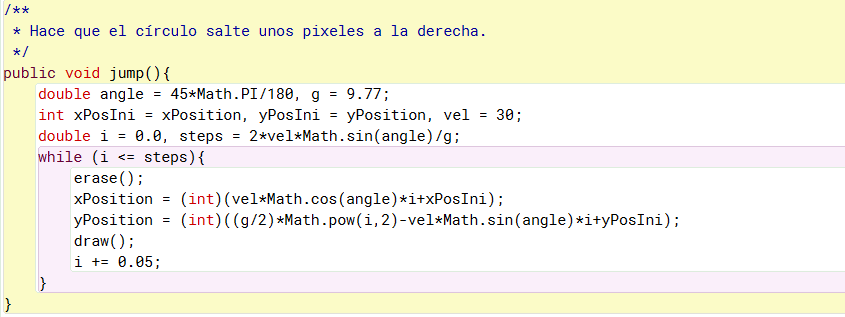


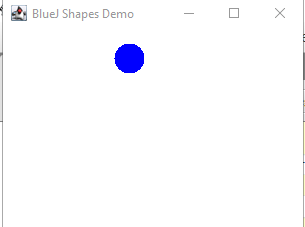
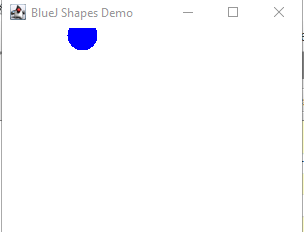
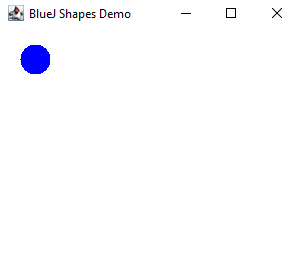
* 1. Desarrollen en Circle el método area(). ¡Pruébenlo!



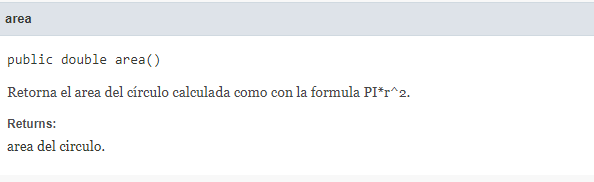


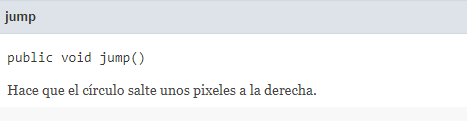
* 1. Desarrollen en Circle el método jump() (que hace que salte a la derecha). ¡Pruébenlo!

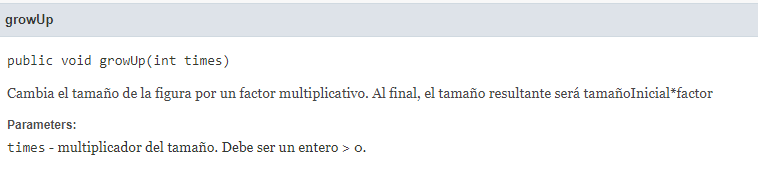




* 1. Generen nuevamente la documentación y revise la información de estos nuevos métodos. Capture la pantalla.

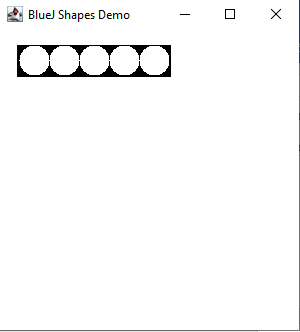
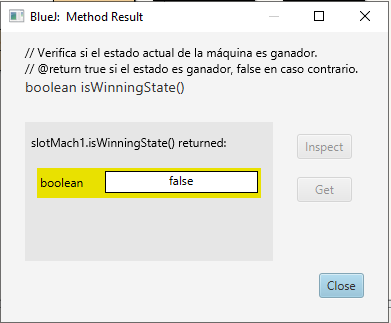
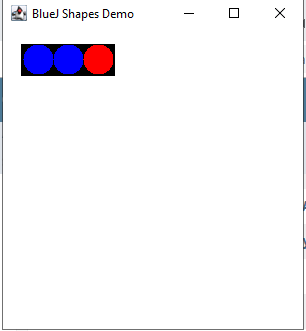




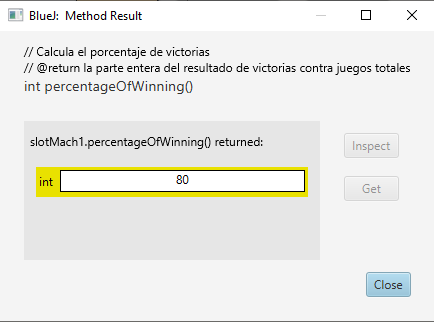


**SLOT MACHINE**

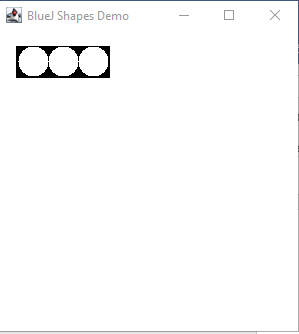
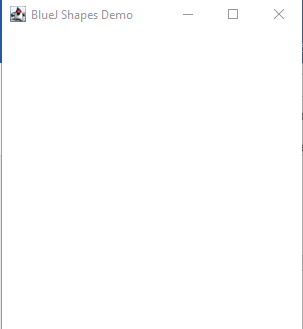
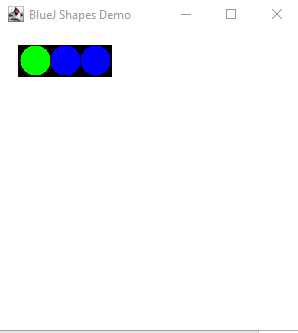
1. **IMPLEMENTANDO UNA NUEVA CLASE. SlotMachine.**
   1. ¿De qué tipo de elementos es el tragamonedas? ¿Cuáles son los diferentes elementos?
      1. Se hará la representación de la máquina con unos rectángulos (incluyendo los espacios de las frutas). Para representar a las frutas se usarán círculos de diferentes colores.
   2. ¿Cuál es la probabilidad de ganar en esta máquina? Explique su respuesta.
      1. Primero se calcula la cantidad de eventos posibles. Como hay espacios disponibles y hay elementos (como lo dice el enunciado) entonces la cantidad de eventos posibles es .
      2. Después se calcula la cantidad de eventos ganadores. Como hay una fila ganadora por cada elemento y hay elementos entonces hay eventos ganadores posibles.
      3. Usando el teorema de probabilidad clásica, sabemos que la probabilidad de que un evento suceda es . Reemplazando esa información con la anteriormente conocida, se calcula que hay una probabilidad de , que es lo mismo que .
      4. Por ejemplo, para el caso , la probabilidad es .
   3. Clasifiquen los métodos en: constructores, analizadores y modificadores.
      1. **Constructores:**
         * SlotMachine
      2. **Analizadores:**
         * isWinningState
         * percentajeOfWinning
      3. **Modificadores:**
         * Pull
         * Pull(times)
         * Reset
         * makeInvisible
         * makeVisible
         * move
   4. Desarrollen la clase SlotMachine considerando los miniciclos. Al final de cada miniciclo realicen una prueba. Capturen las pantallas relevantes.
      1. miniCiclo 1:

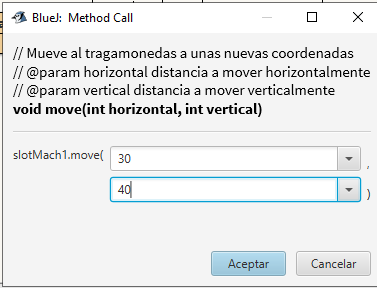
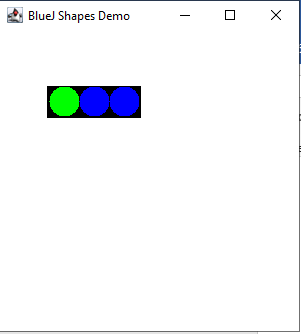
 

* + 1. miniCiclo 2:



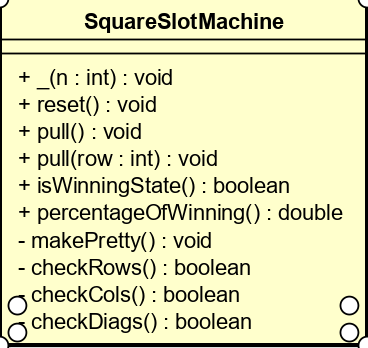
* + 1. miniCiclo 3:

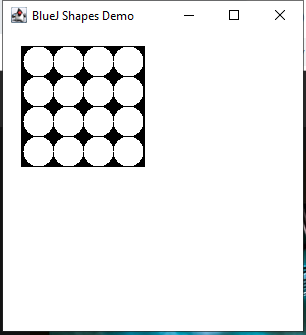
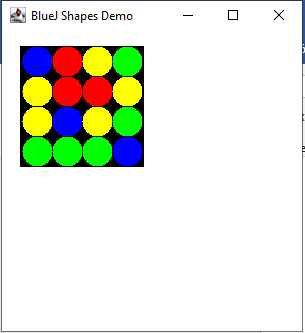
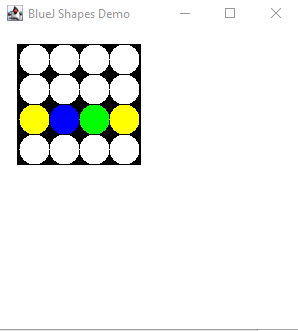
 

* 1. ¿Cuál es el porcentaje de estados ganadores después de hacer 1, 10, 100 y 1000 jugadas? Presente un análisis de los datos considerando la respuesta dada en 2.

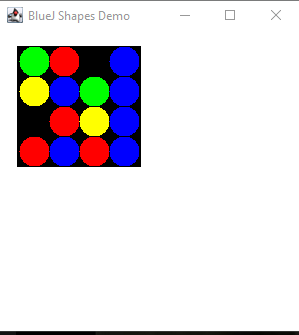
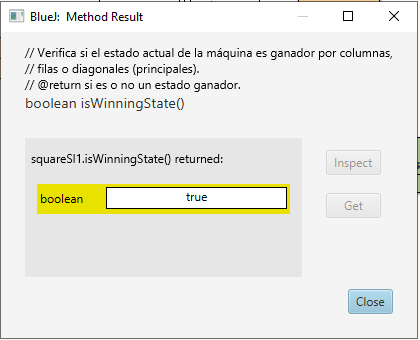
1. **DEFINIENDO Y CREANDO UNA NUEVA CLASE. SquareLotMachine**
   1. Diseñen la clase SquareSlotMachine, es decir, definan los métodos que debe ofrecer.

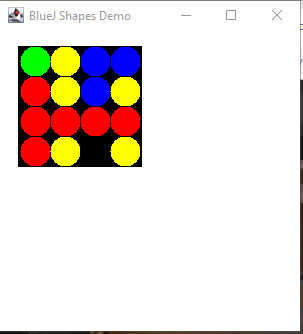
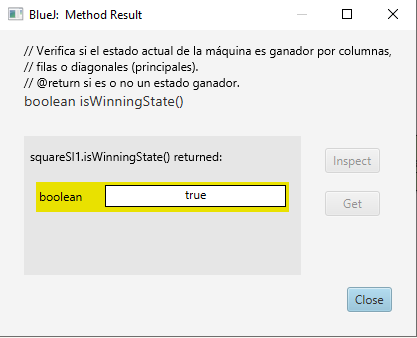


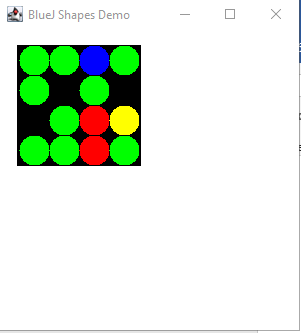
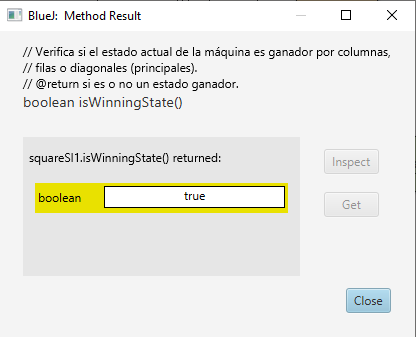
* 1. Planifiquen la construcción considerando algunos miniciclos.
     1. Miniciclo 1:
        + Crear
        + makePretty()
        + Pull()
        + Pull(row)
     2. MiniCiclo 2:
        + isWinningState()
        + checkRows()
        + checkCols()
        + checkDiags()
        + reset()
     3. Miniciclo 3:
        + percentageOfWinning()
        + Mensajes de error
  2. Implementen la clase. Al final de cada miniciclo realicen una prueba de aceptación. Capturen las pantallas relevantes.}
     1. Miniciclo 1:

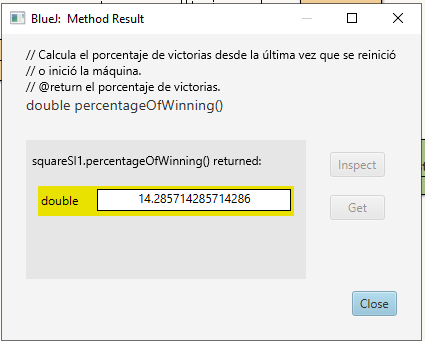
* + 1. Miniciclo 2:

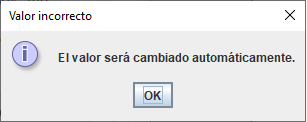
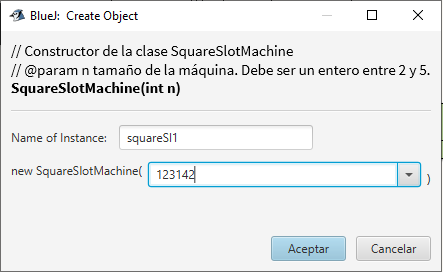
 

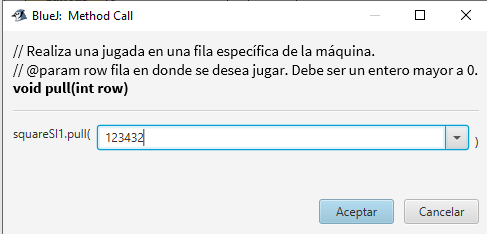
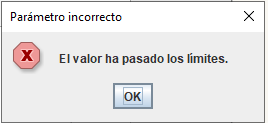
 

* + 1. Miniciclo 3:





* 1. Indiquen las extensiones necesarias para reutilizar la clase SlotMachine. Explique.
     1. Básicamente SquareSlotMachine se compone de varias filas de SlotMachine, por lo que en lugar de usar una matriz de círculos se podría haber untilizado un ArrayList de SlotMachine. Sin embargo, se tendría que modificar la accesibilidad para que SquareSlotMachine tuviera mejor control sobre sus elementos.
     2. Otra idea sería crear una clase general y extender SlotMachine de esta, aprovechando que SlotMahcine es solo un tragamonedas con dimensión 1\*n y SquareSlotMachine de n\*n.
  2. Propongan un nuevo método para enriquecer el juego.