

PROGRAMACIÓN ORIENTADA A OBJETOS

Construcción. Clases y objetos

2021-1

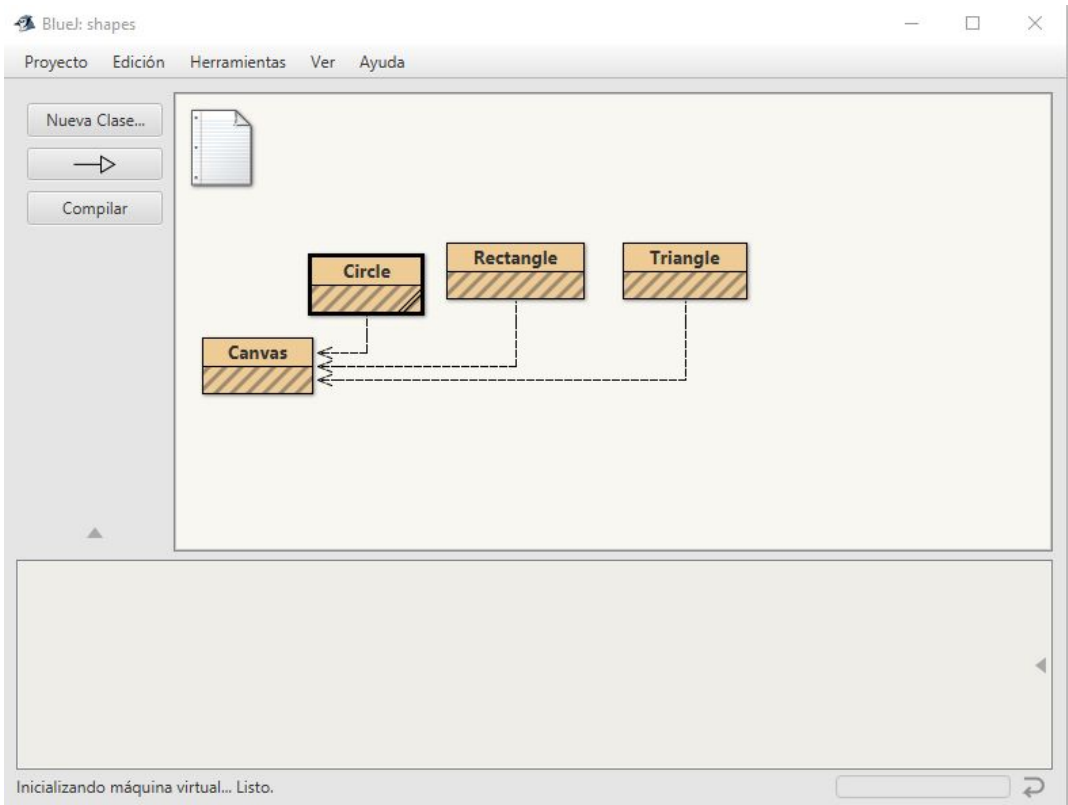
Laboratorio 1

Orjuela-Macana

SHAPES

Conociendo el proyecto shapes

1. El proyecto “shapes” es una versión modificada de un recurso ofrecido por BlueJ. Para trabajar con él, bajen shapes.zip y ábralo en BlueJ. Captura la pantalla.



2. El diagrama de clases permite visualizar las clases de un artefacto software y las relaciones entre ellas. Considerando el diagrama de clases de “shapes”

(a) ¿Qué clases ofrece?

Canvas,Circle,Rectangle,Triangle

(b) ¿Qué relaciones existen entre ellas?

La clase canvas

3. La documentación2 presenta las clases del proyecto y, en este caso, la especificación de sus componentes públicos. De acuerdo con la documentación generada:

(a) ¿Qué clases tiene el paquete shapes?

Canvas,Circle,Rectangle,Triangle

(b) ¿Qué atributos ofrece la clase Rectangle?

EDGES

(c) ¿Cuántos métodos ofrece la clase Rectangle?

12 métodos

(d) ¿Cuáles métodos ofrece la clase Rectangle para que la figura cambie (incluya sólo el nombre)?

Para nosotros los criterios para cambiar una figura serían cambiar su color y tamaño y lo haríamos por medio de estos dos métodos: changeColor,changeSize.

4. En el código de cada clase está el detalle de la implementación. Revisen el código de la clase Rectangle. Con respecto a los atributos:

(a) ¿Cuántos atributos realmente tiene?

7 atributos

(b) ¿Cuáles atributos describen la forma del círculo?.

height,width,color

Con respecto a los métodos:

(c) ¿Cuántos métodos tiene en total?

14 métodos

(d) ¿Quiénes usan los métodos privados?

los métodos privados los usa la clase para colaborar con los métodos públicos.

5. Comparando la documentación con el código

(a) ¿Qué no se ve en la documentación?

los atributos y métodos privados

(b) ¿por qué debe ser así?

Es información irrelevante para ese manual técnico.

6. En el código de la clase Rectangle revisen el detalle del atributo EDGES.

(a) ¿Qué se está indicando al decir que es static?

Es un valor que ocupa un solo espacio en la memoria, es decir, al llamarlo, no se genera un nuevo valor, sino que simplemente se le hace un llamado al ya creado.

(b) ¿Cómo decimos que EDGES es una constante?

con la palabra reservada “final”

(c) Actualízalo.

```
public class Rectangle{

    public static final int EDGES = 4;

    private int height;
    private int width;
    private int xPosition;
    private int yPosition;
    private String color;
    private boolean isVisible;
```

7. En el código de la clase Rectangle revisen el detalle del tipo de los atributos height y width

(a) ¿Qué se está indicando al decir que es int?.

Es una variable que contendrá un valor numérico entero.

(b) Si sabemos que todos nuestros rectángulos van a ser muy pequeños (lados menor a 100), ¿de qué tipo deberían ser estos atributos?

byte: como su propio nombre denota, emplea un solo byte (8 bits) de almacenamiento. Esto permite almacenar valores en el rango [-128, 127].

(c) Si algunos de nuestros rectángulos pueden ser muy grandes (lados mayor a 220000000)¿de que tipo deberían ser estos atributos?

int: emplea 4 bytes de almacenamiento y es el tipo de dato entero más empleado. El rango de valores que puede representar va de -2^31 a 2^31-1.

(d) ¿qué restricción tendría? Expliquen claramente sus respuestas.

- Como los atributos Height y Width son Int, solo se deben ingresar valores de este tipo
- Para el byte manejar valores en un rango de [-128, 127].
- Para el Int manejar valores en un rango de -2^31 a 2^31-1.

8. ¿Cuál dirían es el propósito del proyecto “shapes”?

Experimentar con clases y objetos.

y para un usuario el propósito es:

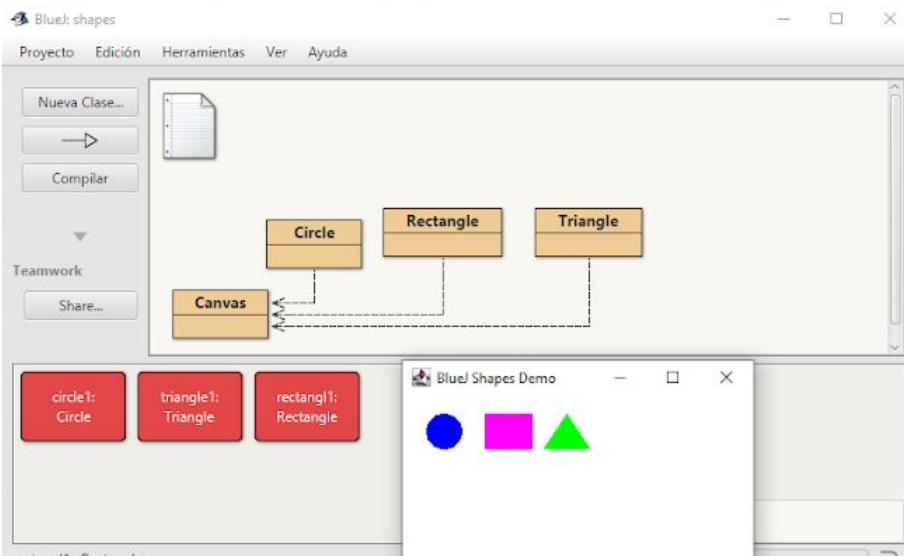
manipular tres figuras básicas, círculos, cuadrado, triangulo, cambiando atributos como color posición y tamaño

Manipulando objetos. Usando opciones.

[En lab01.doc] [TP 8]

1. Creen un objeto de cada una de las clases que lo permitan .

(a) ¿Cuántas clases hay? cuatro, tres clases lo permiten

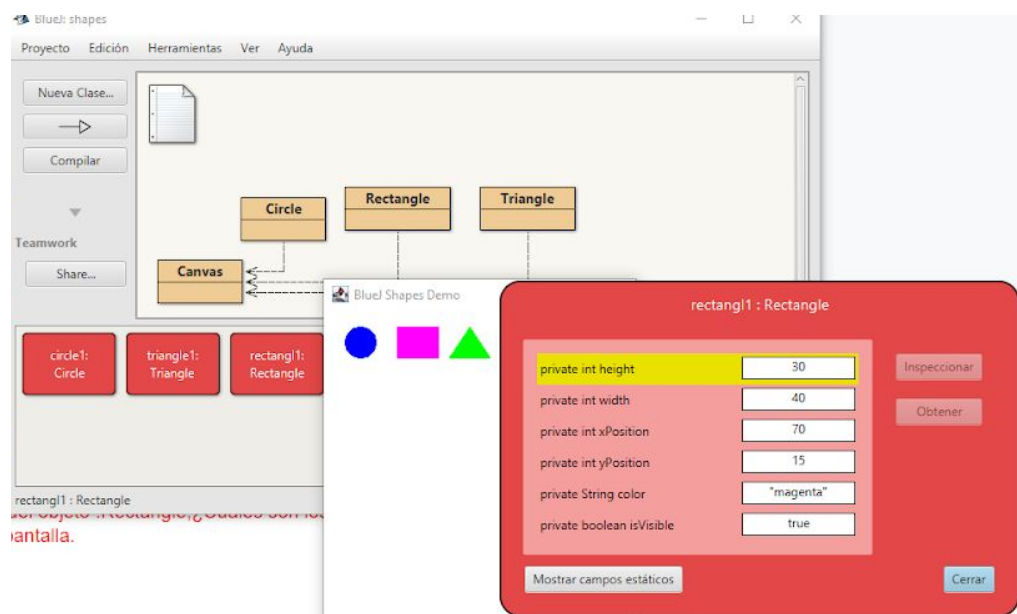


¿Cuántos objetos crearon? tres objetos

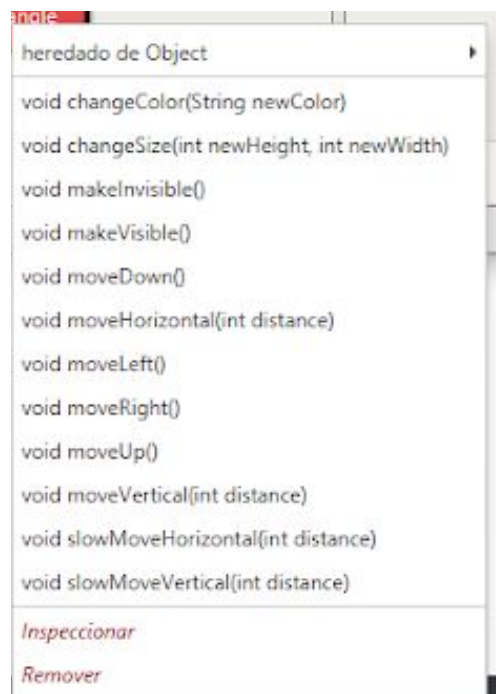
(b) ¿Por qué?

Porque queríamos visualizar cada objeto permitido y como el canvas funciona como una interfaz para lograr ver los tres objetos.

2. Inspeccionen el estado del objeto :Rectangle,¿Cuáles son los valores de inicio de todos sus atributos? Capture la pantalla.



3. Inspeccionen el comportamiento que ofrece el objeto :Rectangle. (a) Capture la pantalla. (b) ¿Por qué no aparecen todos los que están en el código?



debido a que algunos métodos son privados, es decir, son métodos que no son útiles para el usuario final.

4. Construyan, con “shapes” sin escribir código, una propuesta de la imagen del logo de su red social favorita.
(a) ¿Cuántas y cuáles clases se necesitan?

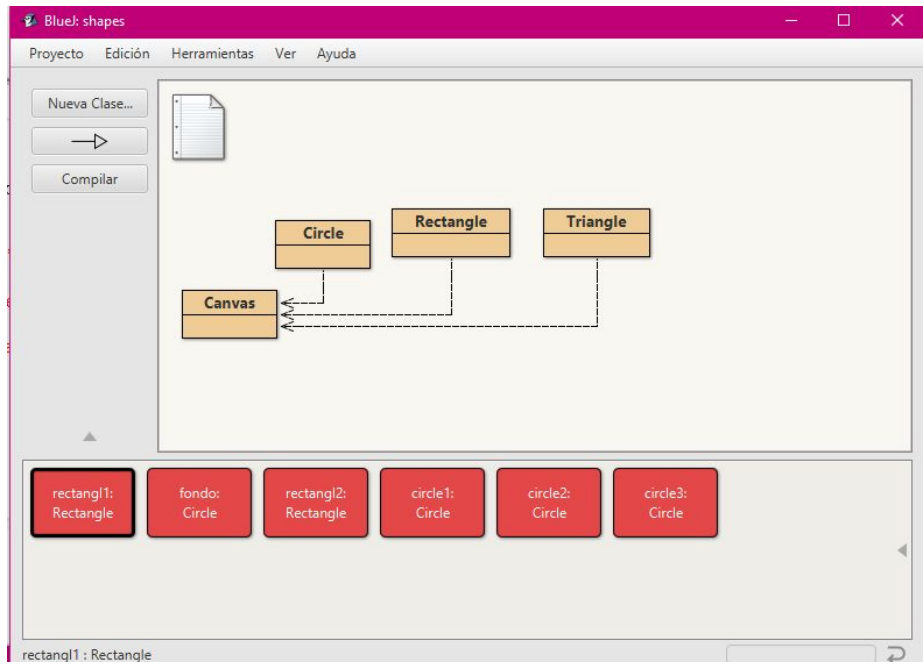
Tres clases, Circle, Rectangle y Canvas

(b) ¿Cuántos objetos se usan en total?

6 objetos

(c) Capturen la pantalla





Manipulando objetos. Analizando y escribiendo código.

```
Rectangle yellow;  
Rectangle blue;  
Rectangle red;  
//1  
yellow= new Rectangle();  
blue= new Rectangle();  
red= yellow;  
yellow.makeVisible();  
//2  
yellow.changeSize(30,80);  
yellow.changeColor("yellow");  
//3  
  
blue = new Rectangle();  
blue.changeSize(20,80);  
blue.changeColor("blue");  
blue.moveVertical(30);  
//4  
red.changeColor("red");  
red.changeSize(20,80);  
red.moveVertical(50);  
red.makeVisible();  
//5  
blue.makeVisible();  
//6
```

- 1. Lean el código anterior.
- (a) ¿cuál es la figura resultante?

Una bandera

- (b) Píntala.



- 2. Habiliten la ventana de código en línea , escriban el código. Para cada punto señalado indiquen:

- (a) ¿cuántas variables existen?

3 variables

- (b) ¿cuántos objetos existen?

2 objetos

- (c) ¿qué color tiene cada uno de ellos?

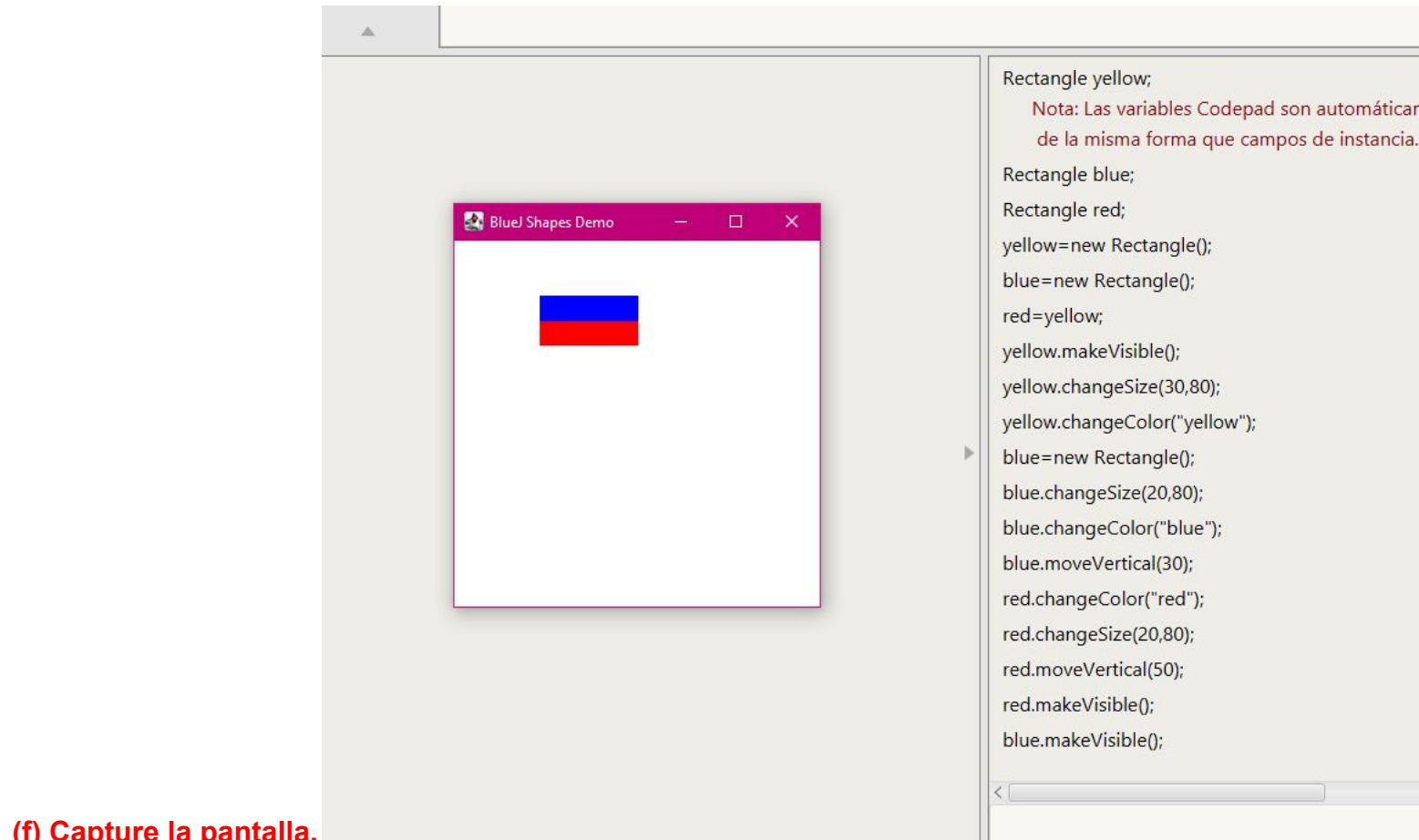
rojo y azul

- (d) ¿cuántos objetos se ven?

2 objetos

- (e) Expliquen sus respuestas.

al crear el objeto red y yellow en algunos momentos estas dos variables se igualan lo que en java genera un error lógico, se piensa que red heredará los cambios que se hicieron en yellow pero no, lo que realmente hace es que ahora red y yellow se guardarán en el mismo espacio de memoria haciendo los dos objetos uno solo.



(f) Capture la pantalla.

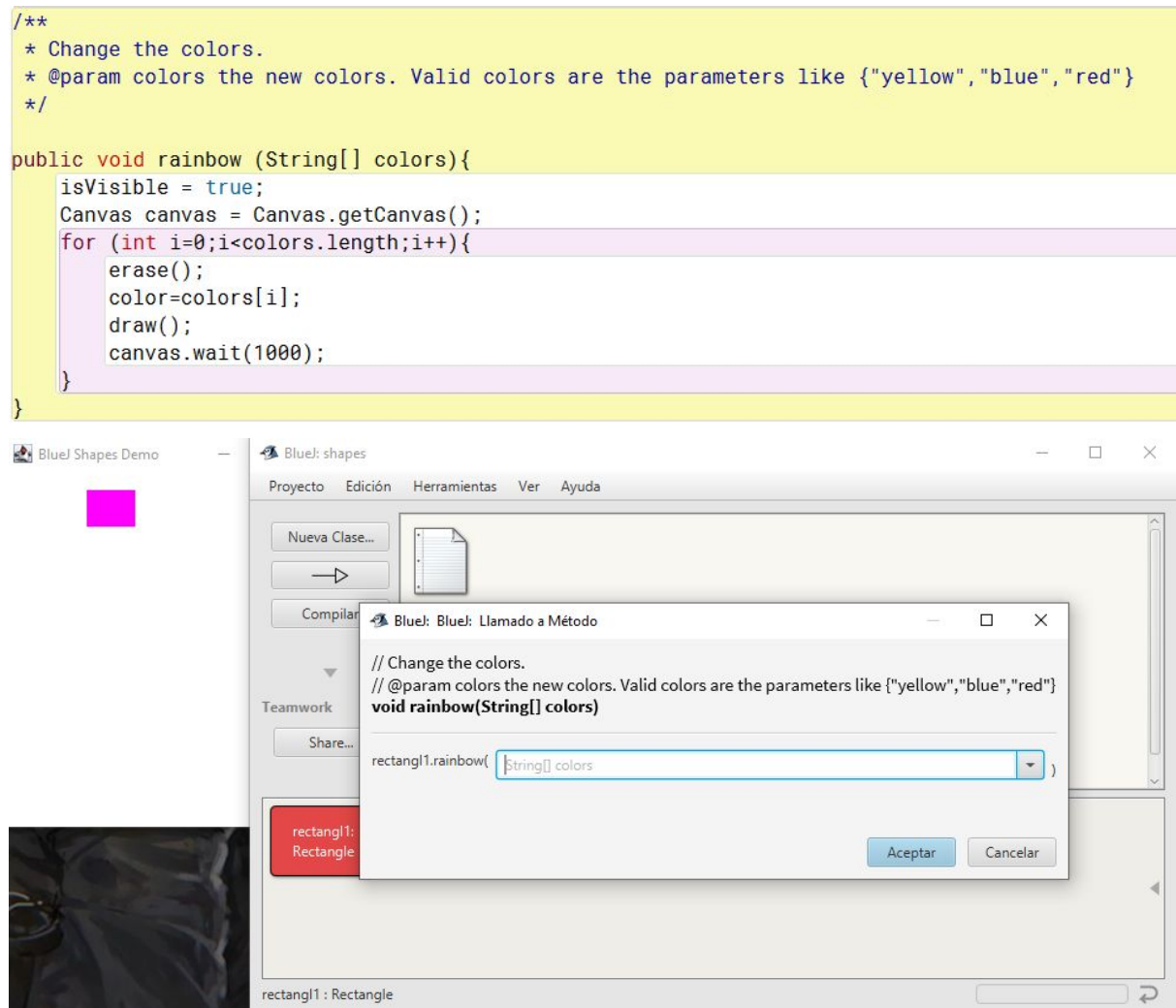
3. Compare figura pintada en 1. con la figura capturada en 2. , (a) ¿son iguales? (b) ¿por qué?



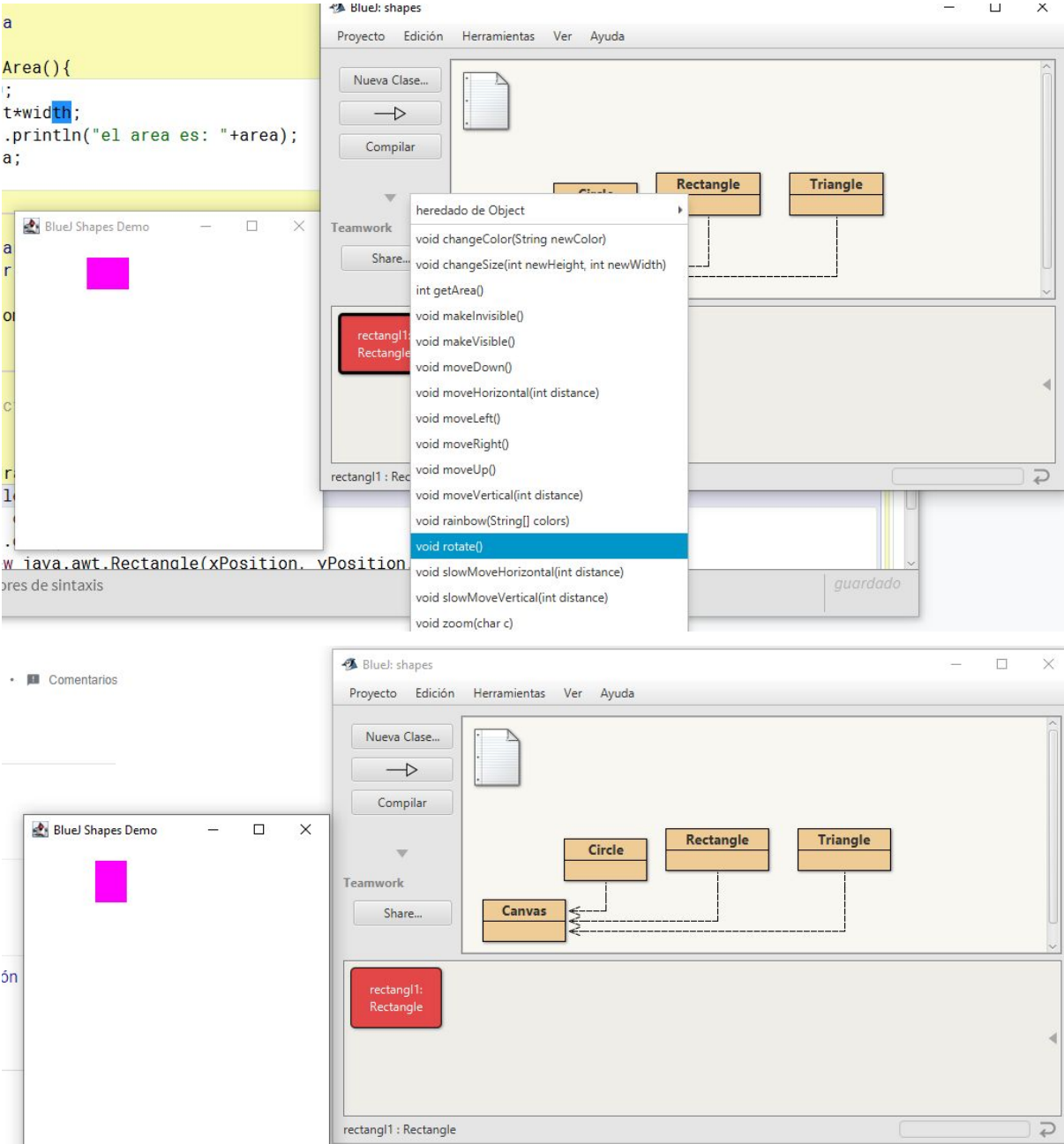
Si, son iguales ya que mi compañera y yo ya teníamos este concepto en mente.

Extendiendo clases

1. Desarrollen en Rectangle el método rainbow(String [] colors) (pasa por los colores indicados en al parametros) . ¡Pruébenlo! Capturan dos pantallas.

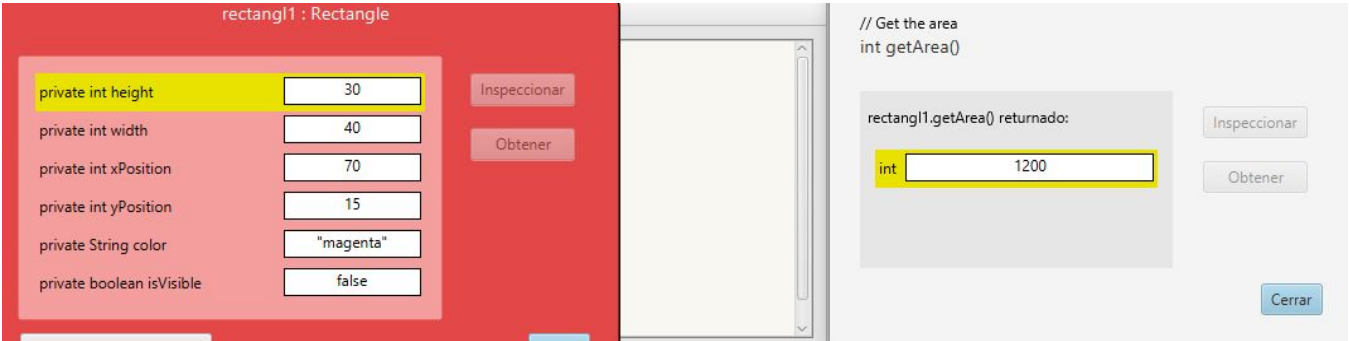


2. Desarrollen en Rectangle el método rotate() (rotar el rectángulo 90 grados en sentido del reloj) . ¡Pruébenlo! Capturan dos pantallas.



```
public void rotate(){
    int aux=0;
    aux=height;
    height =width;
    width =aux;
    draw();
}
```

3. Desarrollen en Rectangle el método área(). ¡Pruébenlo! Captura en una pantalla.



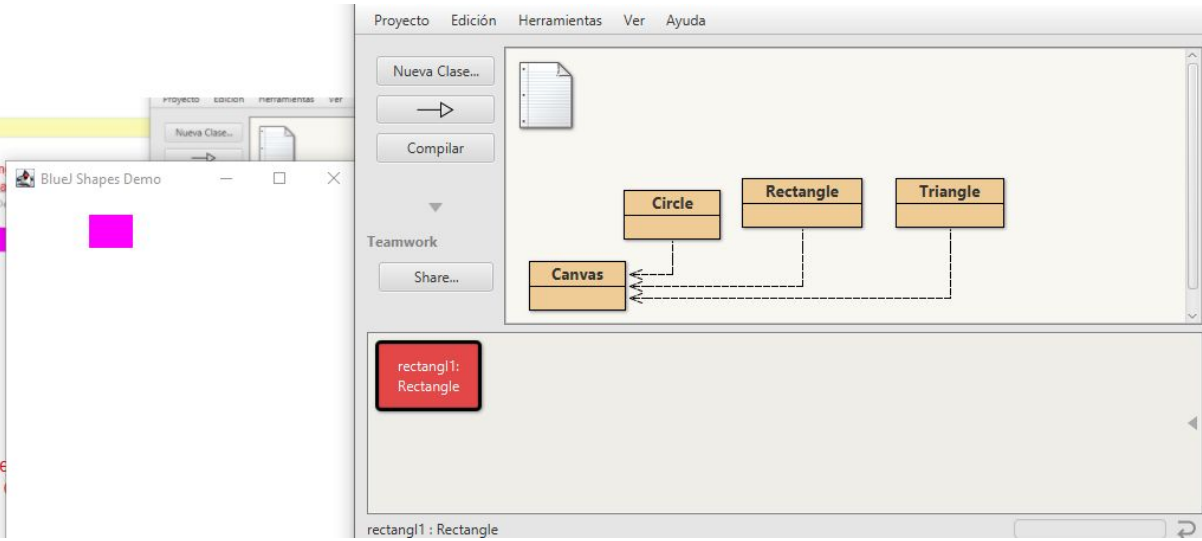
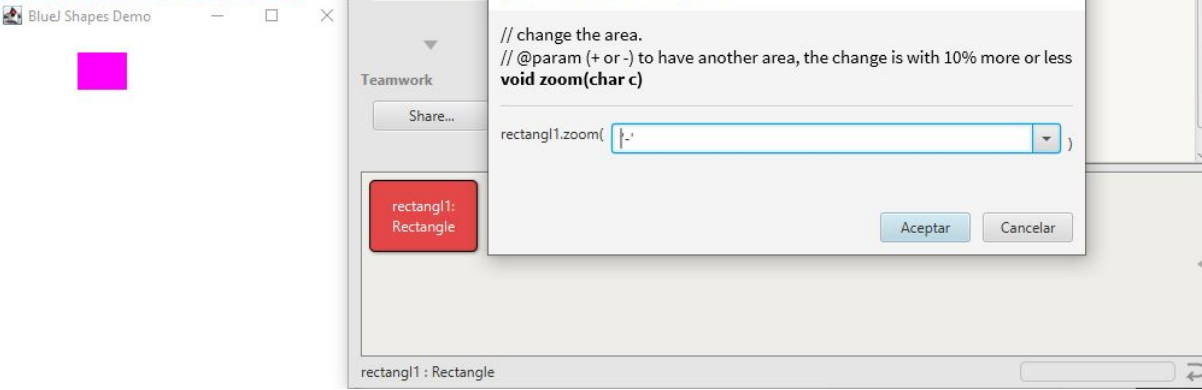
```
/**
 * Get the area
 */
public int getArea(){
    int area=0;
    area=height*width;
    System.out.println("el area es: "+area);
    return area;
}
```

4. Desarrollen en Rectangle el método zoom(char c) (aumenta (+) y disminuye (-) 10% de su área) . ¡Pruébenlo! Capturan dos pantallas.

```
public void zoom(char c){

    if (c=='+'){
        height = ((height*10)/100)+height;
        width = ((width*10)/100)+width;
    }
    else if (c=='-'){
        height =height-((height*10)/100);
        width = width-((width*10)/100);
    }
    else{
        System.out.println("Digite unicamente + o -");
    }
    draw();
}
```

llen en Rectangle el método zoom(char c) y pruébenlo! Capturan dos pantallas.



5. Generen nuevamente la documentación y revisen la información de estos nuevos métodos. Captura la pantalla.

All Methods	Instance Methods	Concrete Methods	
Modifier and Type		Method	Description
void		changeColor(String newColor)	Change the color.
void		changeSize(int newHeight, int newWidth)	Change the size to the new size
double		getArea()	Get the area
void		makeInvisible()	Make this rectangle invisible.
void		makeVisible()	Make this rectangle visible.
void		moveDown()	Move the rectangle a few pixels down.
void		moveHorizontal(int distance)	Move the rectangle horizontally.
void		moveLeft()	Move the rectangle a few pixels to the left.
void		moveRight()	Move the rectangle a few pixels to the right.
void		moveUp()	Move the rectangle a few pixels up.
void		moveVertical(int distance)	Move the rectangle vertically.
void		rainbow(String[] colors)	Change the colors.
void		rotate()	
void		slowMoveHorizontal(int distance)	Slowly move the rectangle horizontally.
void		slowMoveVertical(int distance)	Slowly move the rectangle vertically.
void		zoom(char c)	change the area.

rotate

```
public void rotate()
```

Rotate the figure turn of the clockwise

rainbow

```
public void rainbow(String[] colors)
```

Change the colors.

Parameters:

colors - the new colors. Valid colors are the parameters like {"yellow","blue","red"}

getArea

```
public double getArea()
```

Get the area

zoom


```
public void zoom(char c)
```

change the area.

DICESE. Dice.

<p>DICESE es un juego de dados solitario. En este punto vamos a construir dos versiones de DICESE</p> <p>La primera versión es simple: sólo tiene una fila. Para ganar se deben tener todos los elementos ordendados.</p> <p>La segunda versión es cuadrado. Para ganar se deben tener ordenados todos los elementos de una fila, una columna o una diagonal.</p> <p>Las dimensiones debe estar entre 2 y 6.</p>	
--	--

Implementando una nueva clase. Dice.

	<div><div>Dice</div><div><pre>+ Dice() + getValue() : int + roll() : void + changeColor(color : String) : void + moveHorizontal(distance : int) : void + makeVisible() : void + makeInvisible() : void</pre></div></div>	<p>MINICICLOS</p> <ol style="list-style-type: none">DicemakeVisible makeInvisiblemoveHorizontalroll getValue
--	--	--

1. Revisen el diseño y clasifiquen los métodos en: constructores, analizadores y modificadores.

Constructores:

Dice()

Analizadores:

getValue()

Modificadores:

roll()

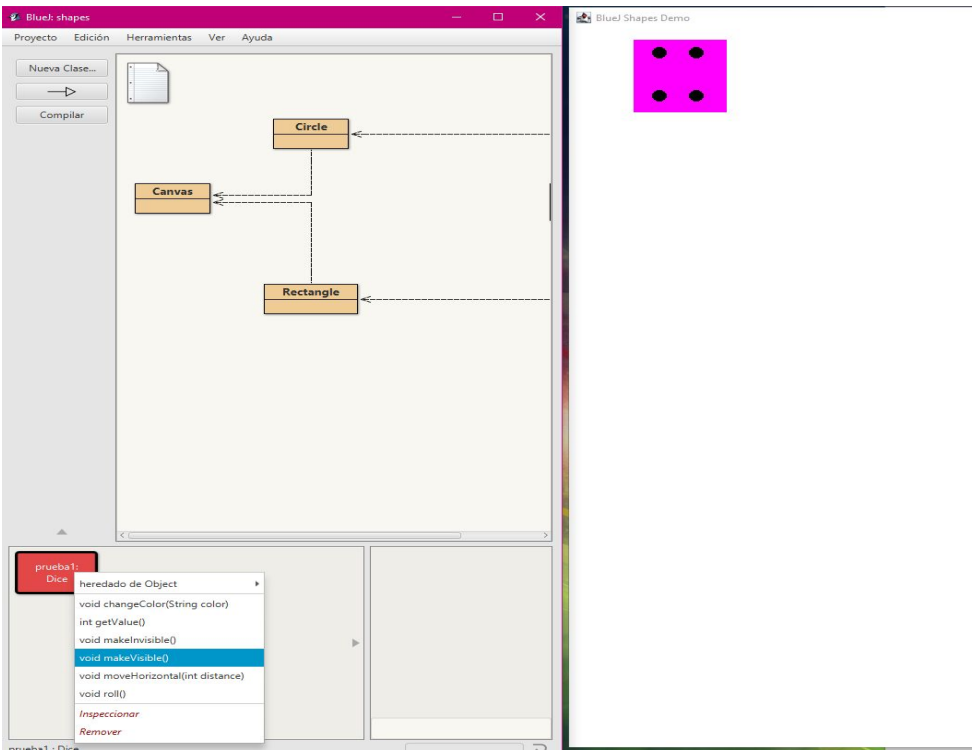
changeColor(color:String)

moveHorizontal(distance:int)

makeVisible()

MakeInvisible()

2. Desarrollen la clase Dice considerando los 4 mini-ciclos. Al final de cada mini-ciclo realicen una prueba indicando su propósito. Capturen las pantallas relevantes.



El propósito de esta prueba es evidenciar el funcionamiento del constructor y el método makeVisible

Dice (shapes)

← → ↻ 🏠 ⓘ Archivo | C:/Users/57350/Desktop/versionCool/shapes/doc/Dice.html

PACKAGE CLASS TREE INDEX HELP

ALL CLASSES

SUMMARY NESTED FIELD CONSTR METHOD DETAIL FIELD CONSTR METHOD

Class Dice

java.lang.Object
Dice

public class Dice
extends Object

DICE es una clase que simula el uso de un dado.

Author:
(Diego Macana-Natalia Orjuela)

Constructor Summary

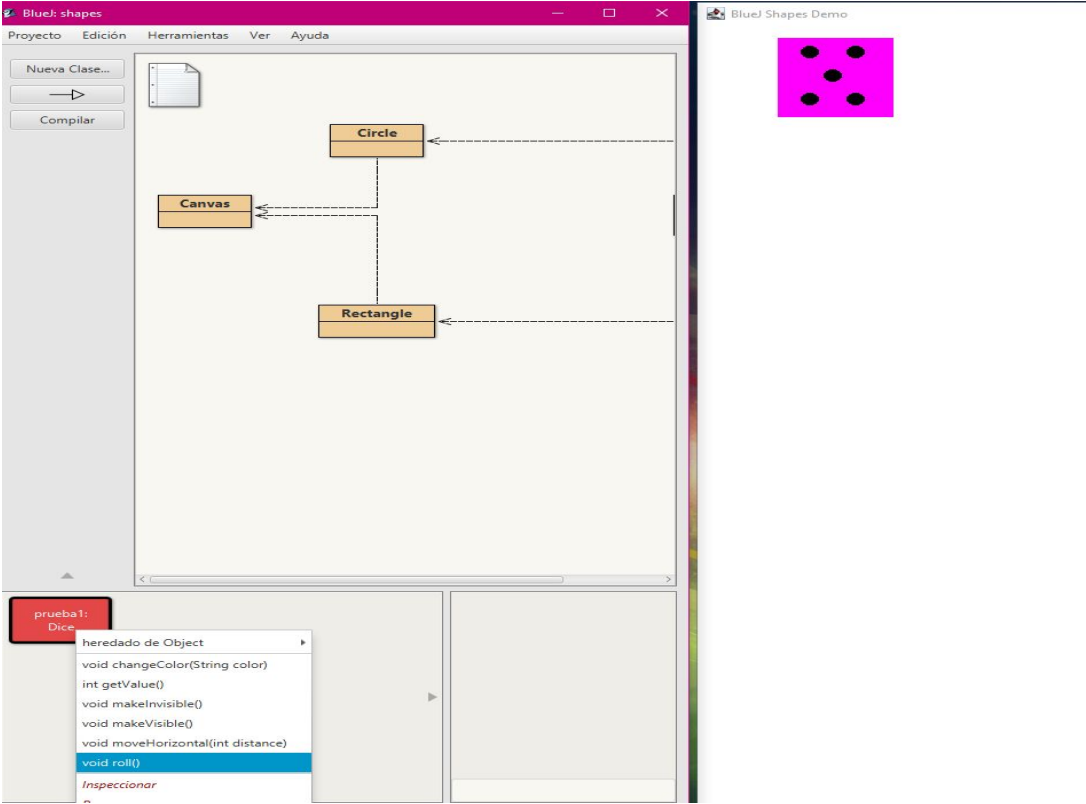
Constructors

Constructor	Description
Dice()	Constructor of dice to play

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	changeColor(String color)	Método que permite cambiar el color del dado
int	getValue()	Este método devuelve el valor del dado
void	makeInvisible()	Este método oculta el dado
void	makeVisible()	Este método muestra el dado
void	moveHorizontal(int distance)	Método que permite trasladar el dado horizontalmente
void	roll()	Este método lanza aleatoriamente el dado



El propósito de esta prueba es evidenciar el uso del método roll

Implementando una nueva clase. Dicese.

1. ¿De qué tipo son los elementos en Dicese?

De tipo Dice

¿Cuántos posibles estados puede tener?

Dos estados, true o false ya que retorna un valor booleano.

2. ¿Cuál es la probabilidad de ganar? Explique su respuesta.

$$G(k)=2 (7 - k) / 6^k$$

Donde k es la cantidad de dados a lanzar.
Analizamos los casos posibles ascendente y descendente y observamos que teníamos 7-k posibilidades de ganar tanto ascendente como descendente sobre las posibles combinaciones que pueden surgir de k dados.

3. Clasifiquen los métodos en: constructores, analizadores y modificadores.

Constructores:

Dicese(n)

Analizadores:

isWinningState()

percentageofWinningStates()

Modificadores:

reset()

play()

play(times)

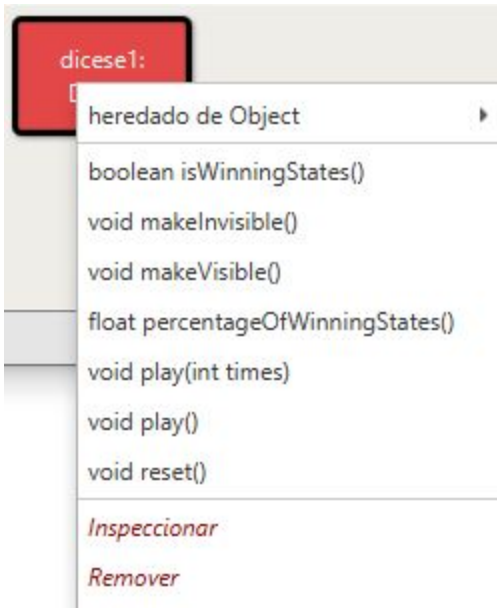
makeVisible()

makeInvisible()

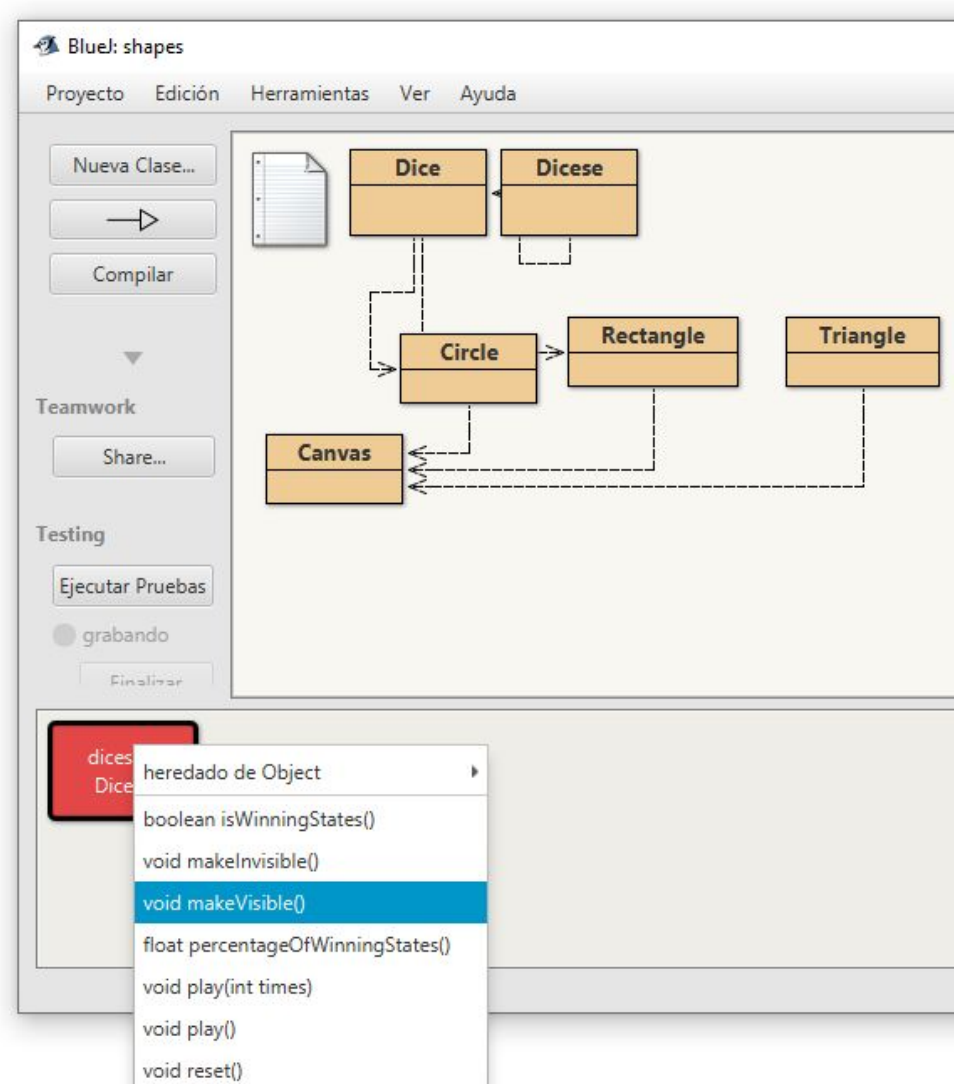
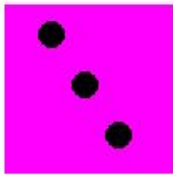
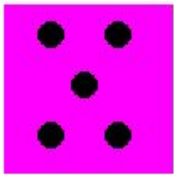
move(horizontal,vertical)

4. Desarrollen la clase Dicese considerando los miniciclos. Al final de cada miniciclo realicen una prueba. Capturen las pantallas relevantes.

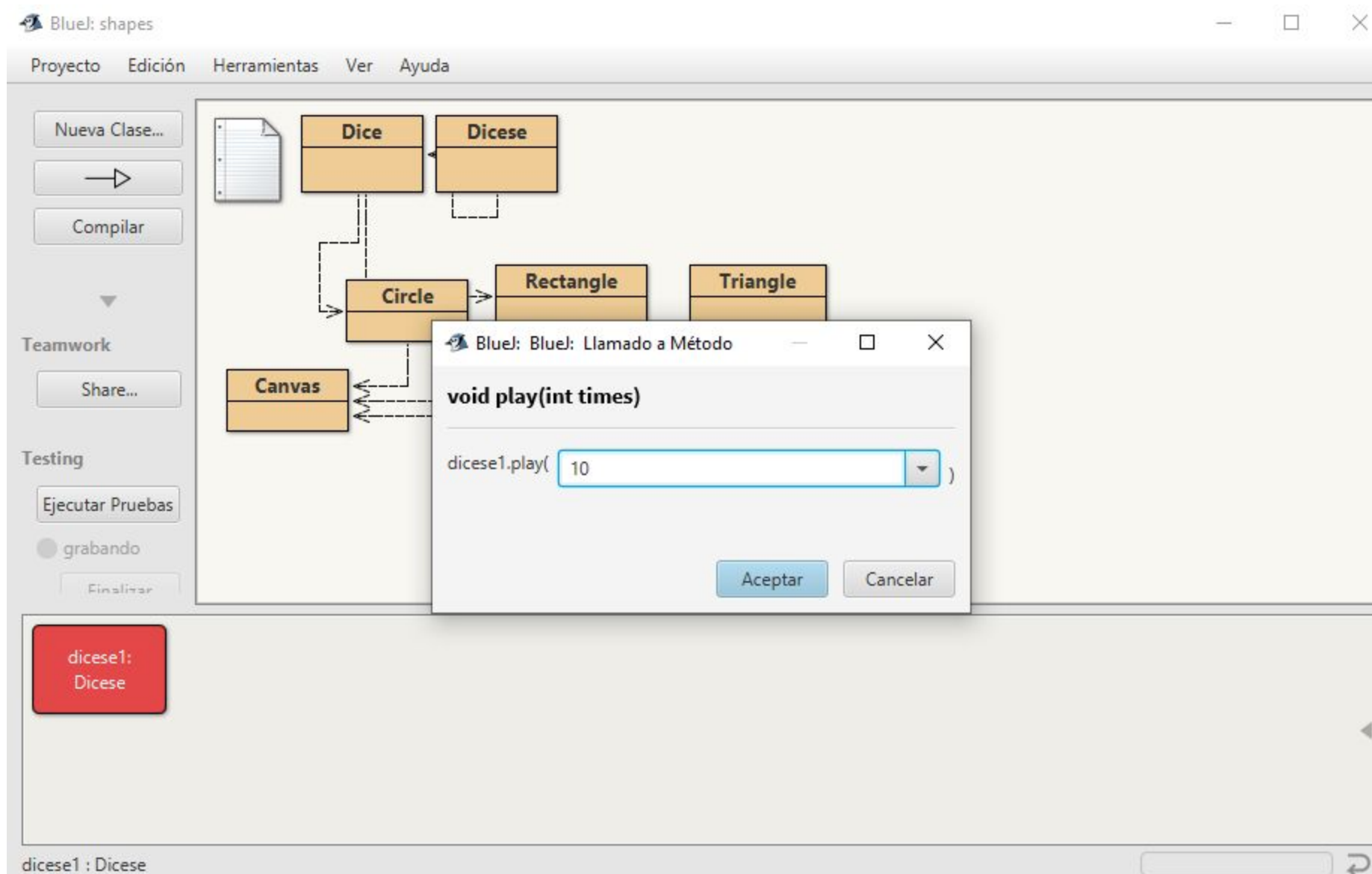
metodos:

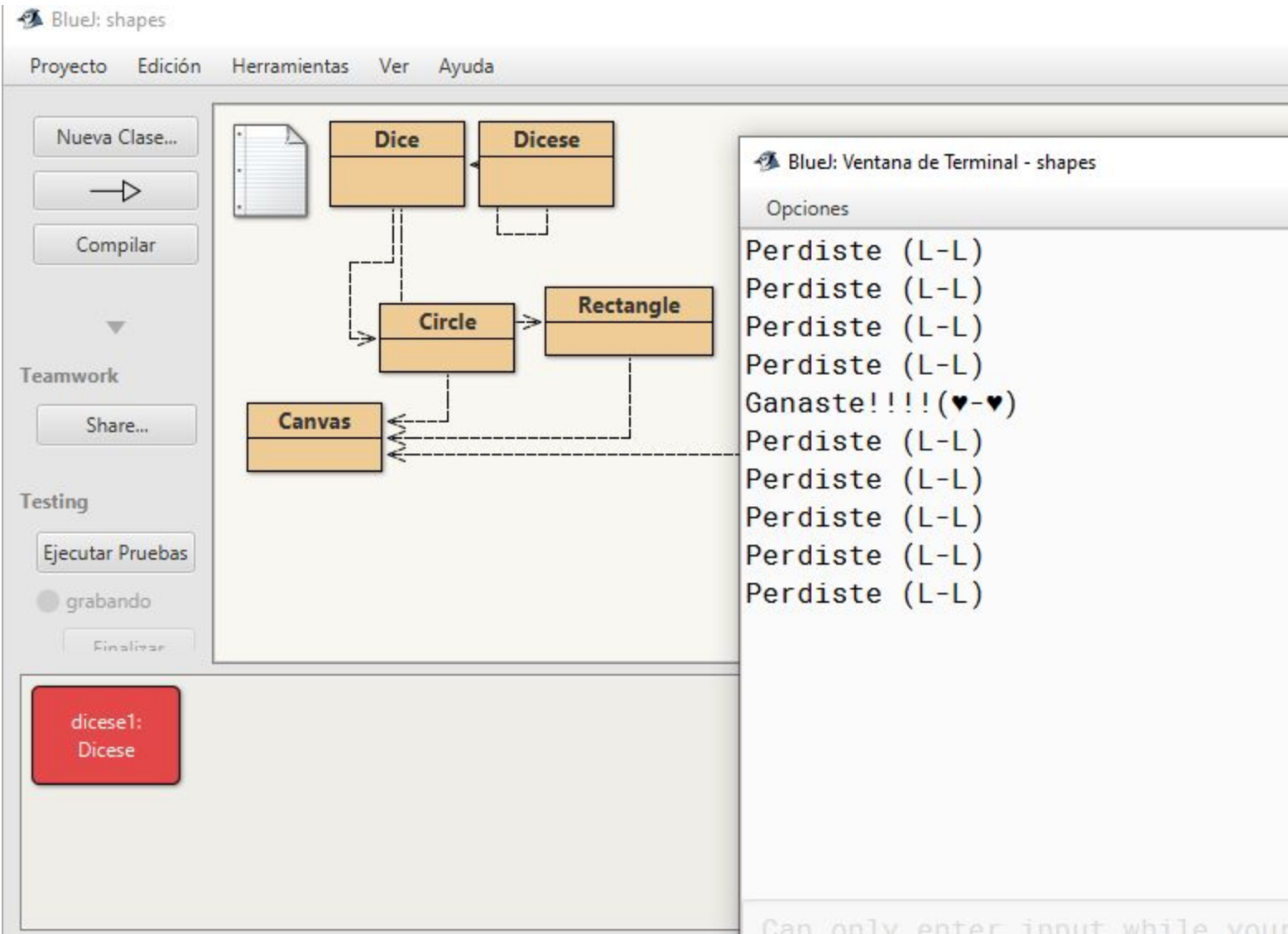


Makevisible()

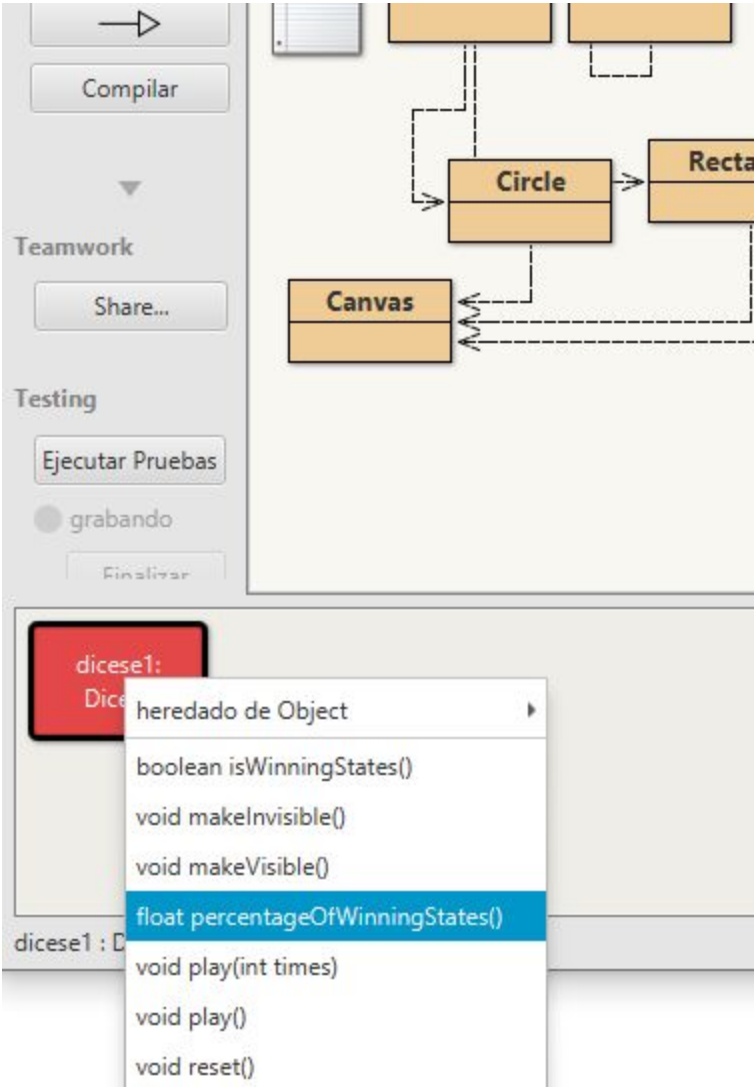


play(times):

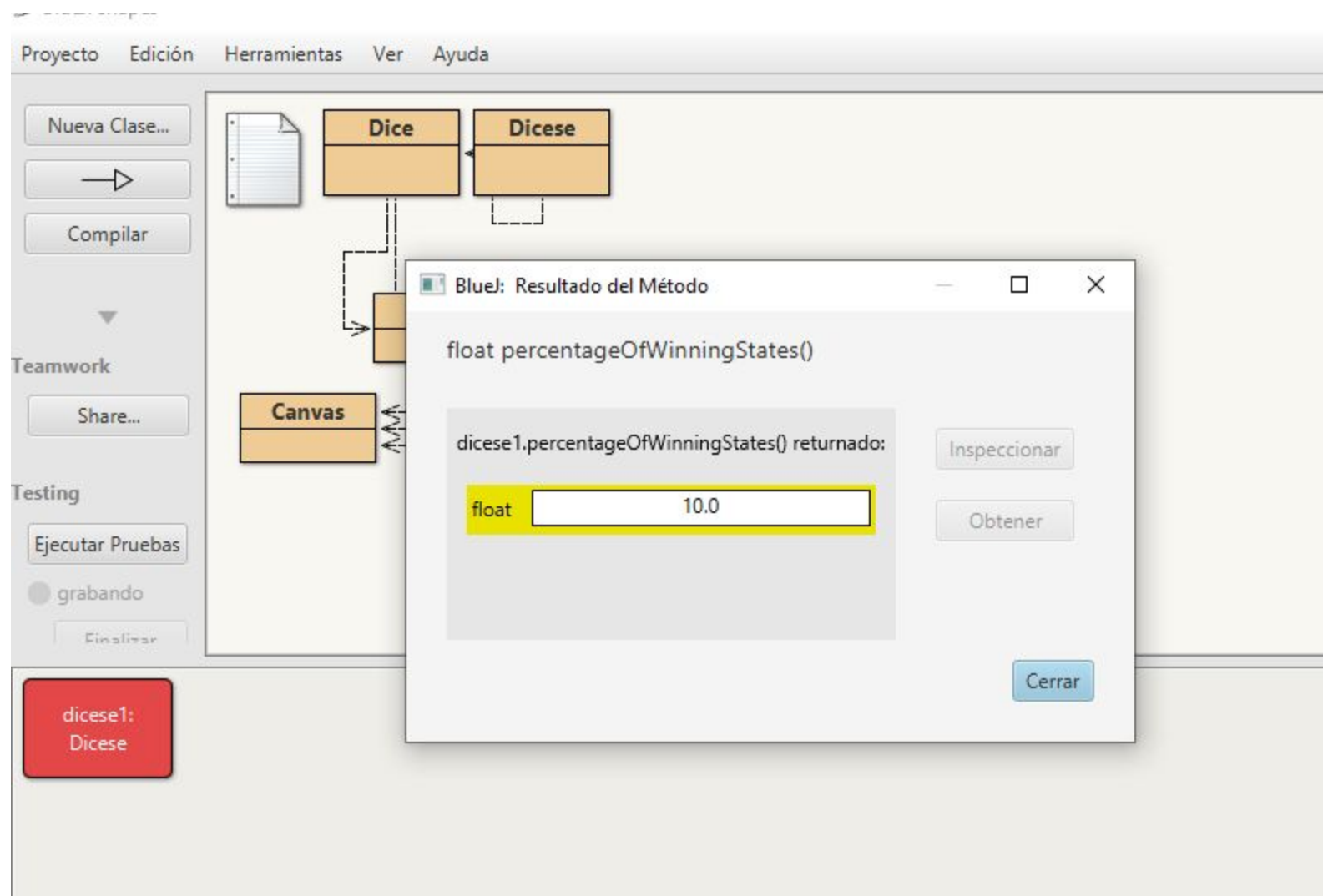




percentage of wins:

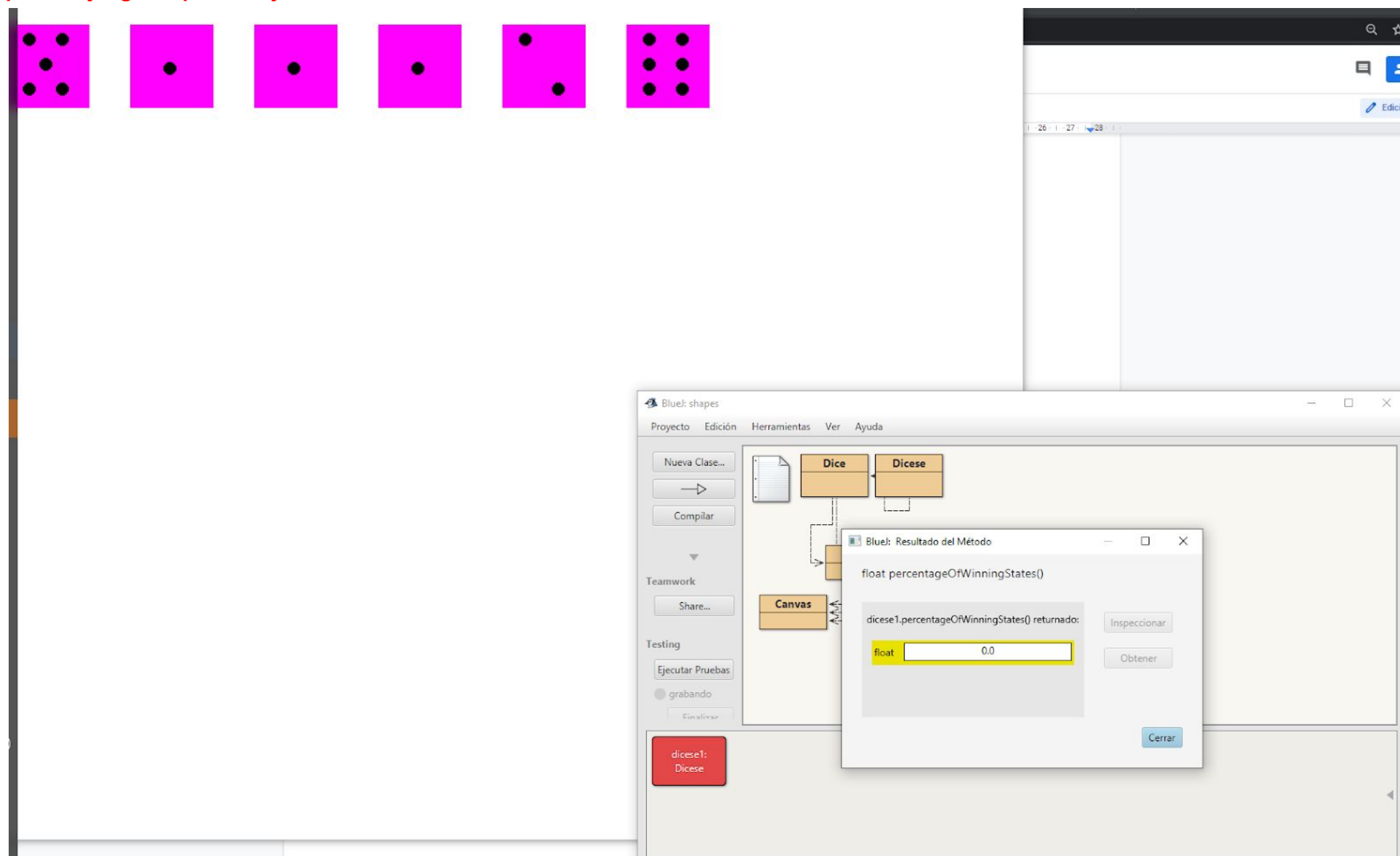


porcentaje de victorias retorna que es 10% ya que se ganó un juego de 10

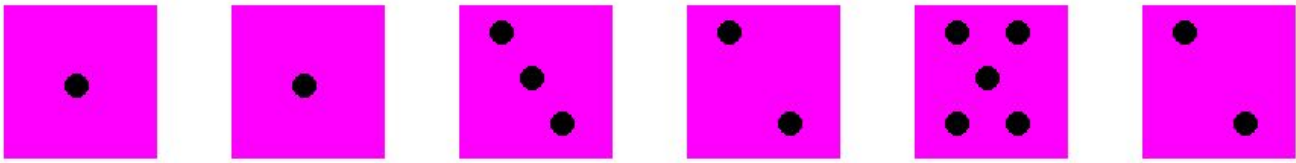


5. ¿Cual es el porcentaje de estados ganadores después de hacer 1, 10, 100 y 1000 jugadas? Presente un análisis de los datos considerando la respuesta dada en 2.

para un juego su porcentaje de victoria es 0%



para 10 juegos su porcentaje de victorias es 0%



Blue!: shapes

Proyecto Edición Herramientas Ver Ayuda

Nueva Clase...

→

Dice

Dicese

ii

|

|

Blue!: Resultado del Método

float percentageOfWinningStates()

dicese1.percentageOfWinningStates() retornado:

float 0.0

Inspeccionar

Obtener

Cerrar

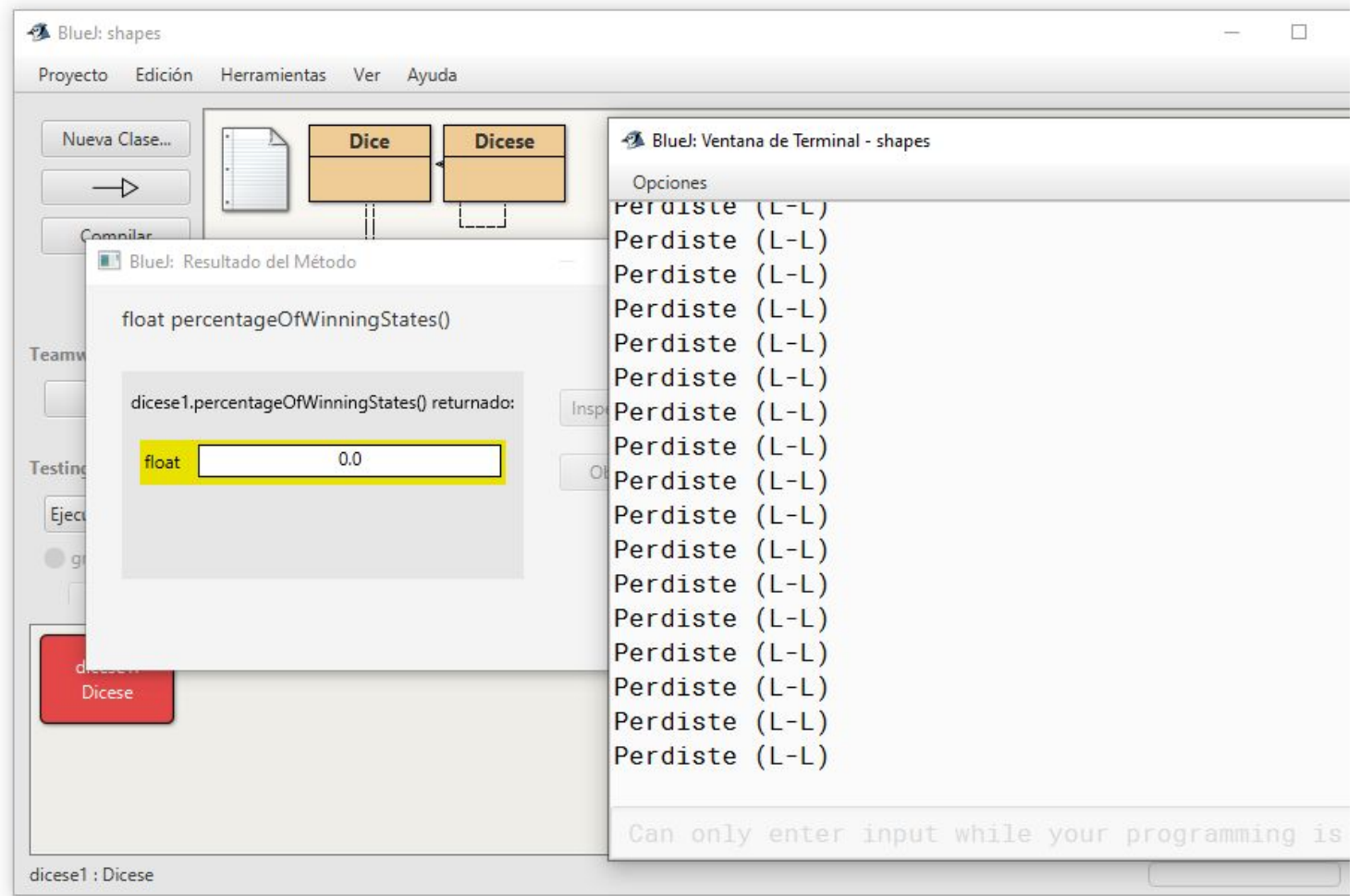
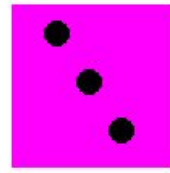
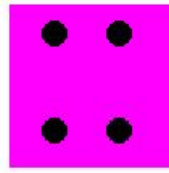
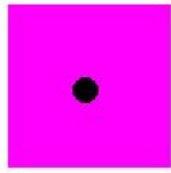
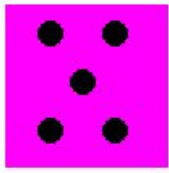
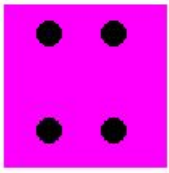
Blue!: Ventana de Terminal - shapes

Opciones

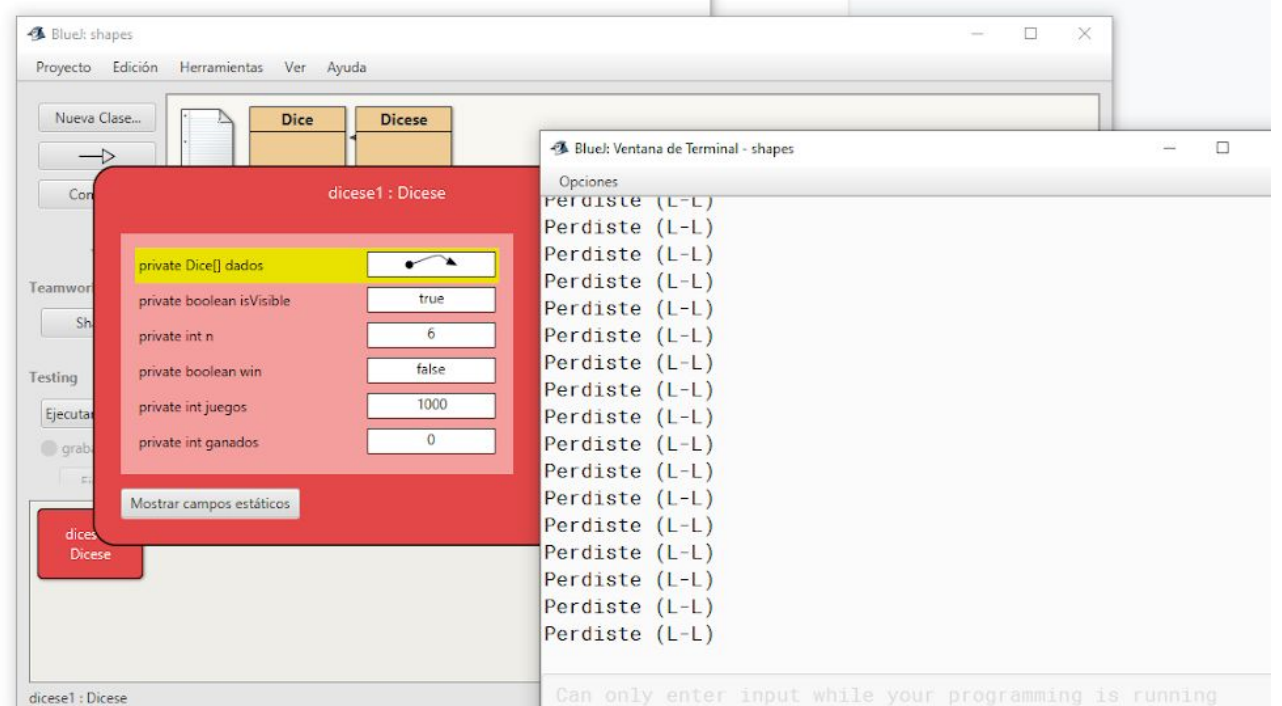
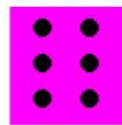
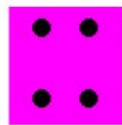
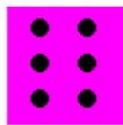
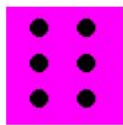
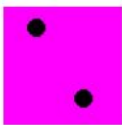
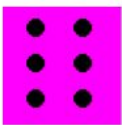
Perdiste (L-L)
Perdiste (L-L)
Perdiste (L-L)
Perdiste (L-L)
Perdiste (L-L)
Perdiste (L-L)
Perdiste (L-L)
Perdiste (L-L)
Perdiste (L-L)
Perdiste (L-L)

Can only enter input while your programming is running

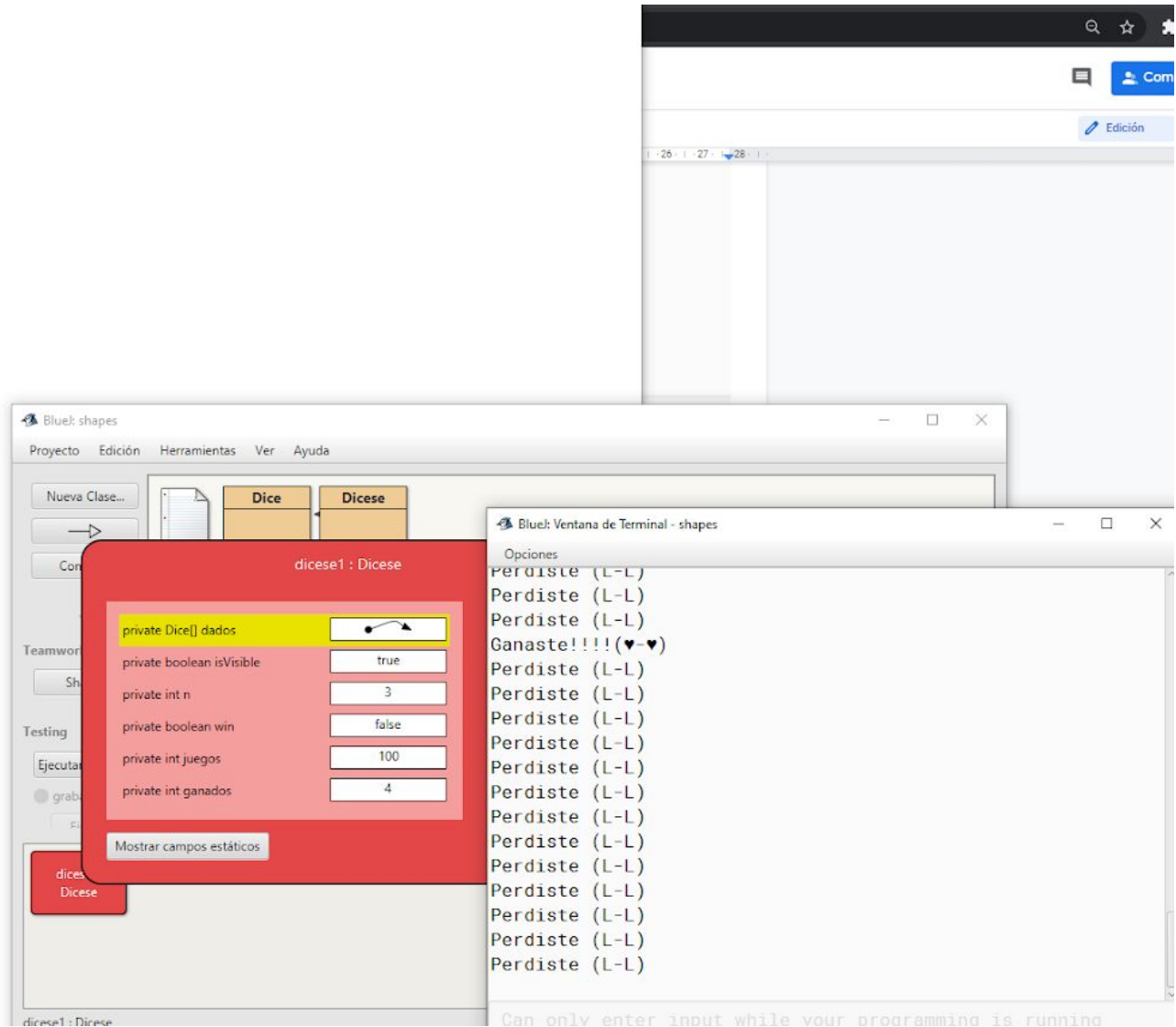
para 100 juegos su porcentaje de victorias es 0%



para 1000 juegos su porcentaje de victorias es 0%



aca muestro la ejecución de 100 veces, donde solamente 4 veces se gano



para concluir cuando se jugó con 3 dados el porcentaje de victoria es de solo el 3,7%
cuando se jugó con 6 dados el porcentaje de victoria es de 0.000000000004% incluso en 1000 lances se obtuvo 0 victorias
cuando jugamos con 3 dados nuestro porcentaje de victoria fue 4% donde concluimos que la fórmula funciona correctamente para k dados

Definiendo y creando una nueva clase. BiDicese
El objetivo es implementar, diseñar e implementar un tragamonedas cuadrado. Requisitos funcionales
3. Permitir crear un BiDicese indicando el tamaño.

```
public BiDicese(int n)
{
    // initialise instance variables
    if (n<2 || n>6){
        JOptionPane.showMessageDialog(null, "Por favor ingresar un valor entre 2 y 6.");
        System.exit(0);
    }
    this.n=n;
    juegos=new Dicese[n];
    juegosJugados=0;
    porcenSave=0;
    for (int i=0;i<n;i++)
    {
        juegos[i]=new Dicese(n);
        for (int j=0;j<i;j++)
        {
            juegos[i].moveVertical();
        }
        juegos[i].makeVisible();
    }
}
```

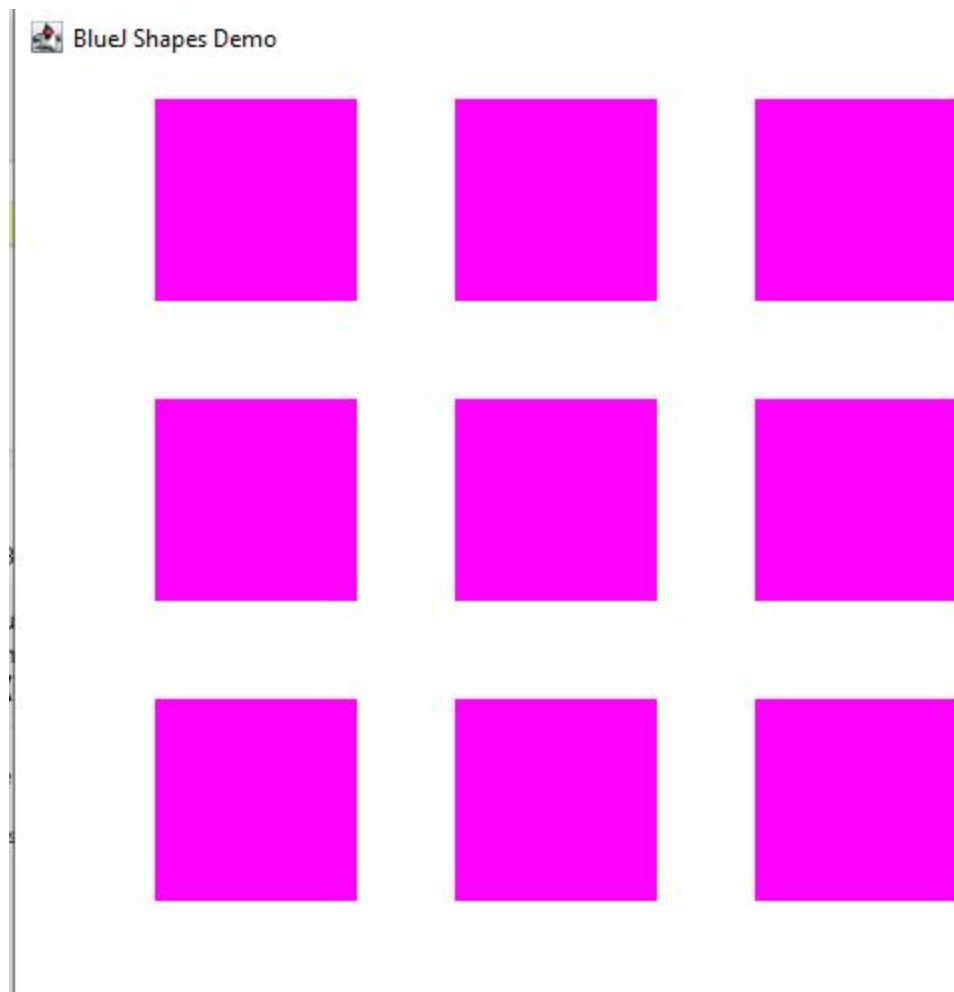

BlueJ: BlueJ: Crear Objeto

// Constructor for objects of class BiDicese
// @param int n cantidad de dados
BiDicese(int n)

Nombre de Instancia:

new BiDicese()

Aceptar Cancelar

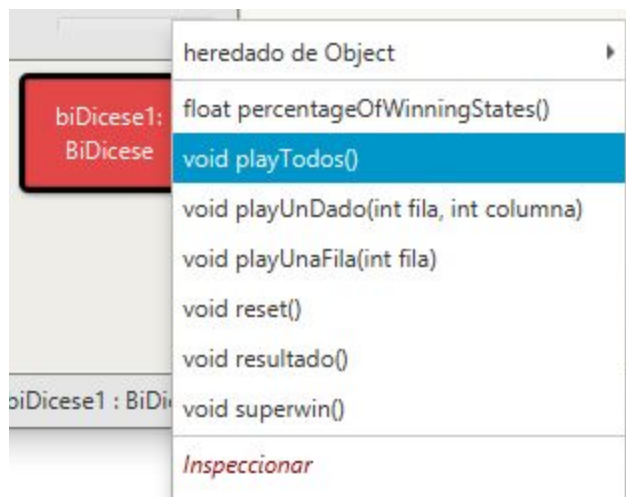


4. Permitir reiniciar

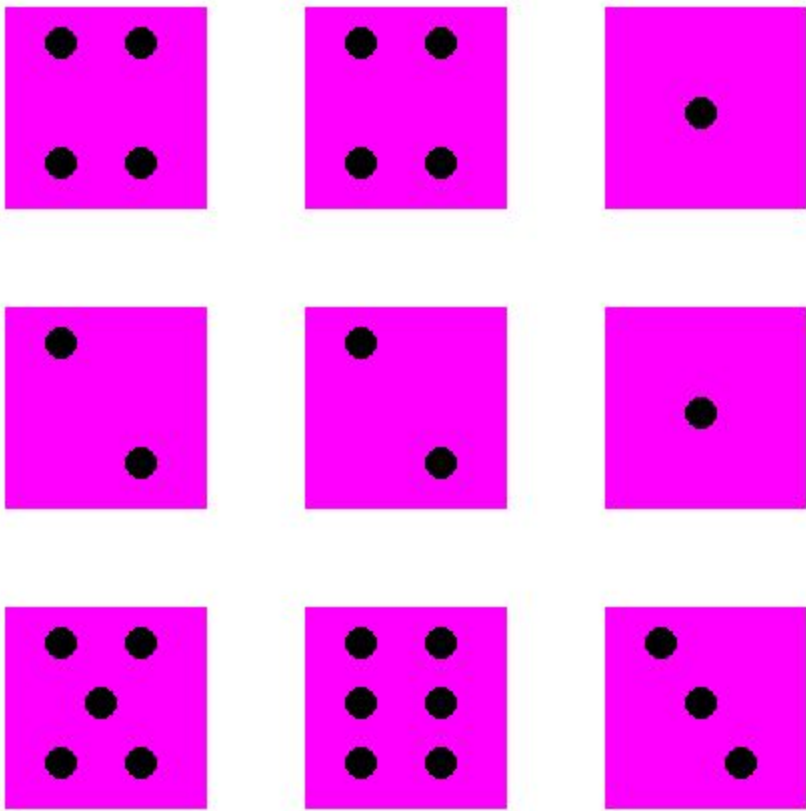
```
/**
 * metodo para reiniciar un juego
 */
public void reset()
{
    juegosJugados=0;
    ganados=0;
    for (int i=0;i<n;i++)
    {
        juegos[i].reset();
    }
}
```

5. Permitir jugar (todas filas giran)

```
/**
 * metodo para jugar con todas las filas de dados
 * juegos[] es un array de filas de dados
 */
public void playTodos(){
    for (int i=0;i<n;i++)
    {
        juegos[i].play();
    }
    juegosJugados+=1;
}
```

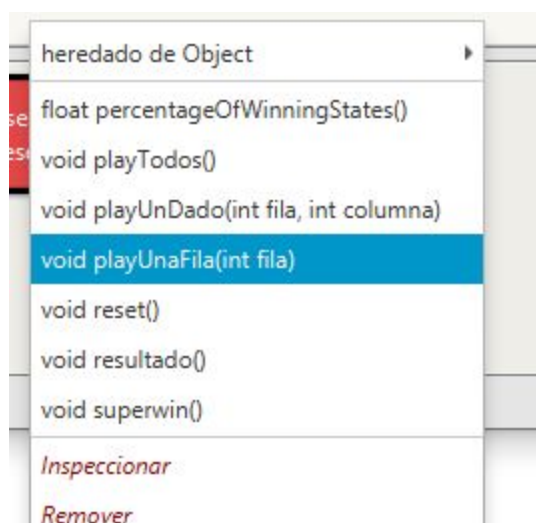


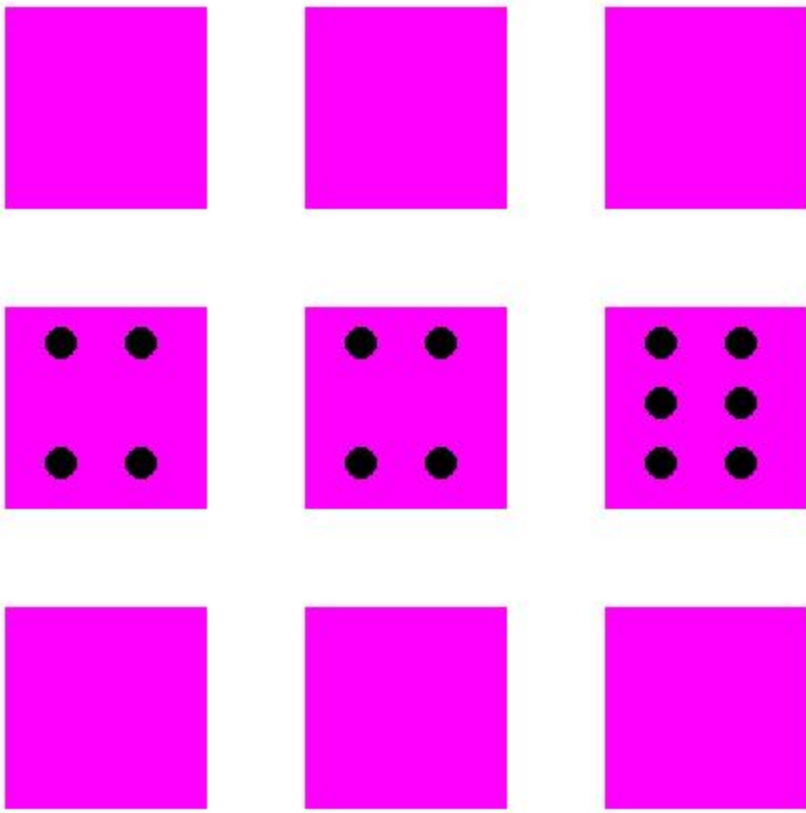
BlueJ Shapes Demo



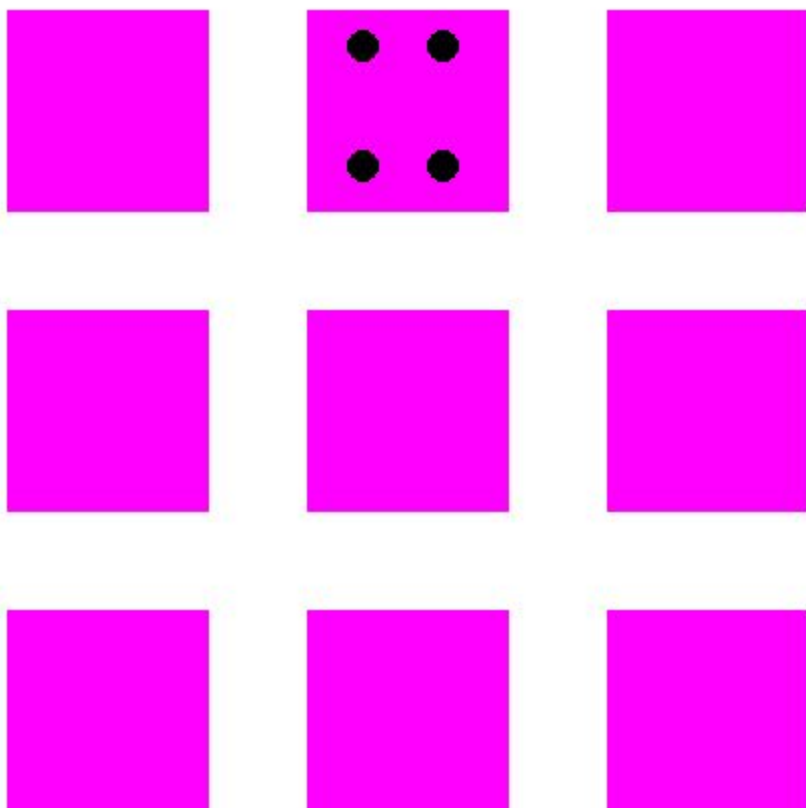
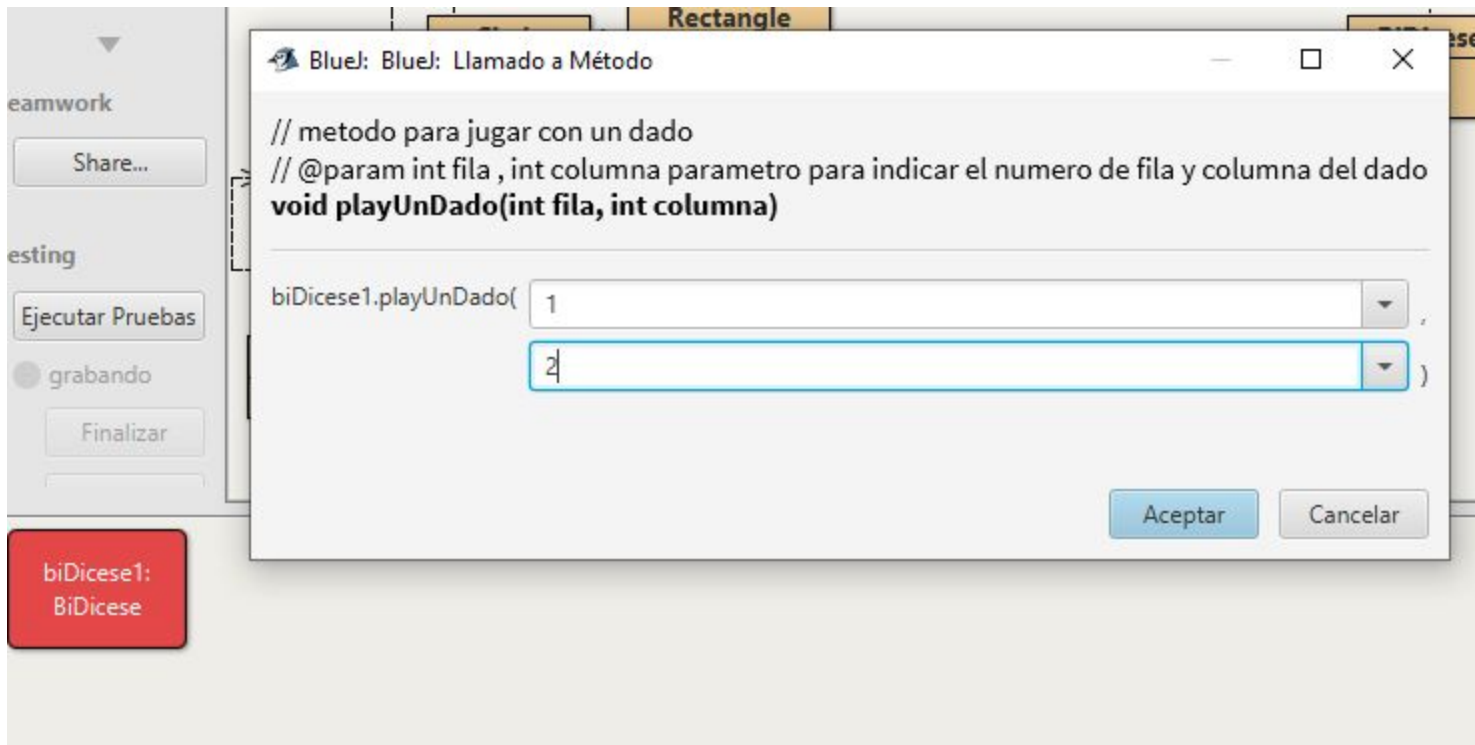
6. Permitir jugar indicando la fila con el que se desea jugar (1 ...n de derecha a izquierda)

```
/**
 * metodo para jugar con una fila de dados
 * @param int fila parametro para indicar el numero de fila
 */
public void playUnaFila(int fila){
    juegos[fila-1].play();
    juegosJugados+=1;
}
```





7. Permitir jugar indicando el dado con el que se desea jugar (fila, columna)



```

/**
 * metodo para jugar con un dado
 * @param int fila , int columna parametro para indicar el numero de fila y columna del dado
 */
public void playUnDado(int fila, int columna){
    juegos[fila-1].playUnDado(columna);
    juegosJugados+=1;
}

```

8. Permitir consultar si se ha ganado

```

/**
 * metodo para saber si se gano o no.
 *
 */
public void resultado(){

    int conta=juegos[0].getDado(0);
    int ganoAD=1,ganoDD=1,ganoAC=1,ganoDC=1,ganoFIL=0,ganoCOL=0;
    for (int i=1;i<n;i++) //filas
    {
        for (int j=1;j<n;j++) //columnas
        {
            if (i==j)
            {
                // System.out.println("i"+i+"j"+j);
                //System.out.println("dado a comparar :"+conta+"contra: "+juegos[i].getDado(j));
                if (conta+1==juegos[i].getDado(j))
                {
                    ganoAD+=1;
                    //System.out.println("+");
                    conta=juegos[i].getDado(j);
                }
                else if (conta-1==juegos[i].getDado(j))
                {
                    ganoDD+=1;
                    //System.out.println("-");
                    conta=juegos[i].getDado(j);
                }
                else{ j=n;}
            }
        }
    }
}

```



```

int j=0;
for (int i=0;i<n;i++) //filas
{
    conta=juegos[0].getDado(i);
    //System.out.println("pivote nuevo: "+conta);
    //System.out.println("AC"+ganoAC);
    //System.out.println("DC"+ganoDC);
    if (juegos[i].resultadoFIL()){
        ganoFIL=n;
    }
    for (int k=1;k<n;k++) //columnas
    {
        //System.out.println("comparamos "+conta+"contra"+juegos[k].getDado(i));
        if (conta+1==juegos[k].getDado(i))
        {
            ganoAC+=1;
            //System.out.println("+");
            conta=juegos[k].getDado(i);
        }
        else if (conta-1==juegos[k].getDado(i))
        {
            ganoDC+=1;
            //System.out.println("-");
            conta=juegos[k].getDado(i);
        }
    }
}

```

```

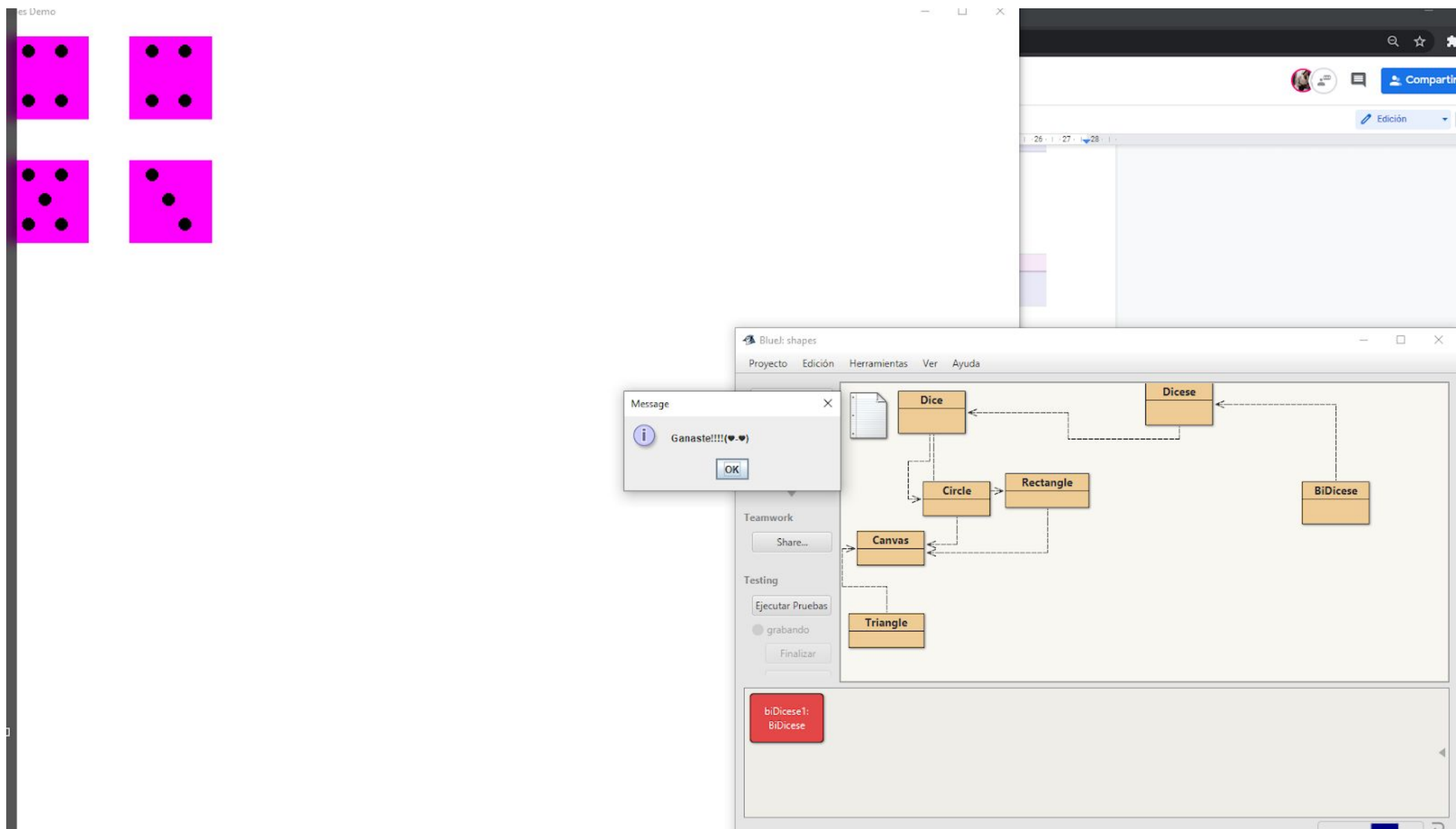
    if (ganoAC==n || ganoDC==n){
        ganoCOL=n;
    }

    //System.out.println("AC antes"+ganoAC);
    //System.out.println("DC antes"+ganoDC);

    ganoAC=1;
    ganoDC=1;
    j+=1;
}

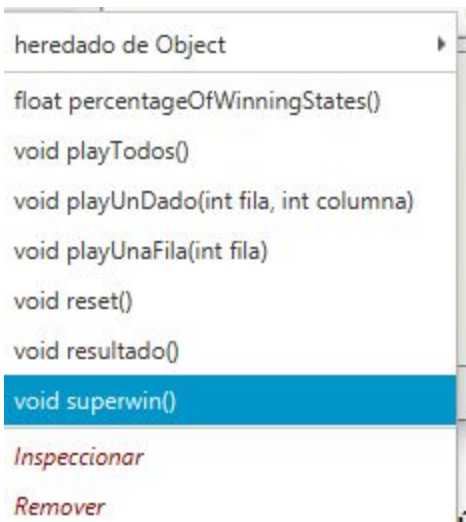
if (ganoAD==n || ganoDD==n || ganoCOL==n || ganoFIL==n )
{
    //System.out.println("Ganaste!!!!(♥-♥)");
    JOptionPane.showMessageDialog(null, "Ganaste!!!!(♥-♥)");
    win=true;
    ganados+=1;
}
else
{
    JOptionPane.showMessageDialog(null, "Perdiste (L-L)");
    //System.out.println("Perdiste (L-L)");
    win=false;
}

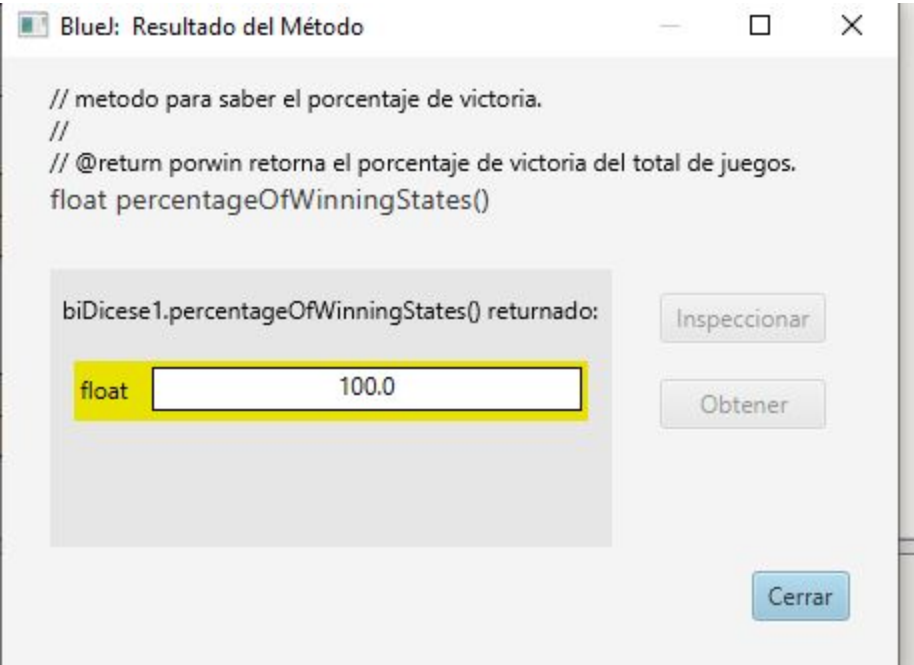
```



9. Permitir consultar el porcentaje de juegos ganadores desde el último reinicio

```
/**
 * metodo para saber el porcentaje de victoria.
 *
 * @return porwin retorna el porcentaje de victoria del total de juegos.
 */
public float percentageOfWinningStates(){
    float porwin=ganados*100/juegosJugados;
    porcenSave=porwin;
    return porwin;
}
```





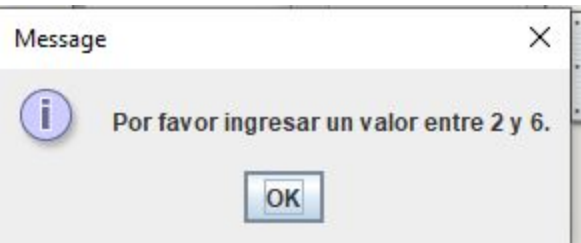
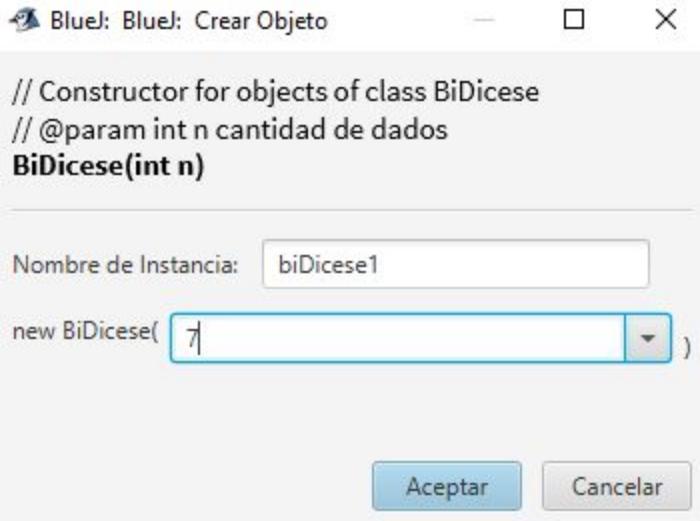
Requisitos de interfaz 9.

Las operaciones se deben ofrecer como métodos públicos de la clase

10. BiDicese debe “sonar” cada vez que llega a un estado ganador



11. Se debe presentar un mensaje amable al usuario si hay algún problema. Consulte y use el método showMessageDialog de la clase JOptionPane.



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

Method Summary		
All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
float	percentageOfWinningStates()	metodo para saber el porcentaje de victoria.
void	playTodos()	metodo para jugar con todas las filas de dados juegos[] es un array de filas de dados
void	playUnaFila(int fila)	metodo para jugar con una fila de dados
void	playUnDado(int fila, int columna)	metodo para jugar con un dado
void	reset()	metodo para reiniciar un juego
void	resultado()	metodo para saber si se gano o no.
void	superwin()	metodo para cambiar los colores de los dados cuando se consigue ganar el juego

2. Planifiquen la construcción considerando algunos miniciclos.

Method Detail

playTodos

```
public void playTodos()
```

metodo para jugar con todas las filas de dados juegos[] es un array de filas de dados

playUnaFila

```
public void playUnaFila(int fila)
```

metodo para jugar con una fila de dados

Parameters:

int - fila parametro para indicar el numero de fila

playUnDado

```
public void playUnDado(int fila, int columna)
```

metodo para jugar con un dado

Parameters:

int - fila , int columna parametro para indicar el numero de fila y columna del dado

resultado

```
public void resultado()
```

metodo para saber si se gano o no.

reset

```
public void reset()
```

metodo para reiniciar un juego

superwin

```
public void superwin()
```

metodo para cambiar los colores de los dados cuando se consigue ganar el juego

percentageOfWinningStates

```
public float percentageOfWinningStates()
```

metodo para saber el porcentaje de victoria.

Returns:

porwin retorna el porcentaje de victoria del total de juegos.

3. Implementen la clase . Al final de cada miniciclo realicen una prueba de aceptación. Capturen las pantallas relevantes.
Los pantallazos se adjuntan en un punto anterior, adjuntamos pruebas de aceptación.

4. Indiquen las extensiones necesarias para reutilizar la clase Dicese. Explique.
realizamos diferentes métodos como manipular un solo dad, otros que tenían que ver con el color y algunos Get necesarios para dar solución al problema.

5. Propongan un nuevo método para enriquecer el juego.


```
/**
 * metodo para cambiar los colores de los dados cuando se consigue ganar el juego
 *
 */
public void superwin(){
    if (win){
        for (int i=0;i<n;i++){
            juegos[i].superWin();
        }
    }
}
```

método que ejecuta rainbow sobre todos los dados a la hora de ganar.

RETROSPECTIVA

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

36 horas

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

El laboratorio está completo y todo está funcional porque manejamos bien los tiempos y la realización de cada mini ciclo se logró con éxito.

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

Programación en pareja, ya que esto fue lo que pusimos en práctica en la realización de este laboratorio y nos fue muy útil.

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

Que todas las implementaciones funcionaran a la perfección, porque fue un gran reto para nosotros ya que tuvimos muchas dificultades con la sintaxis.

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

La lógica de algunos métodos, evaluar varias ideas y ponerlas en práctica

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

Escuchar nuestras ideas, alternar nuestros roles de trabajo, ayudarnos mutuamente.