

PROGRAMACIÓN ORIENTADA A OBJETOS

Introducción. Clases y objetos.

2019-2

Laboratorio 1

SHAPES

Conociendo el proyecto shapes

2. Considerando el diagrama de clases de “shapes”

¿qué clases ofrece?

- Shapes ofrece cuatro clases llamadas rectangle, triangle, circle y canvas

¿qué relaciones existen entre ellas?

- Todas las clases están relacionadas con el canvas

3. De acuerdo con la documentación generada:

¿qué clases tiene el paquete shapes?

- El paquete shapes tiene las clases triangle, circle, rectangle y canvas

¿qué atributos tiene la clase Circle?

- Los atributos son: PI, diameter, xPosition, yPosition, color, isVisible

¿cuáles métodos ofrece la clase Circle para que la figura cambie (incluya sólo el nombre)?

- Los metodos son:
makeVisible, makeInvisible, moveRight, moveLeft, moveUp, moveDown,
moveHorizontal, moveVertical, slowMoveHorizontal, slowMoveVertical, changeSize,
changeColor, draw, erase

4. Revisen el código de la clase Circle. Con respecto a los atributos:

¿cuántos atributos realmente tiene?

- La clase circle tiene 6 atributos

¿cuáles son privados y cuáles públicos?

- Son públicos: PI
- Son privados: diameter, xPosition, yPosition, color, isVisible

Con respecto a los métodos:

¿cuántos métodos tiene en total?

- En total son 15 métodos

¿cuáles son privados?

- Son privados: draw y erase

¿Quiénes usan los componentes privados?

- Únicamente el desarrollador del software

5. ¿Qué no se ve en la documentación?

- En la documentación no está presente el código

¿por qué debe ser así?

- Por seguridad, ya que solo los desarrolladores deben tener acceso a él.

6. En el código de la clase Circle revisen el detalle del atributo PI.

¿qué se está indicando?

- Se indica que el atributo PI es público y estático(No requiere de un objeto para su uso).

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

- Enseñar, diseñar y codificar clases para de esa manera crear objetos.

Manipulando objetos. Usando opciones.

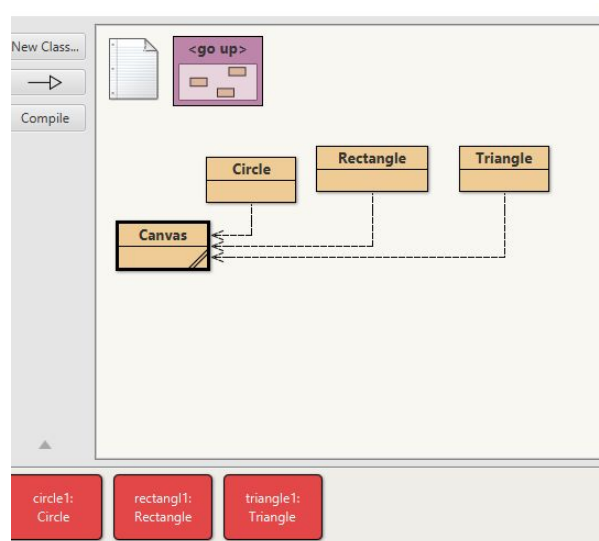
1. Creen un objeto de cada una de las clases que lo permitan.

¿cuántas clases hay?

- Hay dos tres clases que permiten crear objetos(rectangle,triangle,circle)

¿cuántos objetos crearon?

- Creamos 3 objetos uno por cada clase.



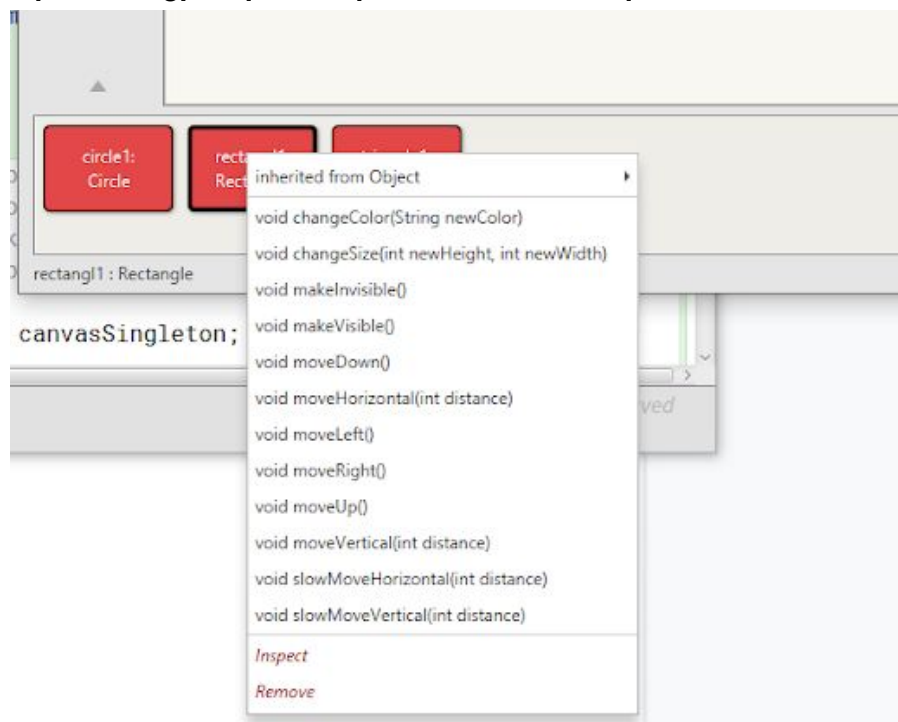
¿por qué?

- Porque de cada clase se puede crear uno o varios objetos.

2. ¿cuáles son los valores de inicio de todos sus atributos? Capturen las pantallas



3. Capture la pantalla. ¿por qué no aparecen todos los que están en el código?



- Por que los faltantes son privados

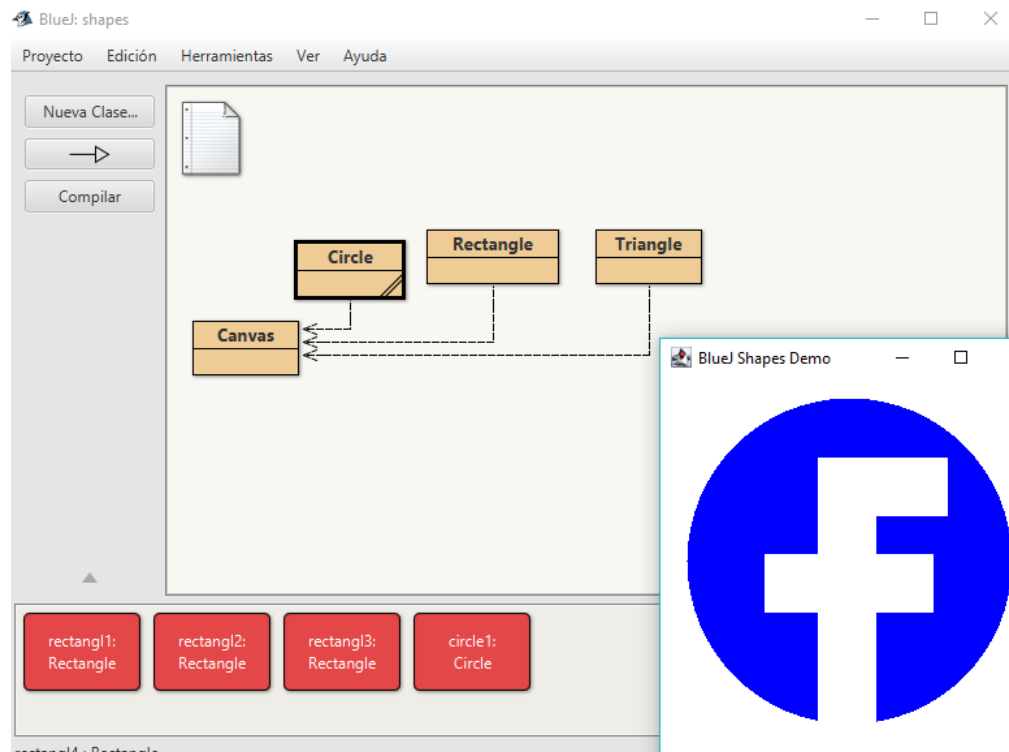
4. Construyan, con “shapes” sin escribir código, una propuesta de la imagen de su app favorito. ¿Cuántas y cuáles clases se necesitan? ¿Cuántos objetos se usan en total? Capturen la pantalla.

¿Cuántas y cuáles clases se necesitan?

- Se necesitaron de dos clases circle y rectangle

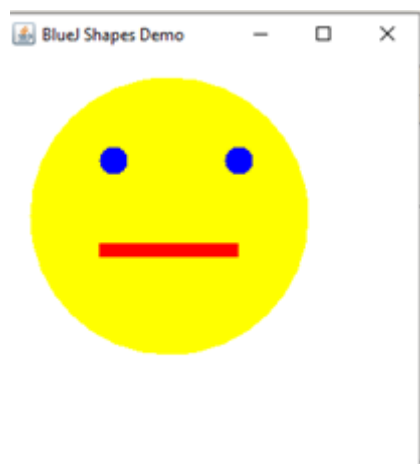
¿Cuántos objetos se usan en total?

- Se usaron 4 objetos 3 rectángulos y un círculo



Manipulando objetos. Analizando y escribiendo código.

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



2. Habiliten la ventana de código en línea , escriban el código y para cada punto señalado indiquen: ¿cuántos objetos existen? ¿qué color tiene cada uno de ellos? ¿cuántos objetos se ven? Expliquen. Capturen la pantalla.

¿cuántas variable existen?

- Existen cuatro variables :face, rEye ,lEye,mouth

¿cuántos objetos existen?

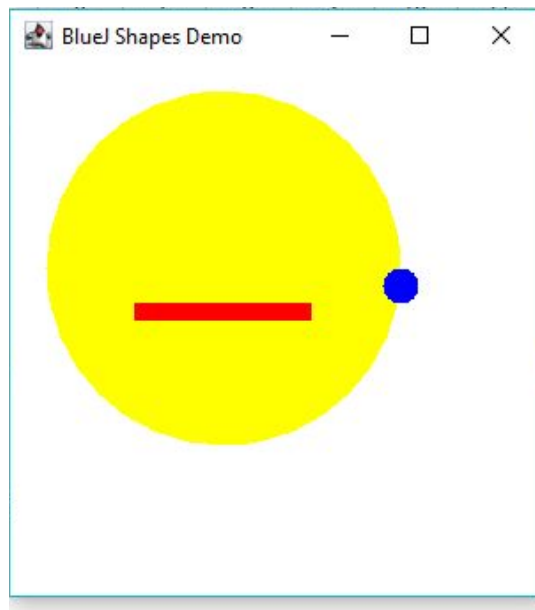
- Existen tres objetos: circle, circle y rectangle

¿qué color tiene cada uno de ellos?

- circle:yellow
- circle:blue
- rectangle:red

¿cuántos objetos se ven?

- Se ven tres objetos, esto es debido a que rEye y lEye se refieren al mismo objeto y si una variable se modifica la otra también



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

- La figura 1 es diferente a la figura 2, ya que la figura 1 tiene un círculo azul de más, esto es debido a que en la figura 2 rEye y lEye se refieren al mismo objeto y si una variable se modifica la otra también

Extendiendo clases

[En lab01.doc y *.java]

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

```
public void growUp(int times){  
    for(int i = 0; i < times; i++){  
        diameter+=1;  
        draw();  
    }  
}
```

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

```
public double area(){
    double result;
    result=PI*(diameter/2)*(diameter/2);
    return result;
}
```

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

```
public void jump(){
    for(int i = 0; i < 15;i++){
        slowMoveVertical(-10);
        slowMoveHorizontal(10);
        slowMoveVertical(10);
        slowMoveHorizontal(10);
        draw();
    }
}
```

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

```
/**
 * Resize circle.
 * @param times the number of times you want to increase
 */
public void growUp(int times){
    for(int i = 0; i < times; i++){
        diameter+=1;
        draw();
    }
}
```

```
/**
 * Returns the area of the circle.
 */
public double area(){
    double result;
    result=PI*(diameter/2)*(diameter/2);
    return result;
}
```

```
/**
 * Makes the circle jump to the right.
 */
public void jump(){
    for(int i = 0; i < 15;i++){
        slowMoveVertical(-10);
        slowMoveHorizontal(10);
        slowMoveVertical(10);
        slowMoveHorizontal(10);
        draw();
    }
}
```

growUp

```
public void growUp(int times)
```

Resize circle.

Parameters:

times - the number of times you want to increase

area

```
public double area()
```

Returns the area of the circle.

jump

```
public void jump()
```

Makes the circle jump to the right.

Implementando una nueva clase. SlotMachine.

1. ¿De qué tipo de elementos es el tragamonedas? ¿Cuáles son los diferentes elementos?



Este es un tragamonedas de frutas. Las posibles frutas son naranja, ciruela, manzana y limón.

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

La probabilidad de ganar en esta máquina es del 4/64 ya que la cantidad de combinaciones que se puede hacer en esta máquina se da por la fórmula

n^r = Cosas para elegir

r = Las cosas que se eligen

n^r

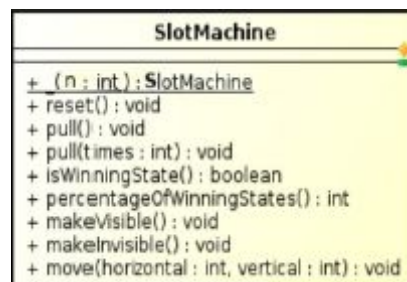
$$4^3 = 64$$

y se coloca el número de posibilidades que hay de ganar sobre todos los posibles resultados, en este caso hay cuatro posibilidades de ganar, una posibilidad por cada una de las 4 frutas

$$4/64 = 0.0625$$

lo que equivale a 6,3%

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



Constructores

- SlotMachine(n:int)

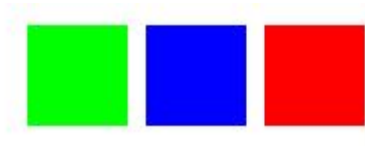
Analizadores

- IsWinningState()
- percentageOfWinningStates()

Modificadores

- reset()
- pull()
- pull(times:int)
- makeVisible()
- makeInvisible()
- move(horizontal:int, vertical:int)

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



MINICICLO 1

```
import java.util.Random;
public class SlotMachine{
    private int n;
    private Rectangle machine[];
    private int c1;
    private int c2;
    private int c3;
    private int horizontal;
    private int vertical;
    private int partidaJugada;
    private int partidaGanada;
    private boolean isVisible;
    private String[] color={"blue", "red", "yellow", "green"};

    public SlotMachine(int n){
        n = n;
        machine=new Rectangle[n];
        for (int x=0;x<n;x++){
            machine[x] = new Rectangle();
            machine[x].changeSize(50,50);
            machine[x].changeColor(color[x]);
            machine[x].moveHorizontal(horizontal+=59);
            machine[x].moveVertical(vertical+=0);
            machine[x].makeVisible();
        }
        horizontal=20;
        vertical=20;
        partidaJugada=0;
        partidaGanada=0;
    }

    public void pull(int times){
        for(int i=0;i<times;i++){
            pull();
        }
    }
}
```



```

public void pull(){
    Random p_1=new Random();
    Random p_2=new Random();
    Random p_3=new Random();
    c1 = p_1.nextInt(4);
    c2 = p_2.nextInt(4);
    c3 = p_3.nextInt(4);
    Canvas canvas=Canvas.getCanvas();

    for(int i=0;i<4;i++){
        canvas.wait(200);
        machine[0].changeColor(color[i]);
        canvas.wait(200);
        machine[1].changeColor(color[i]);
        canvas.wait(200);
        machine[2].changeColor(color[i]);
    }

    machine[0].changeColor(color[c1]);
    machine[1].changeColor(color[c2]);
    machine[2].changeColor(color[c2]);

    if (isWinningState()){
        partidaGanada+=1;
        partidaJugada+=1;
    }
    else{
        partidaJugada+=1;
    }
}

```

```

public boolean isWinningState(){
    for(int i=0;i<n;i++){
        if(machine[i]!=machine[0]){
            return false;
        }
    }
    return true;
}

```

MINICICLO 2

```

public void reset(){
    partidaJugada=0;
    partidaGanada=0;
}

```

```

public double percentageOfWinning(){
    double resultado = 1;
    for (int i = 1; i <= 3; i++) {
        resultado = resultado * 4;
    }
    double x=4;
    double r=x/resultado;
    return r*100;
}

```

MINICICLO 3

```
public void makeVisible(){
    for (int x=0;x<3;x++){
        machine[x].makeVisible();
    }
}

public void makeInvisible(){
    for (int x=0;x<3;x++){
        machine[x].makeInvisible();
    }
}

public void moveHorizontal(int mover){
    if(IsVisible) {
        Canvas canvas = Canvas.getCanvas();
        canvas.erase(this);
    }
    horizontal += mover;
    for (int x=0;x<3;x++){
        machine[x].moveHorizontal(horizontal);
        machine[x].makeVisible();
    }
}

public void moveVertical(int mover){
    if(IsVisible) {
        Canvas canvas = Canvas.getCanvas();
        canvas.erase(this);
    }
    vertical += mover;
    for (int x=0;x<3;x++){
        machine[x].moveVertical(vertical);
        machine[x].makeVisible();
    }
}
```

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.

1 jugada: El porcentaje es de 6,3%

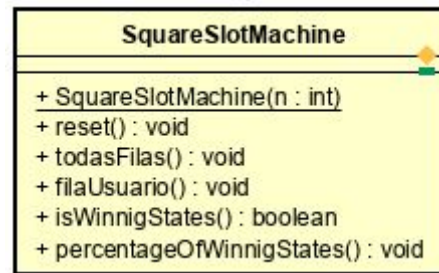
10 jugadas: 1 jugada equivale a tener 6,3% de ganar, cuanto equivaldría 10 jugadas, para esto usamos una regla de tres y el resultado es
 $(10 * 6.3)/1 = 63\%$

100 jugadas: Con la regla anterior la probabilidad de ganar con 100 jugadas es 630%

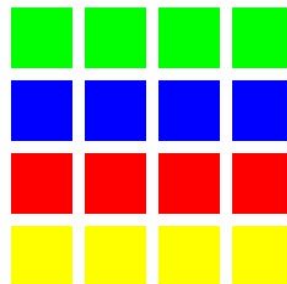
1000 jugadas: Para ganar jugando 1000 veces con lo anteriormente dicho sería 6300%

Definiendo y creando una nueva clase. SquareSlotMachine

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



2. Planifiquen la construcción considerando algunos miniciclos.



```
import java.util.Random;
public class SquareSlotMachine{
    private int n;
    private Rectangle machine[][];
    private int horizontal;
    private int vertical;
    private int partidaJugada;
    private int partidaGanada;
    private String[] color={"green", "blue", "red", "yellow", "purple"};

    public SquareSlotMachine(int n){
        n = n;
        machine=new Rectangle[n][n];
        for (int x=0;x<n;x++){
            for (int z=0;z<n;z++){
                machine[x][z] = new Rectangle();
                machine[x][z].changeSize(50,50);
                machine[x][z].changeColor(color[z]);
                machine[x][z].moveHorizontal(x*60-65);
                machine[x][z].moveVertical(z*60);
                machine[x][z].makeVisible();
            }
        }
        horizontal=20;
        vertical=20;
        partidaJugada=0;
        partidaGanada=0;
    }
}
```

3. Implementen la clase . Al final de cada miniciclo realicen una prueba de aceptación. Capturen las pantallas relevantes.

```
public void reset(){
    partidaJugada=0;
    partidaGanada=0;
}

public void todasFilas(){
    Random p=new Random();
    Canvas canvas=Canvas.getCanvas();

    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            int c1=p.nextInt(n+1);
            canvas.wait(200);
            machine[i][j].changeColor(color[c1]);
            canvas.wait(200);
        }
    }
}

public void filaUsuario(int numfi){
    Random p=new Random();
    Canvas canvas=Canvas.getCanvas();
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            int c1=p.nextInt(n+1);
            canvas.wait(200);
            machine[i][numfi].changeColor(color[c1]);
            canvas.wait(200);
        }
    }
}
```

4. Indiquen las extensiones necesarias para reutilizar la clase SlotMachine. Explique.

Las extensiones reutilizables de la clase serían:

- reset()
- percentageOfWinnigState()
- moveVertical()
- moveHorizontal()

5. Propongan un nuevo método para enriquecer el juego.

Permitir jugar de otra manera, es decir con figuras dentro de los cuadrados, ganando únicamente por figuras, no por colores e ir mostrando las veces jugadas y ganadas.

RETROSPECTIVA

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

(8h/Ortiz Cesar)(8h/Alfaro Angélica)

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

El estado del laboratorio es incompleto, pues nos faltaron cosas por desarrollar y no contamos con el tiempo necesario para terminar.

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

Las prácticas XP más útiles para nosotros fueron:

- La programación por parejas y la planificación, ya que esto nos permitió organizar nuestras ideas para trabajar y de la misma manera ir verificando el código, pues estando los dos en el mismo computador es más fácil identificar errores.

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

El mayor logro fue lograr codificar los métodos ya que se nos dificulta el lenguaje de java.

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

El mayor problema lo encontramos a la hora de desarrollar los métodos propuestos, ya que no contábamos con los conocimientos suficientes en java. Para resolverlo consultamos páginas de internet y videos.

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

Dimos solución a las dificultades que teníamos, teniendo en cuenta las opiniones e ideas del otro..

Invertir más tiempo para reunirnos y trabajar en conjunto, para así afianzar nuestros conocimientos.