



**UNIVERSIDAD
DE LA FRONTERA**

Introducción a la Programación

Taller 7. Figuras Geométricas

Septiembre 2016



Departamento de
Computación e Informática



BlueJ es un entorno de desarrollo Integrado (IDE), el cual provee un editor, un compilador y un entorno de ejecución. BlueJ es una herramienta muy cómoda para los principiantes en la programación con Java.

BlueJ se puede instalar en su computador de escritorio bajo cualquier sistema operativo y su instalador se puede descargar desde el link de su pagina oficial (<http://bluej.org/>).

Esta guía permite acceder rápidamente a los principales elementos de la herramienta para conocer y comenzar a utilizar.

Actividad 1, "Figuras Geométricas"

Java posee ciertas librerías que permiten generar figuras dentro de un área determinada. Estas figuras se pueden "Dibujar" dentro de un Canvas para que sean visualizadas.



Paso 1.1

Acceder a la ruta de los proyectos de ejemplo:

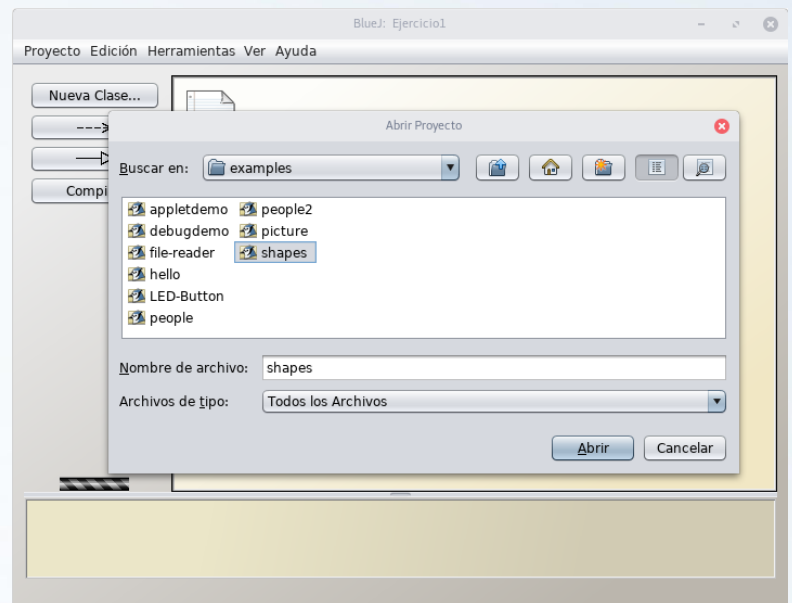
- Windows

C:\Program Files (x86)\BlueJ\examples

- Linux

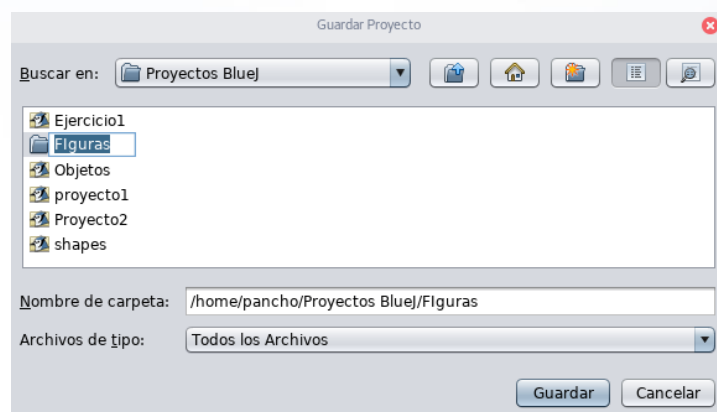
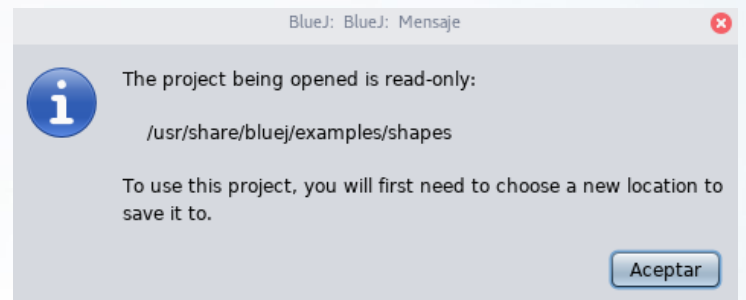
/usr/share/bluej/examples

Luego acceder al proyecto de BlueJ llamado "shapes"

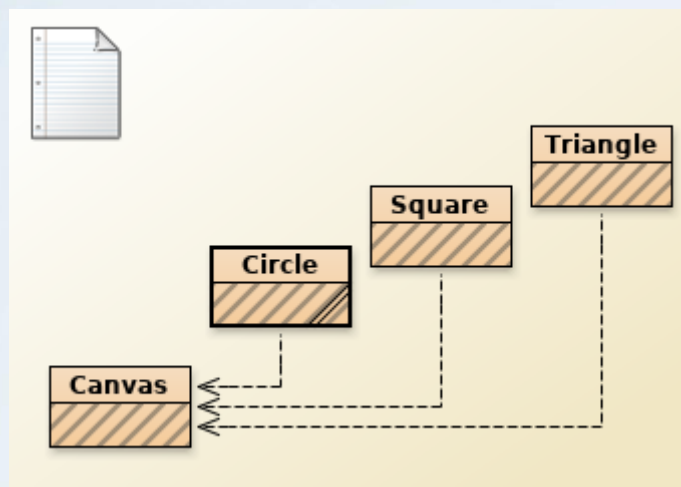


Paso 1.2

Luego será necesario crear una copia del proyecto en alguna carpeta nueva. Con el nombre que se desee..

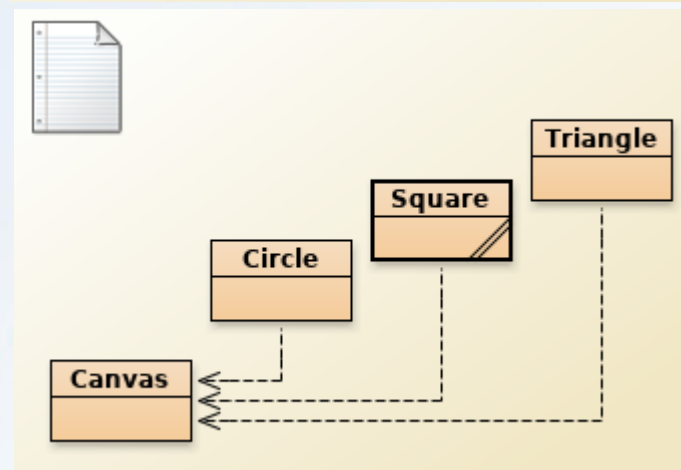


FINISHED?



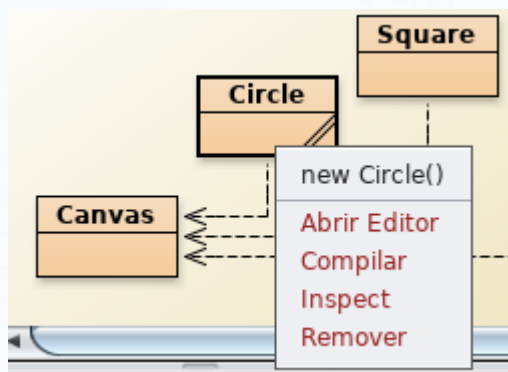
Paso 1.3

- Al abrir el proyecto procedemos a compilar todas las clases correspondientes a las figuras.
- Notará que al estar relacionada la clase Canvas por una flecha esta se compilara de forma automática.



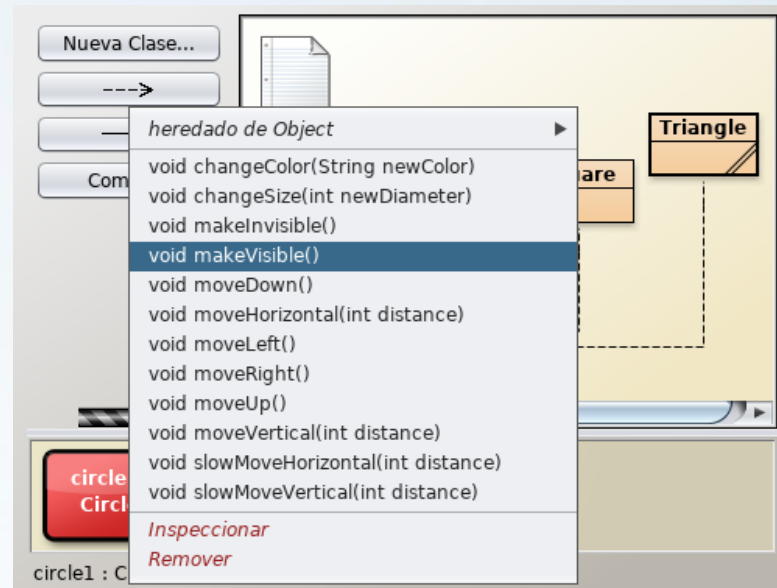
Paso 1.4

- Se procede a crear un nuevo objeto de cada clase de figura.
- El cual se visualizará en la parte inferior



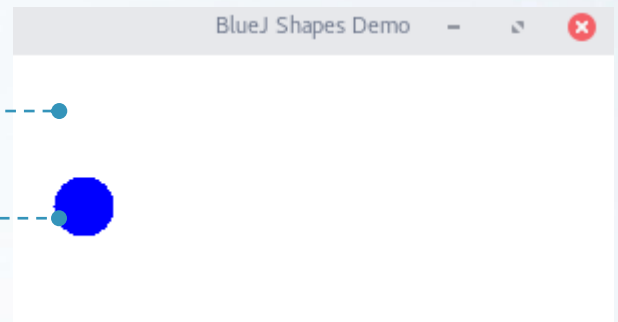
Paso 1.5

- Al hacer Click derecho sobre el objeto de la clase circulo, podemos ver los métodos disponibles para este objeto
 - Para visualizar el objeto, es necesario hacer click en makeVisible().



- El método asigna un valor “true” o verdadero a la variable “isVisible” y luego llama al método “draw()” que es el encargado de:

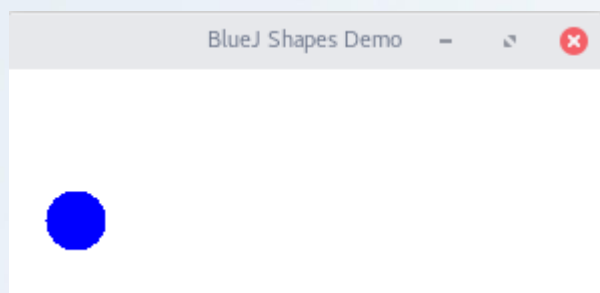
- Crear el un objeto de la clase Canvas
- Dibujar la Figura dentro del canvas.



```
private boolean isVisible;

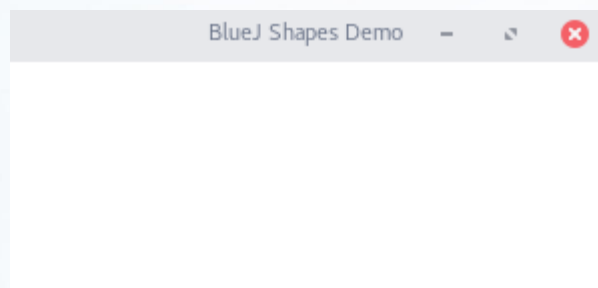
public void makeVisible()
{
    isVisible = true;
    draw();
}

private void draw()
{
    if(isVisible) {
        Canvas canvas = Canvas.getCanvas();
        canvas.draw(this, color, new Ellipse2D.Double(xPosition, yPosition,
            diameter, diameter));
        canvas.wait(10);
    }
}
```

```
public void makeInvisible()
{
    erase();
    isVisible = false;
}
```

```
private void erase()
{
    if(isVisible) {
        Canvas canvas = Canvas.getCanvas();
        canvas.erase(this);
    }
}
```



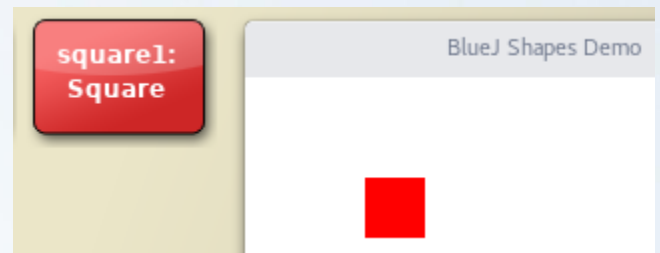
Paso 1.6

- Para poder eliminar u ocultar una figura será necesario llamar al método “makeInvisible()”

Actividad

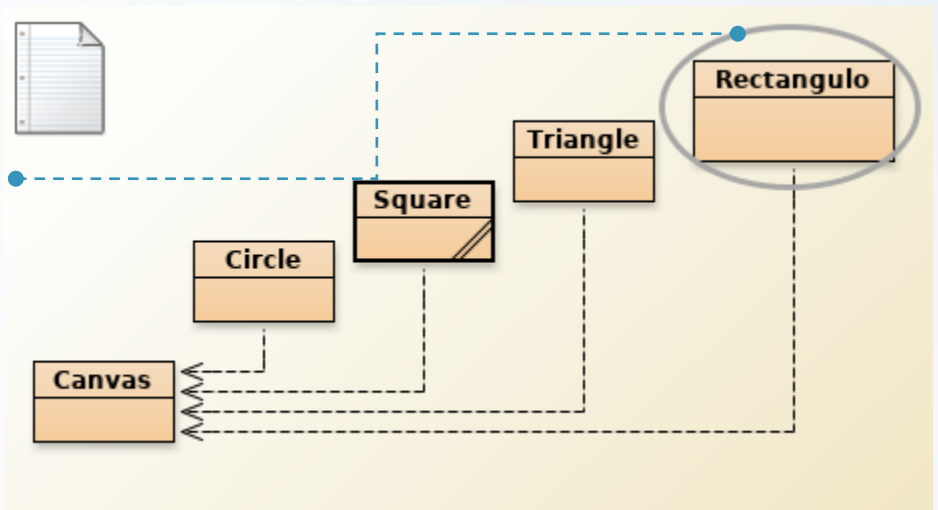
Paso 1

- Crear un objeto de la clase "Square"
- Revisar los métodos de la clase



Paso 2

- Crear una clase Rectangulo



Paso 3

- Copiar código de clase "Square"
- Agregar y modificar atributos

```
private int size;
private int xPosition;
private int yPosition;
private String color;
private boolean isVisible;

private int lado1;
private int lado2;
private int xPosition;
private int yPosition;
private String color;
private boolean isVisible;
```

Paso 4

- Modificar o crear los siguientes métodos para considerar los nuevos atributos de la clase Rectangulo

- Modificar Constructor

```
public Rectangulo()
{
```

- Modificar draw

```
private void draw()
{
```

- Crear método para cambiar el tamaño del lado 1

```
public void changelado1(int newSize)
{
```

- Crear método para cambiar el tamaño del lado 2

```
public void changelado2(int newSize)
{
```

- Elimine el método no necesario la clase Rectangulo.

Actividad 2,"Cuadrado"

Paso 1

- Crear una clase Cuadrado
- Agregar relación de Herencia

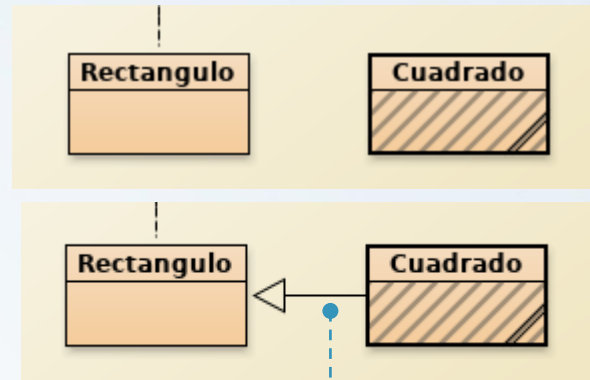
Nota

- La palabra "extends" generada con la flecha continua e punta blanca. De esta forma es posible llamar todas las funciones publicas de la clase "rectángulo".

Paso 2

- Crear el constructor para generar un "Rectángulo" de lados iguales.

- El operador "super", permite llamar de forma directa a una función de la clase padre, en este caso, la clase "Rectangulo".



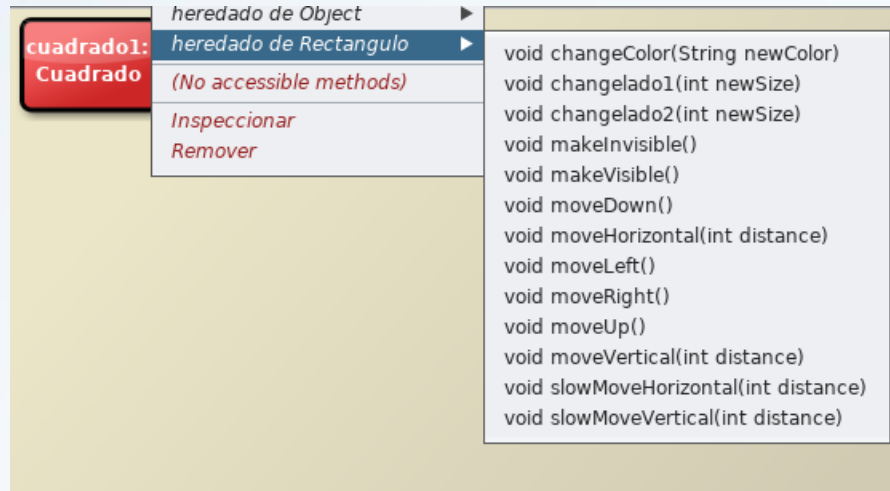
```
public class Cuadrado extends Rectangulo
{
```

```
public class Cuadrado extends Rectangulo
{
    private int lado;
    public Cuadrado(){
        lado = 50;
        super.changelado1(lado);
        super.changelado2(lado);
    }
}
```

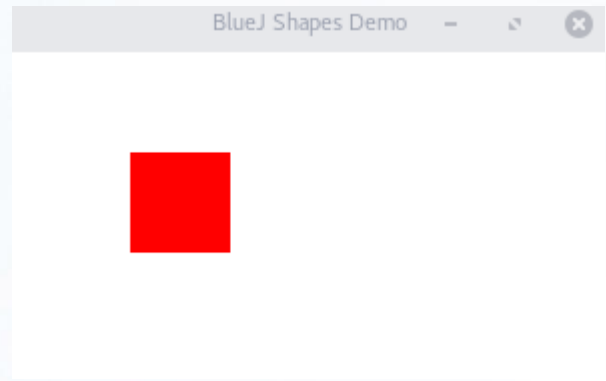
FINISHED?

Paso 3

- Crear un objeto de la clase Cuadrado



- Llamar al método heredado "makeVisible()"



Actividad 3

- Cree un método "area()" en la clase Rectangulo, la cual muestre el área de la figura.
- Cree un método "perimetro()" en la clase Cuadrado, la cual muestre el perímetro de la figura de la figura.
- Llame ambos métodos de un objeto de la clase Cuadrado.

```
public void area(){
    //Muestra el area de la figura
}

public void perimetro(){
    //Mostrar el perimetro del cuadrado
}
```



**UNIVERSIDAD
DE LA FRONTERA**

Curso de Introducción a la Programación

Taller 7. Figuras Geométricas

UNIVERSIDAD DE LA FRONTERA

FACULTAD DE INGENIERÍA Y CIENCIAS

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E INFORMÁTICA

Avda Francisco Salazar 01145
Temuco – Chile / casilla 54-D
Fono (56) 45 2325000 /2744219

dci.ufro.cl



Departamento de
Computación e Informática