

**Asignatura: Programación Orientada a Objetos**

**Curso: 19/20**

**PRÁCTICA**

---

**Objetivos:**

Comprobar que el alumno ha adquirido destreza con el diseño e implementación de programas orientados a objetos.

**Normas:**

La entrega de la práctica es **obligatoria** para aprobar la asignatura.

La práctica se realizará de manera individual.

La práctica se entregará mediante un **fichero ZIP** subido al campus virtual. Dicho fichero deberá contener:

- Un **PDF** con los **diagramas** estáticos de clases que describan las abstracciones desarrolladas, así como un **texto explicativo** de cada clase. Además deben presentarse los diagramas dinámicos que se consideren oportunos para entender el programa (por ejemplo el diagrama de secuencia de dibujo de una línea, o el diagrama de actividad de la operación “save”). La fecha límite de entrega de esta parte es el 10 de noviembre.
- El **proyecto Netbeans** con el código fuente y con el código compilado listo para ejecución mediante un JAR. La fecha límite de entrega de esta parte es el 2 de diciembre.

El día del examen práctico se pedirá una modificación sobre la práctica que ya habrá entregado el alumno. Este examen práctico corresponde al 40% de la nota de la asignatura. Si la modificación solicitada no se realiza o si su implementación es muy incorrecta la práctica estará suspensa.

Además de la modificación solicitada, en la práctica que se entregue el día del examen también se tendrán en cuenta respecto al conjunto completo entregado criterios de: funcionamiento, calidad del código, calidad del diseño, capacidad para escalar y presentación del documento.

**Enunciado:**

La práctica consistirá en el diseño e implementación en Java de un programa de dibujo vectorial llamado **JDrow**.

El programa contendrá en cada momento un **hoja** abierta. En dicha hoja se podrán realizar figuras geométricas mediante la introducción de comandos por la consola.

Toda la comunicación con el programa se realizará mediante la consola de texto y la salida gráfica se realizará usando un navegador. La interacción con el programa consistirá en la escritura de instrucciones mediante el teclado. Así, el sistema

ofrecerá un ***prompt*** y el usuario podrá escribir una instrucción, la cual se ejecutará tras presionar la tecla *Return*.

Se devolverá el texto ERROR si se introduce un comando no válido o que tenga algún error. A continuación se describen los comandos que deberá admitir el *prompt* de JDraw.

## Comandos:

### **line x0,y0,x1,y1**

Dibuja una línea desde la coordenada “(x0,y0)” a la coordenada “(x1,y1)”. En adelante, la coordenada (0,0) es la esquina superior izquierda de la hoja. Por ejemplo, el siguiente comando dibuja una línea del punto (10,10) al punto (50,50):

```
line 10,10,50,50
```

### **pencolor o,r,g,b**

Cambia el color con el que se dibujará el borde de las siguientes figuras. El valor de “o” indica el porcentaje de opacidad y puede variar entre 0 y 1. Los valores de “r”, “g” y “b” están entre 0 y 255 y corresponden a la cantidad de rojo, verde y azul que tendrá el color. Inicialmente el color de relleno es negro con 1 de opacidad (100,0,0,0).

### **fillcolor o,r,g,b**

Cambia el color con el que se dibujará el relleno de las siguientes figuras. El valor de “o” indica el porcentaje de opacidad y puede variar entre 0 y 1. Los valores de “r”, “g” y “b” están entre 0 y 255 y corresponden a la cantidad de rojo, verde y azul que tendrá el color. Inicialmente el color de relleno es negro con 0 de opacidad (0,0,0,0), esto es transparente.

### **width n**

Cambia el ancho de las líneas con el que se dibujará el borde de las siguientes figuras. El valor de n será un número entero. Inicialmente el ancho de las líneas es 1.

### **rect x,y,w,h**

Dibuja un rectángulo con la esquina superior “(x,y)” y ancho “w” y alto “h”.

### **circle x,y,r**

Dibuja un círculo con centro en “(x,y)” y radio “r”.

**text x,y,mensaje**

Escribe un texto en las coordenadas “(x,y)” de la hoja.

**ellipse x,y,rx,ry**

Dibuja una elipse con centro en “(x,y)” y radio en el eje x de “rx” y radio en el eje y de “ry”.

**clear**

Elimina el dibujo actual y deja la hoja vacía.

**undo**

Permite deshacer la última operación realizada. Se podrá invocar de manera consecutiva para eliminar secuencialmente los últimos cambios realizados.

**save ruta y nombre de fichero**

Permite guardar un dibujo a disco. La ruta será relativa al directorio de ejecución de JDraw si no se especifica una ruta absoluta. Dicho fichero contendrá la secuencia de comandos ejecutados desde la última vez que se invocó al comando “clear”.

**load ruta y nombre de fichero**

Permite cargar un dibujo de disco. La ruta será relativa al directorio de ejecución de JDraw si no se especifica una ruta absoluta. Una vez cargado se mostrará el dibujo.

## Tecnología de dibujo:

Si el siguiente fragmento de código HTML (Hyper Text Markup Language) es guardado en un fichero de disco con extensión “html” y luego dicho fichero es abierto con un navegador web podrá comprobar que aparecen las diversas formas geométricas de la figura adjunta.

```

<html>
  <head>
    <meta charset="UTF-8"/>
    <meta http-equiv="refresh" content="2"/>
  </head>
  <body>
    <svg width="1024" height="1024">
      <circle cx="50" cy="50" r="40"
        stroke-width="1" stroke="rgb(0,0,0)"
        fill="rgb(0,0,0)" fill-opacity="0"/>
      <line x1="100" y1="0" x2="200" y2="200"
        stroke-width="1" stroke="rgb(0,255,0)"
        fill="rgb(0,0,0)" fill-opacity="0"/>
      <rect x="10" y="100" width="40" height="60"
        stroke-width="4" stroke="rgb(0,255,0)"
        fill="rgb(255,0,0)" fill-opacity="100"/>
    </svg>
  </body>
</html>

```

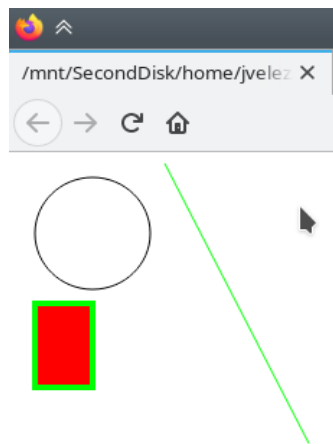


Figura 1.- Formas geométricas obtenidas al abrir con un navegador un fichero que contiene el código HTML adjunto.

Obsérvese que el código HTML anterior contiene una hoja SVG (Scalable Vector Graphics) de 1024x1024. En dicha hoja se encuentran las diferentes ordenes que permiten dibujar dichas figuras geométricas.

En esta práctica se propone que la aplicación JDraw genere un fichero HTML equivalente y que lo abra con el navegador predeterminado del sistema.

Observe que cada vez que JDraw vuelva a generar el fichero HTML el navegador automáticamente actualizará el dibujo. Esto se debe a que en la cabecera de la página web se ha añadido una orden de autorefresco cada 2 segundos.

Así, la siguiente secuencia de comandos generaría el código HTML anterior y por tanto el dibujo de la figura y lo guardaría a disco en un fichero llamado “mi\_dibujo.jd”.

```
circle 50,50,40
pencolor 100,0,255,0
line 100,0,200,200
fillcolor 100,255,0,0
width 4
save mi_dibujo.jd
```

## Recomendaciones:

- Para saber más de SVG se recomienda esta web:  
[https://www.w3schools.com/graphics/svg\\_examples.asp](https://www.w3schools.com/graphics/svg_examples.asp)
- Se recomienda crear una clase que gestione el *prompt*.
- Se recomienda crear una clase por cada comando. Dicha clase tendría métodos para saber si una cadena corresponde a ese comando o métodos para generar el código SVG correspondiente a ese comando.
- Se recomienda mantener los comandos ejecutados como una lista de objetos de tipo String que se ejecuta cada vez que se genera el fichero.
- Se recomienda crear un fichero temporal del sistema para almacenar el HTML que contiene el dibujo.

```
File file = File.createTempFile("jdraw_tmp_file", "html");
String fileName = file.getAbsolutePath();
```

- Se recomienda usar la clase RandomAccessFile para escribir en fichero.

```
RandomAccessFile rfile = new RandomAccessFile(file, "rw");
rfile.writeChars("<html> <body> Hola </body> </html>");
rfile.close();
```

- Se recomienda utilizar los siguientes comandos para abrir el fichero con el navegador al principio de la ejecución de JDraw. Se podrá suponer que el navegador no es cerrado por el usuario durante la ejecución de JDraw.

```
Desktop desktop = Desktop.getDesktop();
desktop.browse(URI.create(file.toURI()));
```