

Práctica 3

Objetivo: Practicar con la creación de objetos y la invocación de sus métodos, para hacer un dibujo animado

Software suministrado: Un conjunto de clases para dibujar figuras en una ventana:

- **Círculo:** representa un círculo
- **Rectángulo:** representa un rectángulo
- **Línea:** representa una línea
- **Imagen:** representa una imagen obtenida de un archivo en formato ***jpg*** o ***png***
- **Figura:** clase auxiliar que representa una figura abstracta
 - es una clase abstracta, lo que impide crear objetos de ella
 - no la usamos directamente

Práctica 3 (cont.)

Coordenadas: las coordenadas de las figuras se describen en *píxeles* (puntos de pantalla)

- el sistema de coordenadas se sitúa en la esquina superior izquierda de la ventana
 - las coordenadas x avanzan hacia la *derecha* en el dibujo
 - las coordenadas y avanzan hacia *abajo* en el dibujo
- la ventana mide 900x900 píxeles

Documentación:

- Para comprender el uso de las clases suministradas puede consultarse la documentación desde un navegador de Internet
 - generar la documentación desde *Bluej* con Tools ->Project Documentation
 - no hacer caso de los métodos `protected`: no son para nosotros
 - esta documentación se guarda en el directorio `doc` dentro del proyecto

Repaso: Crear y usar objetos

Crear objetos de una clase:

- Sintaxis

```
Clase nombreObjeto=new Clase(parámetros);  
// los parámetros son los del constructor
```

- Ejemplo

```
Rectangulo ventana=new Rectangulo(10,10,30,30);
```

Invocar un método `void` del objeto

- Sintaxis

```
nombreObjeto.nombreMétodo(parámetros);  
// los parámetros son los que requiera el método
```

- Ejemplo

```
ventana.mueve(20,30);  
// mueve la ventana 20 píxeles a la derecha  
// y 30 hacia abajo
```

Práctica 3: Descripción

Se pide hacer un programa principal (nueva clase con método `main`) que haga una especie de animación del disparo de un proyectil, con los siguientes pasos:

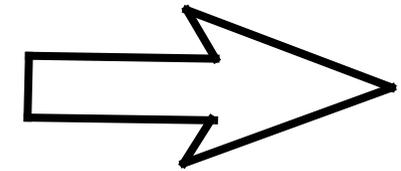
- Crear un objeto de la clase `Imagen` hacia el centro-derecha del dibujo
 - esto será el blanco
 - puede descargarse una foto de Internet
 - pulsando el botón derecho del ratón sobre la imagen, elegir "**Guardar imagen como**", o similar, y guardarla en la carpeta del proyecto
 - elegir una imagen de 500x500 píxeles como máximo
- Crear un objeto de la clase `Circulo` situado hacia la izquierda del dibujo
 - esto será el proyectil
- Esperar un segundo con el método estático `espera` definido en la clase `Figura`
 - ejemplo: `Figura.espera(1000); // espera 1000 ms`

Práctica 3: Descripción (cont.)

- Mover el círculo 40 píxeles hacia la derecha y esperar 200 milisegundos
- Reproducir las acciones del paso anterior tantas veces como sea necesario para que el círculo quede aproximadamente en el centro de la imagen
- Usando el fichero `boom.png` crear un objeto de la clase `Imagen` colocado en el centro de la imagen anterior, tapando el círculo
 - esto representará el impacto del proyectil
 - esta imagen mide 64x64 píxeles

Práctica 3: Parte avanzada

Queremos crear un proyectil compuesto por varias líneas o figuras (al menos 4), como en el ejemplo



Para ello crearemos una nueva clase, llamada `Proyectil` con:

- atributos:
 - los objetos de la clase `Figura` que forman el proyectil
- métodos:
 - Constructor que crea las figuras del proyectil; las coordenadas de la posición inicial central o de la punta del proyectil se le pasarán al constructor como parámetros
 - método que desplaza el proyectil en la cantidad de píxeles horizontales y verticales que se indican como parámetros; lo hace a base de desplazar cada figura individualmente

Práctica 3: Parte avanzada (cont.)

Crear una nueva clase principal, copia de la anterior que haga una animación similar a la de la parte obligatoria, pero con el nuevo proyectil en lugar del círculo original

- solo debería cambiar la creación del objeto proyectil
- el resto debe ser igual

Práctica 3: Entregar

Parte obligatoria

- El código de la clase creada
- Una captura de pantalla del dibujo tal como queda hacia el principio de la ejecución del programa
- Una captura de pantalla del dibujo tal como queda al final de la ejecución

Parte avanzada

- El código de las clases creadas en la parte opcional
- Una captura de pantalla con el dibujo que queda hacia el principio de la ejecución