

Laboratorio S1: Clases y Objetos.

Objetivos:

- ☐ Entender la relación entre clases objetos.
- ☐ Utilizar el Entorno BlueJ para instanciar objetos.
- ☐ Ver la independencia de cada objeto con BlueJ.
- ☐ Usar métodos en BlueJ.
- ☐ Primer acercamiento al lenguaje JAVA.

Herramientas a utilizar:

- ☐ Proyecto: *shapes*.
- ☐ Entorno BlueJ.
- ☐ Lenguaje de programación Java.

Entrega:

- ☐ El final del Laboratorio se entregará un fichero **zip** con el proyecto **shapes** y con el documento del enunciado del laboratorio en el que se habrán respondido las preguntas correspondientes.
- ☐ Nombre del fichero: **apellido_nombre.zip**
- ☐ La entrega es **individual** y a través de Egela al final del laboratorio

Tareas a realizar en el laboratorio

1. Descárgate de **Egela** el proyecto **shapes**. Abre el proyecto **shapes** con BlueJ.
2. Pulsa el botón derecho sobre la caja **Circle**, selecciona **new Circle()** en el menú desplegable y pulsa **Ok**. Acabas de crear (**instanciar**) tu primer **objeto**. La referencia al objeto se guarda en la variable *circle1* y se coloca en el banco de objetos (*object bench*) de BlueJ para su uso posterior.
 - Ahora, pulsa el botón derecho del ratón sobre el objeto circulo creado (*circle1*, en el banco de objetos) y selecciona **Inspect** para invocar al **Inspector de Objetos (Object Inspector)**. Los campos mostrados son los **atributos**. Los atributos describen las características de un objeto. El conjunto de valores que tienen asignados los atributos de un objeto determinan su **estado**.
 - Cuando pulsas el botón derecho del ratón sobre un objeto, además del **Inspect** se muestran todos sus **métodos (comportamientos)**. Los métodos permiten ejecutar acciones (p.ej. mover) y/o modificar el estado del objeto:
 - ¿Qué valor tiene isVisible? _____
 - Pulsa el botón derecho sobre el objeto y selecciona makeVisible(). ¿Qué sucede?
 - ¿Cuál es la posición (x, y) del círculo? _____
 - Pulsa el botón derecho sobre el objeto y selecciona moveDown().
 - ¿Cuál es la nueva posición (x, y) del círculo? _____
 - ¿Qué crees que pasará si **invocas** el método (es decir, **pase de mensaje**) moveDown() de nuevo? Indica el resultado sin ejecutarlo _____
 - ¿Has acertado? _____
 - Invoca el método moveRight() del círculo. ¿Cuál es la nueva posición (x, y) del círculo? _____
3. Crea un nuevo objeto de la clase **Circle** (sigue las instrucciones del ejercicio anterior).

- Pulsa el botón derecho del ratón sobre el objeto y selecciona *Inspect* para invocar al *Inspector de Objetos*.
 - ¿Por qué no puedes ver el círculo en el dibujo? _____
 - Haz que sea visible
 - Cambia el color del círculo (p.ej. "red").
 - ¿Qué ocurre si invocas el método `changeColor()`, y escribes un color *sin* comillas en el cuadro de texto? _____
 - ¿Qué ocurre si especificas un color desconocido? _____
 - Cambia el tamaño del diámetro a 10.
 - Pásale el mensaje de `moveVertical` el valor de 10 píxeles. ¿Qué ha ocurrido con su *estado*, exactamente con el atributo `yPosition`? _____
 - ¿Cómo puedes hacer que el círculo se desplace hacia arriba 25 píxeles? _____
 - Mueve el círculo de manera que quede en la esquina superior izquierda del lienzo (*canvas*).
4. Crea un nuevo objeto de la clase **Square**. Invoca al *Inspector de objetos*, haz visible el objeto y cambia su color a azul
- Muévelo a la esquina superior derecha del lienzo, ¿cómo? _____
 - Pon el cuadrado en el centro del lienzo, ¿cómo? _____
 - Mueve lentamente el cuadrado 50 píxeles a la derecha con el método `slowMoveHorizontal()`.
5. Crea un objeto de la clase **Triangle**. Abre el *Inspector de Objetos* para cada figura.
- Escribe los nombres de los atributos que los 3 tipos de objetos tienen en común: _____
 - Escribe los nombres de los atributos propios de cada figura
 - ¿Qué métodos distinguen el comportamiento del triángulo de las otras figuras?
6. Utilizando el proyecto **shapes** y el menú contextual de *BlueJ* para el **paso de mensajes** a los objetos:
- 6.a. Abre la ventana de terminal de *BlueJ* (*Ver→Mostrar Terminal*) y activa la opción de registro (*Opciones→Registro de llamadas a método*). Si es necesario, limpia lo que tengamos de ejecuciones previas (*Opciones→Limpiar*)
- 6.b. Crea un **triángulo** y modifícalo de manera que la base cubra el borde inferior completo del lienzo y el

vértice superior esté en el borde superior del lienzo. Anota tanto la **instanciación de los objetos** (es decir, creación de las figuras) como el **pase de mensajes** a los objetos (es decir, la invocación de sus métodos)

Objetos Instanciados
<u>Formato:</u> <i>TipoDatos variable= new Constructora(parámetros);</i> <u>Ejemplo:</u> Circle circle1= new Circle();
Pase de Mensajes
<u>Formato:</u> <i>referenciaObjeto.método(parámetros);</i> <u>Ejemplo :</u> circle1.moveRight();

6.c. Abre la clase **Demo** con el editor de código. Copia en el método *exercise6b()* las instrucciones ejecutadas en el ejercicio 6b. Compíllalo y pruébalo.

Tareas complementarias:

7. Utilizando el proyecto **shapes** y el menú contextual de *BlueJ* para el **paso de mensajes** a los objetos:
 - 7.a. Limpia lo que tengas de ejecuciones previas en la ventana de terminal de BlueJ (*Opciones*→ *Limpiar*)
 - 7.b. Pon un cuadrado (**Square**) de tamaño 10x10 en el origen de coordenadas y desplázalo a lo largo del borde superior del lienzo. Cuando llegue a la esquina superior derecha, desplázalo a la esquina inferior derecha del lienzo. Anota tanto la **instanciación de los objetos** (es decir, creación de las figuras) como el **pase de mensajes** a los objetos (es decir, la invocación de sus métodos)

Objetos Instanciados
Pase de Mensajes

7.c. Abre la clase **Demo** con el editor de código. Copia en el método *exercise7b()* las instrucciones ejecutadas en el ejercicio 7b. Pon en el método main una llamada al método *exercise7b()*. Compíllalo

y pruébalo.