Práctica 1

***Objetivo*: Practicar con la herencia múltiple, los iteradores y la comparación**

***Descripción*: Se desea escribir un conjunto de módulos que nos permitan dibujar figuras en una pantalla, y realizar operaciones geométricas con ellas**

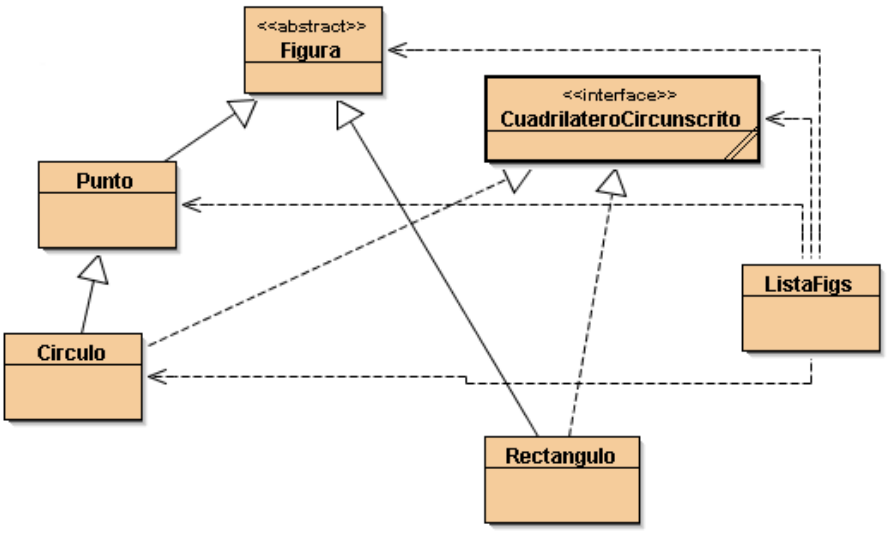
Para dibujar las figuras se usará la clase `Dibujo` del paquete `fundamentos`

La práctica se divide en dos partes:

- **A: clases básicas con herencia múltiple**
- **B: lista de objetos e iteración**

Práctica 1 (cont.)

Jerarquía de clases



Práctica 1 (cont.)

Figura
Dibujo dib ColorFig col
Figura(Dibujo dib, ColorFig col) area() retorna entero nombre() retorna texto coordXCentro() retorna entero coordYCentro() retorna entero dibujar()

CuadrilateroCircunscrito
base() retorna entero altura() retorna entero

Práctica 1 (cont.)

Punto
entero x entero y
Punto(entero x, entero y, Dibujo dib, ColorFig col) nombre() retorna texto coordXCentro() retorna entero coordYCentro() retorna entero dibujar()

Circulo
entero radio
Circulo(entero radio, entero x, entero y, Dibujo dib, ColorFig col) nombre() retorna texto radio() retorna entero area() retorna entero coordXCentro() retorna entero coordYCentro() retorna entero dibujar() altura() retorna entero base() retorna entero

Práctica 1 (cont.)

Rectangulo
entero x1, y1 entero x2, y2
Rectangulo(entero x1, entero y1, entero x2, entero y2, Dibujo dib, ColorFig col) nombre() retorna texto area() retorna entero coordXCentro() retorna entero coordYCentro() retorna entero dibujar() altura() retorna entero base() retorna entero

Práctica 1 (cont.)

Se suministra el siguiente software:

- Clase abstracta `Figura`, interfaz `CuadrilateroCircunscrito`

Parte A:

- Parte A: Implementar `Punto`, `Circulo`, `Rectangulo`
- Implementar una clase `ListaFigs` que permita crear una lista de figuras, usando por ejemplo `ArrayList`, con operaciones:
  - añadir una figura
  - dibujar todas las figuras de la clase; usar la instrucción *for-each*

Práctica 1 (cont.)

Parte B:

- Añadir a la clase `ListaFigs` los siguientes métodos
  - obtener una lista con las figuras cuyo centro está en una zona rectangular dada; usar un *iterador*
  - obtener una lista con las figuras que sean instancias de `CuadrilateroCircunscrito`, y cuya planta (`base*altura`) sea mayor que una cantidad dada
- Hacer un programa principal que facilite la prueba de la clase `ListaFigs`, creando una lista con varias figuras y probando sus métodos

Práctica 1 (cont.)

Entregar

- El código fuente de las clases en un fichero comprimido en formato `zip`, o en formato `jar`

Partes voluntarias

- Implementar la operación `compareTo` e `equals` para las figuras, de modo que para dos figuras A y B:
  - A<B sii area(A)<area(B)
  - A equals B sii (area(A)==area(B) && nombre(A) equals nombre(B))
- Probar estos métodos desde el programa de prueba
- Extender la jerarquía de clases para crear por ejemplo una clase `Elipse`, heredera de `Rectangulo`, que represente la elipse inscrita en el rectángulo