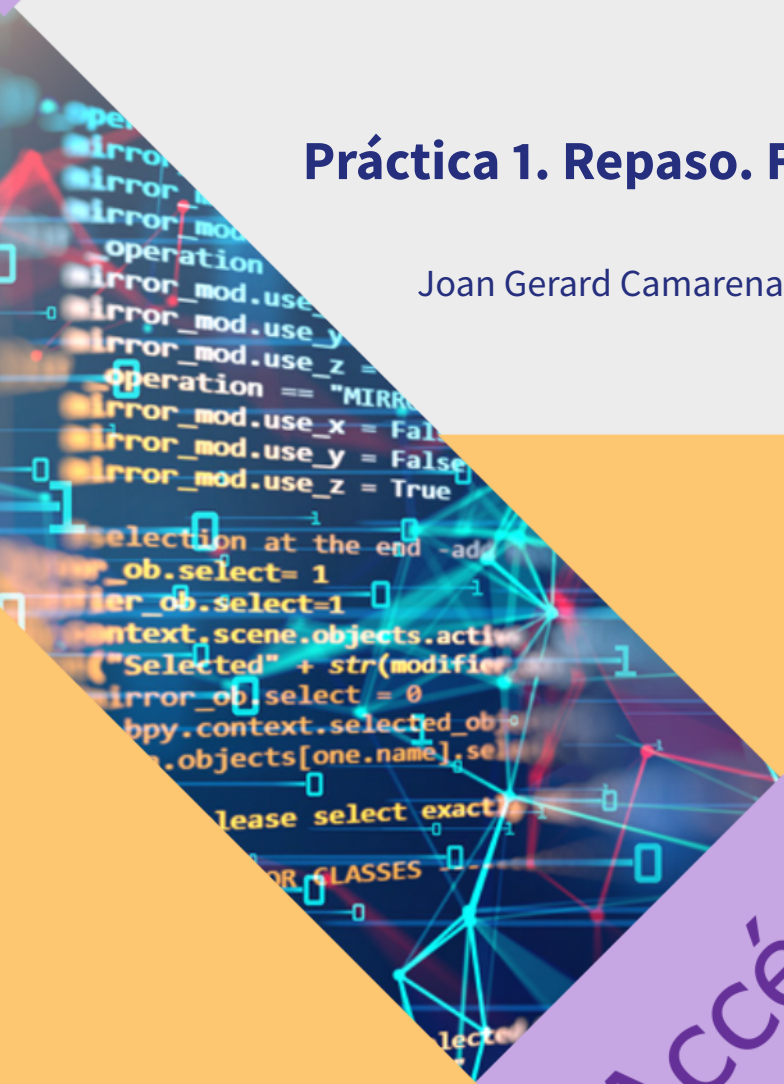
Curs 2020-2021



Práctica 1. Repaso. Figuras Geométricas

Joan Gerard Camarena Estruch

Accés a Dades

Pràctiques

IES Jaume II El Just
Tavernes de la Valldigna

1. Figuas Geométricas 2D.

Para realizar esta práctica vamos a crear una jerarquía de objetos partiendo de una clase que llamaremos `Figura`. De esta clase heredarán una serie de clase que identificarán una serie de figuras geométricas. Además todas estas figuras implementarán una *interfaz* que permitirá al programa dibujar estas figuras en pantalla.

Para la creación de las clases y su implementación seguiremos tadas las directrices vistas en clase en el tema de repaso.

1.1. Clase `Punto`.

Esta clase tendrá dos atributos, `x` e `y`. Nos servirá para poder localizar las figuras y dibujarlas.

1.2 Clase `Figura`.

Será la clase base del arbol de herencia. Contendrá todos los atributos y métodos que necesitamos para realizar cálculos sobre los diferentes tipos de figuras. Como atributos tendrá `origen` de clase Punto que nos servirá para localizar el objeto en el plano. Como métodos tendremos:

- `area()` de tipo double → Calcula el área de la figura
- `perímetro()` de tipo double → Calcula el perímetro de la figura
- `distancia(Figura f)` de tipo double → Calcula la distancia entre la figura y la quele paso como parámetro
- `escalar (int porcentaje)` → Modifica el tamaño de la figura. (100 lo deja igual, 50 a la mitad, 200 el doble y así con cualquier porcentaje).
- `mover(Punto p)` → cambia el punto origen de la figura
- `desplazarh(double x)` → desplaza la figura horizaontalmente (+ derecha, - izquierda)
- `desplazarv(double y)` → desplaza la figura verticalmente (+ arriba, - abajo)

Como atributos tendrá `id` un código, `borde` color del borde, `relleno` color del relleno.

El alumno debe determinar qué metodos se pueden implementar y cuales deben ser abstractos.

1.3. Clases que heredarán de Figura.

- Cuadrado (necesitaremos el lado como atributo)
- Rectángulo (necesitaremos la base y la altura)
- Círculo (necesitaremos el radio)

- Tríangulo, siempre será rectángulo. Pensad qué atributos necesitamos.
- Todas redefinirán `toString()` para poder identificar el tipo de figura, el id, y su posición.

1.4 Interfaz `Dibujable`

Esta interfaz contendrá:

- Constantes para los colores: rojo, verde, azul, amarillo, negro, blanco.
- Métodos: `dibujar()`, `rellenar()`

1.5 Clase `Lienzo`

Clase principal, clase que contendrá una contenedor (libre) con todas las figuras que el usuario quiera crear. Tendrá como métodos:

- dibujar() → Dibujará todas las figuras
- area() → mostrará el área de todas las figuras
- perímetro() → mostrará el perímetro de todas las figuras
- distancia(int id) → dado el identificador de una figura mostrará su distancia con el resto de figuras
- listar() → Dará información de todas las figuras
- escalar (id, porcentaje)
- mover(id, punto)
- desplazarh (id, x)
- despazarv (id, y)

1.6 Ordenación de figuras

Deseamos poder ordenar nuestras figuras, por lo que las deberemos comparar entre ellas por varios criterios. En caso de no indicar lo contrario, las figuras se compararan por su *area*, aunque pueden ordenarse también por su perímetro y por su distancia al origen de coordenadas. Se pide que las figuras implementen la interfaz `Comparable` para compararlas por el área, así como crear varios comparadores (uno para el perímetro y otro para la posición)

1.7 Menú principal

Con el fin de no hacer el menú demasiado extenso, la aplicaicón creará 10 figuras de tipo y en posiciones alaeatorias para comenzar a trabajar. Las opciones serán las siguientes, y llamarán a los métodos de la clase lienzo pertinentes.

1. Listar
2. Dibujar
3. Perímetros
4. Areas
5. Escalar
6. Mover
7. Desplazar
8. Ordenar → Aparecerá un submenú, que nos pregunta por que criterio queremos comparar (area, perímetro o posición), mostrando a continuación todas las figuras ordenadas por dicho criterio. El submenú debe preguntar si el orden de las figuras es ascendente o descendente.