**1.-** Desarrolla una clase Cafetera con atributos:

**capacidadMaxima** (la cantidad máxima de café que puede contener la cafetera)

**cantidadActual** (la cantidad actual de café que hay en la cafetera)

Implementa, al menos, los siguientes métodos:

* **Constructor** predeterminado: establece la capacidad máxima en 1000 (c.c.) y la actual en cero (cafetera vacía).
* **Constructor** con la capacidad máxima de la cafetera; inicializa la cantidad actual de café igual a la capacidad máxima.
* **Constructor** con la capacidad máxima y la cantidad actual. Si la cantidad actual es mayor que la capacidad máxima de la cafetera, la ajustará al máximo.
* **Accedentes y mutadores (getter y setter)**.
* **llenarCafetera()**: hace que la cantidad actual sea igual a la capacidad máxima.
* **servirTaza(int)**: simula la acción de servir una taza con la capacidad indicada por parámetro. Si la cantidad actual de café en la cafetera “no llega” para llenar la taza, se sirve lo que quede.
* **vaciarCafetera()**: pone la cantidad de café actual en cero.
* **agregarCafetera(int)**: añade a la cafetera la cantidad de café indicada en el parámetro. Si esa cantidad excede el máximo, ajusta al máximo.

Escribe el código en el programa principal para probar lo siguiente:

**Capacidad máxima = 1000 Cantidad actual = 0**

**Agregamos 20 c.c. de cafe...**

**Capacidad máxima = 1000 Cantidad actual = 20**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Llenamos la cafetera...**

**Capacidad máxima = 1000 Cantidad actual = 1000**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Servimos una taza de 500 c.c....**

**Todavia quedan 500 c.c.**

**Capacidad máxima = 1000 Cantidad actual = 500**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Servimos una taza de 600 c.c....**

**Se sirve todo lo que quedaba (500 c.c.)**

**Capacidad máxima = 1000 Cantidad actual = 0**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Servimos una taza de 200 c.c....**

**Lo siento, pero no queda nada de cafe.**

**Capacidad máxima = 1000 Cantidad actual = 0**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Llenamos la cafetera con 700 c.c....**

**Capacidad máxima = 1000 Cantidad actual = 700**

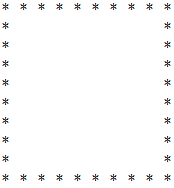
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Llenamos la cafetera con 400 c.c....**

**Capacidad máxima = 1000 Cantidad actual = 1000**

**2.-** Implementa una clase en Java que permita representar cuadrados. Cada objeto Cuadrado vendrá representado por sus cuatro vértices, que serán los atributos de la clase. Además de los métodos para modificar y devolver los valores de los vértices, se piden los siguientes métodos:

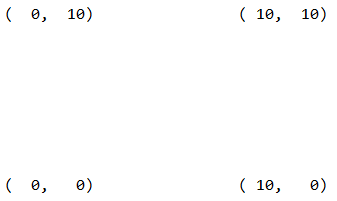
Cada cuadrado tendrá un método (dibujar) para representarse por medio de asteriscos. Por ejemplo, el cuadrado con vértices (0, 0) (0, 10) (10, 10) y (10, 0) se representaría:



También queremos un **método (dibRellenando)** que represente el cuadrado relleno de asteriscos.



Otro **método (dibujarVertices)** para escribir los 4 vértices del cuadrado de la forma siguiente para el ejemplo anterior:



Un **método (esCuadrado)** que devuelva true si se trata de los vértices de un cuadrado y false si no lo es.

La *distancia entre dos puntos* P1: (x1, y1) y P2: (x2, y2) viene dada por la fórmula:

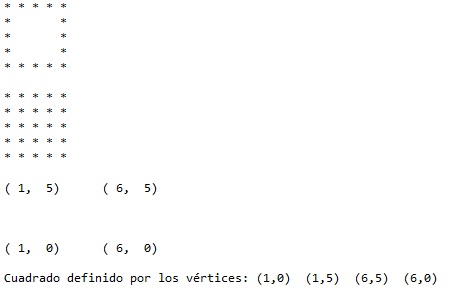
**D=**

Cada cuadrado tendrá un **método (lado)** que devuelve el valor del lado del cuadrado creado.

Escribe un **método (compara**) en la clase Cuadrado para poder comparar el área de dos cuadrados. Este método recibe un cuadrado y lo compara con el cuadrado que invoca el método, actuando igual que el método compare de la clase String.

Implementa luego una clase de prueba que permita crear objetos de la clase Cuadrado y probar todos sus métodos.

*Otro ejemplo:*



Observa que para dibujar los cuadrados vamos a ignorar la distancia al eje de las x y de las y, los pintamos siempre desde el punto P1(x1, y1), teniendo únicamente en cuenta la longitud del lado.

**3.-** Vamos a escribir un programa para representar el consumo de energía de una instalación eléctrica. Para ello, se hará una clase que representa los aparatos conectados en la instalación.

Cada aparato tiene un consumo eléctrico determinado. Al encender un aparato eléctrico, el consumo de energía se incrementa en la potencia de dicho aparato. Al apagarlo, se disminuye el consumo en dicha potencia.

Inicialmente, los aparatos están todos apagados.

Además, se desea consultar el consumo total de la instalación.

Haz un programa que declare dos aparatos eléctricos, una bombilla de 150 watios y una plancha de 2000 watios. El programa deberá imprimir el consumo nada más crear los objetos. Después, se enciende la bombilla y la plancha, y el programa imprime el consumo. Luego se apaga la bombilla, y se vuelve a imprimir el consumo.

**Ejemplo de salida:**

Inicialmente el consumo eléctrico es 0.0

Encendemos la bombilla Potencia 150.0

Encendemos la plancha Potencia 2000.0

El consumo eléctrico es 2150.0

Apagamos la plancha Potencia 2000.0

El consumo eléctrico es 150.0