

### Enunciado 1

Defina una clase Linea con dos atributos: puntoA y \_puntoB. Son dos puntos por los que pasa la línea en un espacio de dos dimensiones. La clase dispondrá de los siguientes métodos:

- Linea(): Constructor predeterminado que crea una línea con sus dos puntos como (0,0) y (0,0).
- Linea(Punto, Punto): Constructor que recibe como parámetros dos objetos de la clase Punto, que son utilizados para inicializar los atributos.
- mueveDerecha(double): Desplaza la línea a la derecha, la distancia que se indique.
- mueveIzquierda(double): Desplaza la línea a la izquierda, la distancia que se indique.
- mueveArriba(double): Desplaza la línea hacia arriba, la distancia que se indique.
- mueveAbajo(double): Desplaza la línea hacia abajo, la distancia que se indique.
- Getters y setters.
- Método que nos permita mostrar la información de la línea de la forma [puntoA,puntoB].

Por ejemplo: [(0.0,0.0),(1.0,1.0)].

### Enunciado 2

Desarrolle una clase Cafetera con atributos \_capacidadMaxima (la cantidad máxima de café que puede contener la cafetera) y \_cantidadActual (la cantidad actual de café que hay en la cafetera). Implemente, 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.
- Getters y setter.
- llenarCafetera(): pues eso, hace que la cantidad actual sea igual a la capacidad.
- servirTaza(int): simula la acción de servir una taza con la capacidad indicada. Si la cantidad actual de café "no alcanza" para llenar la taza, se sirve lo que quede.
- vaciarCafetera(): pone la cantidad de café actual en cero.
- agregarCafe(int): añade a la cafetera la cantidad de café indicada.

### Enunciado 3

Defina los siguientes elementos:

- interface Puerta: con los métodos abrir y cerrar.
- interface PuertaBloqueable: extiende de Puerta, con los métodos bloquea y desbloquea.

- interface Alarma: con los métodos isAlarmaActivada, activarAlarma y desactivarAlarma.
- clase ComponentedeCoche: con los atributos descripción, peso y coste, y un método que muestre dichos atributos.
- clase PuertaCoche, con el atributo boolean estaBloqueada, y que extienda la clase ComponentedeCoche e implemente las interfaces Alarma y Puertabloqueable.

Escriba un programa que pruebe la clase PuertaCoche y todas las operaciones que admite.