BOLETIN 4 – ORIENTACIÓN A OBJETOS

Sistemas de gestión empresarial – Tema 1

***Ejercicio 1*: Números fraccionarios**

- Implementar la clase de los números fraccionarios. El constructor debe, por defecto, poner el denominador = 1. Eso permite representar los números enteros como fracciones.
- Implementar la suma, resta, multiplicación y división.
- Si el resultado de las operaciones es «simplificable», aplicar el algoritmo de Euclides para obtener el resultado simplificado.

***Ejercicio 2:* Rectángulo**

Crea una clase `Rectangulo` que modele rectángulos por medio de cuatro puntos (los vértices). Dispondrá de un constructor, que cree un rectángulo partiendo de sus cuatro vértices. La clase también incluirá un método para calcular la superficie y otro que desplace el rectángulo en el plano.

***Ejercicio 3:* Línea**

Define 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(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.
- muevelzquierda(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.
- Método que nos permita mostrar la información de la línea de la siguiente forma: [puntoA,puntoB]. Por ejemplo: [(0.0,0.0),(1.0,1.0)].

Ejercicio 4: **Cafetera**

Desarrolla 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). Implementa, al menos, los siguientes métodos:

- Constructor con la capacidad máxima de la cafetera; inicializa la cantidad actual de café igual a la capacidad máxima.
- 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

Ejercicio 5: **Fecha**

Crea una clase Fecha con atributos para el día, el mes y el año de la fecha. Incluye, al menos, los siguientes métodos:

- Constructor parametrizado con día, mes y año.
- leer(): pedirá al usuario el día (1 a 31), el mes (1 a 12) y el año (1900 a 2050).
- bisiesto(): indicará si el año de la fecha es bisiesto o no.
- diasMes(int): devolverá el número de días del mes que se le indique (para el año de la fecha).
- valida(): comprobará si la fecha es correcta (entre el 1-1-1900 y el 31-12-2050); si el día no es correcto, lo pondrá a 1; si el mes no es correcto, lo pondrá a 1; y si el año no es correcto, lo pondrá a 1900. Este método se llamará en el constructor parametrizado y en leer().
- corta(): mostrará la fecha en formato corto (02-09-2003).
- diasTranscurridos(): devolverá el número de días transcurridos desde el 1-1-1900 hasta la fecha.
- diaSemana(): devolverá el día de la semana de la fecha (0 para domingo, ..., 6 para sábado). El 1-1-1900 fue domingo.
- larga(): mostrará la fecha en formato largo, empezando por el día de la semana (martes 2 de septiembre de 2003).
- fechaTras(long): hará que la fecha sea la correspondiente a haber transcurrido los días que se indiquen desde el 1-1-1900.

- diasEntre(Fecha): devolverá el número de días entre la fecha y la proporcionada.
- siguiente(): pasará al día siguiente.
- anterior(): pasará al día anterior.
- igualQue(Fecha): indica si la fecha es la misma que la proporcionada.
- menorQue(Fecha): indica si la fecha es anterior a la proporcionada.
- mayorQue(Fecha): indica si la fecha es posterior a la proporcionada.