Ejercicios adicionales tema 4

Ejercicio 1. Crea una clase llamada Libro que guarde la información de cada uno de los libros de una biblioteca. La clase debe guardar el título del libro, autor, número de ejemplares del libro y número de ejemplares prestados. La clase contendrá los siguientes métodos:

- Constructor por defecto.
- Constructor con parámetros.
- Métodos set y get.
- Método *préstamo* que incremente el atributo correspondiente cada vez que se realice un préstamo del libro. No se podrán prestar libros de los que no queden ejemplares disponibles para prestar. Devuelve true si se ha podido realizar la operación y false en caso contrario.
- Método *devolución* que decremente el atributo correspondiente cuando se produzca la devolución de un libro. No se podrán devolver libros que no se hayan prestado. Devuelve true si se ha podido realizar la operación y false en caso contrario.
- Método *toString* para mostrar los datos de los libros. Este método se heredada de Object y lo debemos modificar (override) para adaptarlo a la clase Libro. Escribe un programa para probar el funcionamiento de la clase Libro.

Ejercicio 2. Escribe una clase Cuenta para representar una cuenta bancaria. Los datos de la cuenta son: nombre del cliente (String), número de cuenta (String), tipo de interés (double) y saldo (double).
La clase contendrá los siguientes métodos:

- Constructor por defecto.
- Constructor con todos los parámetros.
- Métodos setters/getters para asignar y obtener los datos de la cuenta.
- Métodos ingreso y reintegro. Un ingreso consiste en aumentar el saldo en la cantidad que se indique. Esa cantidad no puede ser negativa. Un reintegro consiste en disminuir el saldo en una cantidad, pero antes se debe comprobar que hay saldo suficiente. La cantidad no puede ser negativa. Los métodos ingreso y reintegro devuelven true si la operación se ha podido realizar o false en caso contrario.
- Método transferencia que permita pasar dinero de una cuenta a otra siempre que en la cuenta de origen haya dinero suficiente para poder hacerla. Ejemplo de uso del método transferencia: cuentaOrigen.transferencia(cuentaDestino, importe); que indica que queremos hacer una transferencia desde cuentaOrigen a cuentaDestino del importe indicado.

Ejercicio 3. Crea una clase llamada Contador que contenga un único atributo entero llamado cont.
La clase tendrá los siguientes constructores:

- Constructor por defecto.
- Constructor con parámetros para inicializar el contador con un valor no negativo. Si el valor inicial que se recibe es negativo el contador tomará el valor cero como valor inicial.
- Además de los métodos getter y setter, la clase contendrá los métodos:
    - incrementar: incrementa el contador en una unidad.
    - decrementar: decrementa el contador en una unidad. El contador nunca podrá tener un valor negativo. Si al decrementar se alcanza un valor negativo el contador toma el valor cero.

Una vez creada la clase, escribe un método main para probar la clase.

Ejercicio 4. Escribe una clase para representar fracciones. La clase tendrá dos atributos de tipo int: num (numerador) y den (denominador). La clase debe contener los constructores y métodos adecuados para que este método main funcione de forma correcta.

Ejercicio 5. Crea una clase Rebajas con el método descubrePorcentaje, que averigüe el % descuento aplicado a un producto. Para ello el método recibe como parámetro el precio original del artículo y el rebajado.

Ejercicio 6. Implementar la clase Consumo, que forma parte de la centralita electrónica de un coche con los siguientes atributos: kms recorridos, litros consumidos, velocidad media y precio de la gasolina. Se deben implementar los siguientes métodos:

- getTiempo: indicará el tiempo empleado en realizar el viaje.
- consumoMedio: consumo medio del vehículo (en litros cada 100Km).
- consumoEuros: consumo medio del vehículo (en euros cada 100km).

Ejercicio 7. Realizar el control de una votación en la que se puede presentar un número cualquiera de candidatos. En cada momento se puede votar a cualquier candidato y se pueden pedir los siguientes datos:

- Nombre de un candidato concreto, edad y el número de votos que lleva hasta el momento.
- Nombre del candidato más votado hasta el momento y número de votos que lleva conseguidos como atributos estáticos.

Desarrollaremos los métodos:

- buscaMejorCandidato(int edadMinima) que pasando como parámetro la edad mínima para ser representante, mira si su edad es superior a la mínima y si el número de votos es mayor de 100, devolviendo verdadero si se cumplen las dos condiciones y falso en caso contrario.
- costeCampaña(double costePublicidad, double costesVarios) que recibe como parámetro los costes totales de toda la campaña y devuelve el coste total suponiendo que por cada voto se deben restar 100€ al bloque costesVarios; además si la edad de cada candidato es mayor de 50 años se restan 20€ al coste total.