

Ejercicio 1

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.
- Constructor **copia**.

La clase contendrá los siguientes **métodos**:

- Métodos **setter** y **getter**.
- Método **incrementar**: incrementa el contador en una unidad.
- Método **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 2

Crea una clase llamada **Cuenta** que tendrá los siguientes atributos: **titular** y **cantidad** (puede tener decimales).

El titular será obligatorio y la cantidad es opcional. Crea dos **constructores** que cumplan lo anterior.

Crea sus métodos **get**, **set**.

Tendrá dos **métodos** especiales:

- **ingresar(double cantidad)**: se ingresa una cantidad a la cuenta, si la cantidad introducida es negativa, no se hará nada.
- **retirar(double cantidad)**: se retira una cantidad a la cuenta, si restando la cantidad actual a la que nos pasan es negativa, la cantidad de la cuenta pasa a ser 0.

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

Ejercicio 3

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 **constructores**:

- **Constructor por defecto.**
- **Constructor con parámetros.**

La clase contendrá los siguientes **métodos**:

- Métodos **Setters/getters**.
- 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.

Escribe un programa para probar el funcionamiento de la clase Libro.

Ejercicio 4

Escribe una clase **CuentaBancaria** 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 **constructores**:

- **Constructor por defecto**
- **Constructor con todos los parámetros**
- **Constructor copia.**

La clase contendrá los siguientes **métodos**:

- 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.

Prueba el funcionamiento de la clase Cuenta.

Ejercicio 5

Haz una clase llamada **Persona** que siga las siguientes condiciones:

- Sus atributos son: nombre, edad, DNI, sexo (H hombre, M mujer), peso y altura. No queremos que se accedan directamente a ellos. Piensa que modificador de acceso es el más adecuado, también su tipo. Si quieres añadir algún atributo puedes hacerlo.
- Por defecto, todos los atributos menos el DNI serán valores por defecto según su tipo (0 números, cadena vacía para String, etc.). Sexo será hombre por defecto, usa una constante para ello.
- Se implantarán varios constructores:
 - Un constructor por defecto.
 - Un constructor con el nombre, edad y sexo, el resto por defecto.
 - Un constructor con todos los atributos como parámetro.
- Los métodos que se implementarán son:
 - **calcularIMC()**: calcula si la persona está en su peso ideal (peso en $\text{kg}/(\text{altura}^2 \text{ en m})$), si esta fórmula devuelve un valor menor que 20, la función devuelve un -1, si devuelve un número entre 20 y 25 (incluidos), significa que está por debajo de su peso ideal la función devuelve un 0 y si devuelve un valor mayor que 25 significa que tiene sobrepeso, la función devuelve un 1. Te recomiendo que uses constantes para devolver estos valores.
 - **esMayorDeEdad()**: indica si es mayor de edad, devuelve un booleano.
 - **comprobarSexo(char sexo)**: comprueba que el sexo introducido es correcto. Si no es correcto, será H. No será visible al exterior.
 - **toString()**: devuelve toda la información del objeto.
 - **generaDNI()**: genera un número aleatorio de 8 cifras, genera a partir de este su número su letra correspondiente. Este método será invocado cuando se construya el objeto. Puedes dividir el método para que te sea más fácil. No será visible al exterior.
 - Métodos **set** de cada parámetro, excepto de DNI.

Ahora, crea una clase ejecutable que haga lo siguiente:

- Pide por teclado el nombre, la edad, sexo, peso y altura.
- Crea 3 objetos de la clase anterior, el primer objeto obtendrá las anteriores variables pedidas por teclado, el segundo objeto obtendrá todos los anteriores menos el peso y la altura y el último por defecto, para este último utiliza los métodos set para darle a los atributos un valor.
- Para cada objeto, deberá comprobar si está en su peso ideal, tiene sobrepeso o por debajo de su peso ideal con un mensaje.
- Indicar para cada objeto si es mayor de edad.
- Por último, mostrar la información de cada objeto.