

1) Lee el **enunciado completo** del ejercicio antes de empezar a codificar. Crea el proyecto **FusionCuentas** en el que hay que utilizar la clase `CuentaCorriente` del ejercicio 4 de la hoja 1. Modifica el código de la clase para poder hacer lo que se pide:

- El banco quiere saber cuántas cuentas activas existen. Añade los elementos necesarios para poder controlar esto.
- Si un cliente cierra una cuenta, se añadirá la cadena " CERRADA" al nombre del cliente que contenga el atributo `cliente` de la cuenta que se va a cerrar y se pondrá el saldo a cero – al hacer esto el número de cuentas activas disminuye en una unidad-.
- Añade un método estático que se llame **fusiona** que reciba como parámetros dos objetos de la clase `CuentaCorriente` y devuelve un objeto de clase `CuentaCorriente`. Este método debe crear una nueva cuenta – que devolverá como parámetro de salida - cuyo saldo será la suma de los saldos de las cuentas que se le han pasado en los parámetros de entrada y, a continuación cerrará dichas cuentas. Al hacer la fusión de dos cuentas en una, hay que tener en cuenta que:
  - Ambas deben pertenecer al mismo cliente. La nueva cuenta tendrá como cliente al cliente de una de las cuentas antiguas, pero se le asignará un número de cuenta nuevo.
  - No se pueden fusionar dos cuentas si tienen el mismo número de cuenta porque eso nos facilitaría duplicar nuestro dinero.

Hay que comprobar que se cumplen estas condiciones antes de crear la nueva cuenta. Si falla una de las dos no se crea la nueva cuenta, no se cierran las antiguas y se devuelve `null`.

2) En el proyecto del ejercicio anterior crea la clase `GestionFusion` que:

Pida al usuario tres nombres de clientes y cree una cuenta para cada uno con 100€ de saldo inicial.

- Muestre en pantalla los datos de las tres cuentas.
- Cierre la primera cuenta.
- Intente fusionar la segunda y la tercera cuenta en otra nueva.
- Imprima los datos de todas las cuentas, incluyendo la que haya resultado de la fusión en caso de que se haya podido hacer.

3) Crea el proyecto **LibroUnAutor**, que estará formado por las siguientes clases:

Autor
-nombre: String (no tiene valor por defecto) -email: String (no tiene valor por defecto) -genero: char (los posibles valores son 'm' o 'f')
+Autor(nombre: String, email: String, genero: String) +getNombre(): String +getEmail(): String +setEmail(email: String): void +getGenero(): char + cadenaAutor (): String

El método `cadenaAutor()` devolverá la siguiente cadena de caracteres (por ejemplo):

"Autor[nombre = Arturo Perez Reverte, email = apreverte@correo.com, genero = m]"

Libro
-titulo: String (no tiene valor por defecto) -autor: Autor -precio: double -cantidad: int = 0
+Libro(titulo: String, autor: Autor, precio: double) +Libro(titulo: String, autor: Autor, precio: double, cantidad: int) +getTitulo(): String +getAutor(): Autor +getPrecio(): double +setPrecio (precio: double): void +getCantidad(): int +setCantidad(cantidad: int): void +cadenaLibro(): String

Supondremos que un libro sólo tiene un autor y que el método `cadenaLibro()` devolverá una cadena de caracteres como la siguiente (por ejemplo):

"Libro [titulo = El asedio,  
Autor [nombre = Arturo Perez Reverte, email = apreverte@correo.com, genero = m]  
precio = 18,90  
cantidad = 3]"

Escribe la clase **GestionLibroAutor**, que además del método `main` deberá tener otros métodos estáticos para que al ejecutar la aplicación muestre un menú con las siguientes opciones:

- 1- Crear libros. (Esta opción preguntará al usuario cuántos libros quiere crear y a continuación procederá a instanciarlos)
- 2- Modificar autor.
- 3- Modificar libro.
- 4- Listado de libros.
- 5- Listado de autores.
- 0- Salir.

- 4) Crea un nuevo proyecto, al que llamarás **LibroVariosAutores** que utilice las clases del ejercicio anterior y modifica la clase `Libro` para que un libro pueda tener varios autores.

El método `cadenaLibro()` debe devolver, por ejemplo:

"Libro [titulo = Cuentan que cuentan

{ Autor [nombre = Silvia Schujer, email = schujer@correo.com, genero = f ], [Laura Devetach, ldevet@edu.com, genero = f], [Emma Wolf, wolf@correo.com , genero = f]}

precio = 14

cantidad = 6]"