

Tarea 07 – Desarrollo de clases Avanzado

Cuando se crea cada cuenta se almacena en un `ArrayList` `cuentasClientes` de tipo `CuentaBancaria`, el cual se encarga de instanciarlo la `clase Interface Ioperaciones`.

En el constructor de la clase `CuentaBancaria`, del que heredan los demás tipos de cuenta, se llama al método `almacenarCuenta()` que añade al `ArrayList` la nueva cuenta creada,

Métodos comunes a varias operaciones.

Método `MenuApp.seleccionarCuenta()`

- Se comprueba que hay cuentas introducidas en el sistema: `comprobarExisteCuenta()`
- Se listan todas las cuentas que existen, mediante el método de la `Clase CuentaBancaria` `previewCuentas()`
- Se le pide al usuario que introduzca el ID de la cuenta deseada
- Se llama al método de la `Clase CuentaBancaria` `buscarCuenta()`, que devuelve la posición en el `ArrayList` `cuentasBancarias`, de la cuenta que corresponda con el ID indicado.
- Mediante el método `comprobarIndiceCuenta()` Se asegura que la cuenta existe, si `buscarCuenta()` devuelve -1, la cuenta no existe

Método `MenuApp.introducirImporte()`

- Solicita al usuario que introduzca el importe con el que quiere operar.
- Llama al método `comprobarImporte()` y le pasa por parámetro el importe introducido, para que compruebe que no es un valor negativo

Métodos de clase `CuentaBancaria`.

Los métodos `@Override` se llamara al implementado en una clase u otra dependiendo del tipo de la cuenta que se haya escogido.

- **Método `previewCuenta(): String`**
 - Busca la posición de la cuenta con la que queremos operar.
- **Método `previewCuenta(): String`**
 - Imprime un resumen de los detalles del objeto `CuentaBancaria`
- **Método `toString(): String`**
 - Imprime los detalles del objeto `CuentaBancaria`, dependiendo del tipo de cuenta llamará al método `@Override` del tipo correspondiente
- **Método `detallesCuenta(indice)`**
 - Recibe el índice de la cuenta seleccionada en el `método seleccionarCuenta()` de la `clase MenuApp`
 - Llama al método `toString()` del objeto correspondiente a la posición en el `ArrayList` `cuentasClientes` pasado mediante el índice devuelto por el `método seleccionarCuenta()`
- **Método `ingresarEfectivo(importe)`**
 - Recibe el importe con el que se quiere realizar la operación
 - Lee el saldo actual de la cuenta y le suma el importe mediante el `método getSaldo()`
 - Actualiza el saldo de la cuenta mediante el `método setSaldo()`
 - Devuelve el nuevo saldo de la cuenta