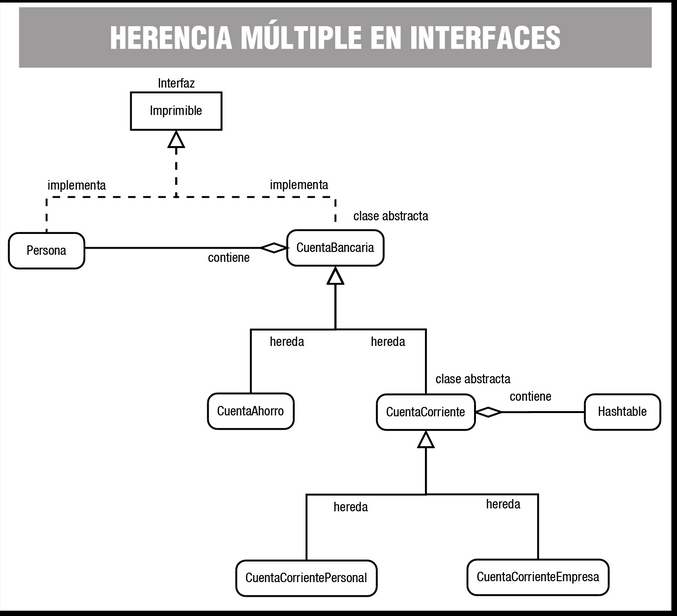
|  |
| --- |
|  |
| **Enunciado.** |
| A lo largo de esta unidad has terminado de familiarizarte con el resto de conceptos relacionados con la **Programación Orientada a Objetos** que faltaban por ver de una manera más formal y con ejemplos explícitos: **composición**; **herencia**; **clases y métodos abstractos**; **sobrescritura de métodos**; **interfaces**; **polimorfismo**; **ligadura dinámica**, etc.  Has experimentando con todos estos conceptos y los has utilizado en pequeñas aplicaciones para comprobar su funcionamiento y su utilidad.  Una vez finalizada la unidad se puede decir que tienes ya un dominio adecuado del lenguaje Java como un lenguaje que permite aplicar todas las posibilidades de la **Programación Orientada a Objetos**. Dado ese supuesto, esta tarea tendrá como objetivo escribir una pequeña aplicación en Java empleando algunas de las construcciones que has aprendido a utilizar.  Se trata de desarrollar una aplicación Java que permita gestionar varios tipos de **cuentas bancarias**. Mediante un menú se podrán elegir determinas operaciones:   1. **Abrir una nueva cuenta**. 2. **Ver un listado de las cuentas disponibles** (código de cuenta, titular y saldo actual). 3. **Obtener los datos de una cuenta concreta**. 4. **Realizar un ingreso en una cuenta**. 5. **Retirar efectivo de una cuenta**. 6. **Consultar el saldo actual de una cuenta**. 7. **Salir de la aplicación.**   Las cuentas se irán almacenando en alguna estructura en memoria según vayan siendo creadas. Cada cuenta será un objeto de una clase que contendrá la siguiente información:   * **Titular** de la cuenta (un objeto de la clase **Persona**, la cual contendrá información sobre el titular: **nombre**, **apellidos**, **fecha de nacimiento**). * **Saldo** actual de la cuenta (número real). * **Número de cuenta** (**CCC - Código Cuenta Cliente**). * **Tipo de interés** anual (si se trata de una **cuenta de ahorro**). * **Lista de entidades** autorizadas para cobrar recibos de la cuenta (si se trata de una **cuenta corriente**). * **Comisión de mantenimiento** (para el caso de una **cuenta corriente personal**). * **Tipo de interés por descubierto** (si es una **cuenta corriente de empresa**). * **Máximo descubierto permitido** (si se trata de una **cuenta corriente de empresa**)   Las **cuentas bancarias** pueden ser de dos tipos: **cuentas de ahorro** o bien **cuentas corrientes**. Las **cuentas de ahorro** son **remuneradas** y tienen un determinado **tipo de interés**. Las **cuentas corrientes** no son remuneradas, pero tienen asociada una **lista de entidades autorizadas** para cobrar **recibos domiciliados** en la cuenta.  Dentro de las **cuentas corrientes** podemos encontrar a su vez otros dos tipos: las **cuentas corrientes personales**, que tienen una **comisión de mantenimiento** (una cantidad fija anual) y las **cuentas corrientes de empresa**, que no la tienen. Además, las **cuentas de empresa** permiten tener una cierta cantidad de **descubierto** (**máximo descubierto permitido**) y por tanto un **tipo de interés por descubierto** y una **comisión fija** por **cada descubierto** que se tenga. Es el único tipo de cuenta que permite tener **descubiertos**.  Cuando se vaya a abrir una nueva **cuenta bancaria**, el **usuario de la aplicación** (**empleado del banco**) tendrá que solicitar al **cliente**:   * **Datos personales**: **nombre**, **apellidos**, **fecha de nacimiento**. * **Tipo de cuenta** que desea abrir: **cuenta de ahorro**, **cuenta corriente personal** o **cuenta corriente de empresa**. * **Saldo inicial**.   Además de esa información, el **usuario de la aplicación** deberá también incluir:   * **Número de cuenta** (**CCC**) de la nueva cuenta. Debe ser válido (habrá que comprobarlo). * **Tipo de interés** de remuneración, si se trata de una **cuenta de ahorro**. * **de mantenimiento**, si es una **cuenta corriente personal**. * **Máximo descubierto permitido**, si se trata de una **cuenta corriente de empresa**. * **Tipo de interés por descubierto**, en el caso de una **cuenta corriente de empresa**. * **Comisión fija por cada descubierto**, también para el caso de una **cuenta corriente de empresa**.   La aplicación deberá asegurarse que la información introducida sea **válida y coherente** (CCC válido; saldos, comisiones y tipos de interés positivos, etc.).  El programa que escribas debe cumplir al menos los siguientes requisitos:   * Para almacenar los objetos de tipo cuenta podrás utilizar cualquier estructura de almacenamiento que consideres oportuna (ArrayList, Hashtable, etc.). * Para trabajar con los datos personales, debes utilizar una clase **Persona** que contenga la información sobre los datos personales básicos del cliente (**nombre**, **apellidos**, **fecha de nacimiento**). * Para trabajar con el número de cuenta debes utilizar el modelo de **Código Cuenta Cliente** (**CCC**), que es posible que también la ya hayas usado en otras unidades. * Para guardar las entidades autorizadas a cobrar recibos debes utilizar una Hashtable que contenga pares de tipo (código de entidad (String), máxima cantidad autorizada para un recibo).   El proyecto deberá contener al menos los siguientes archivos fuente Java:   * **Programa principal** (clase con método main: **AplicacionCuentaBancaria.java**). * Un archivo por cada clase o interfaz que hayas implementado. |

**Estructura del programa:**

****

Tenemos una **clase persona** con sus datos de nombre apellido y fecha de nacimiento validadas.

Tenemos otra clase **Cuenta Bancaria** que será abstracta y de la que dependerán el resto.

La clase **Cuenta de ahorro** y **cuenta Corriente** heredan de la clase **Cuenta Bancaria** y sobrescribirán el método ToString para mostrar los datos como deseemos

La clase **Cuenta corriente** es una clase abstracta de la que hereda las clases **Cuenta corriente personal** y **cuenta corriente Empresa**.

La clase **cuenta corriente empresa** tiene los métodos consultar saldo y pedir efectivo sobrescritos ya que esta clase tiene la particularidad d poder estar en negativo el saldo

La clase **Validaciones** se encarga de los métodos para pedir cantidades, menús, datos y comprobar que estos cumplen con los valores requeridos

La clase **enumerada** **entidades de cobro** realiza una enumeración de las entidades que posiblemente pueden tener derechos de cobro de recibos para más tarde en otros métodos mostrar un menú donde se elegirán por índice que será mostrado. si fuese necesario se podrían ampliar esta enumeración añadiendo otra entidad a la enumeración sin tener que realizar ningún cambio al programa. El método de asignación a una cuenta corriente permite introducir varias entidades guardándolas todas ellas en un Arraylist que es un atributo de las clases