

0485.- Programación.

[Área personal](#) / [Mis cursos](#) / [Programación](#) / [U4 - POO. Clases. Objetos.](#) / [U4: Tarea 1](#)

U4: Tarea 1

En todas estas actividades, además de crear la clase que se pide, deberéis usar la clase en el programa principal (main): Crear un objeto de esa clase, actualizar algún atributo, utilizar todos sus métodos... Debeís subir por tanto la clase con el Main y la clase de cada ejercicio.

Actividad 1: Diseñar la clase **CuentaCorriente**, sabiendo que los datos que caracterizan a un objeto de ese tipo son: saldo, límite de descubrimiento (cantidad de dinero que se permite sacar de una cuenta que ya está a cero), nombre y DNI del titular. Las operaciones típicas con una cuenta corriente son:

- Crear la cuenta: se necesita el nombre y DNI del titular. El saldo inicial será 0 y el límite de descubierto será de -50 euros.
- Sacar dinero: solo se podrá sacar dinero hasta el límite de descubierto. El método debe indicar, por pantalla, un mensaje que indique si ha sido posible llevar a cabo la operación. Además, deberá devolver un valor booleano que indique lo mismo.
- Ingresar dinero: se incrementa el saldo.
- Mostrar información: muestra la información de la cuenta corriente.

Actividad 2: En la clase CuentaCorriente, sobrecargar los constructores para que permitan crear objetos:

- solo con el saldo inicial, no serán necesarios los datos del titular. Por defecto el límite de descubierto será 0 euros.
- con un saldo inicial, con un límite de descubierto y con el DNI del titular de la cuenta.

Actividad 3: Modificar la visibilidad de la clase CuentaCorriente para que sea visible desde clases externas, y la visibilidad de sus atributos para que:

- saldo y limite no sean visibles para otras clases
- nombre sea público para cualquier clase
- dni solo sea visible por clases vecinas

(Comprueba la visibilidad desde el programa principal comentando cada acceso o intento de acceso)

Actividad 4: Todas las cuentas corrientes con las que vamos a trabajar pertenecen al mismo banco. Añadir un atributo que almacene el nombre del banco (que es único) en la clase CuentaCorriente. Diseñar un método que permita modificar el nombre del banco (al que pertenecen todas las cuentas corrientes)

Actividad 5: Diseñar la clase **Texto** que gestiona una cadena de caracteres con algunas características:

- La cadena de caracteres tendrá una longitud máxima, que se especifica en el constructor.
- Permite añadir un carácter, al principio o al final, siempre y cuando exista espacio disponible.
- Igualmente, permite añadir una cadena, al principio o al final del texto, siempre y cuando no se rebase el tamaño máximo establecido.
- Es necesario saber cuántas vocales (mayúsculas y minúsculas) hay en el texto.

Actividad 6: Diseñar la clase **Banco** de la que interesa guardar su nombre, capital y la dirección central. Los bancos tienen las siguientes restricciones:

- a) Siempre tienen que tener un nombre, que no puede ser modificado
- b) Si no se especifica, todos los bancos tienen un capital por defecto de 5.2 millones de euros al crearse.

c) El capital y la dirección de un banco son modificables.

Modificar la clase CuentaCorriente para que cada una esté vinculada a un objeto de tipo Banco. Escribir los métodos necesarios en la clase CuentaCorriente para gestionar el banco al que pertenece (cambiar de banco, mostrar la info de la cuenta, incluidos los datos del banco si lo tiene vinculado). Existe la posibilidad de que una cuenta corriente no esté vinculada a ningún banco.

Actividad 7: Se quiere definir una clase que permita controlar un sintonizador digital de emisoras FM; concretamente, se desea dotar al controlador de una interfaz que permita subir (up) o bajar (down) la frecuencia (en saltos de 0.5 MHz) y mostrar la frecuencia sintonizada en un momento dado (display). Supondremos que el rango de frecuencias a manejar oscila entre los 80MHz y los 108MHz y que, al inicio, el controlador sintonice a 80MHz. Si durante una operación de subida o bajada se sobrepasa uno de los dos límites, la frecuencia sintonizada debe pasar a ser la del extremo contrario. Escribir un pequeño programa principal para probar su funcionamiento.

Actividad 8: Modelar una casa con muchas bombillas, de forma que cada bombilla se pueda encender o apagar individualmente. Para ello hacer una clase Bombilla con una variable privada que indique si está encendida o apagada, así como un método que nos diga el estado de la bombilla. Además, queremos poner un interruptor general, de forma que si saltan los fusibles, todas las bombillas se muestran como apagadas. Cuando el fusible se repara, las bombillas vuelven a estar encendidas o apagadas, según estuvieran antes del percance. Cada bombilla se enciende o se apagan individualmente, pero solo responde si su interruptor particular está activado y además hay luz general.

Actividad 9: Implementar las clases con sus atributos y constructores para gestionar una empresa ferroviaria, en la que se distingue dos grandes grupos: personal y maquinaria. En el primero se ubican todos los empleados de la empresa, que se clasifican en 3 grupos: maquinistas, mecánicos y jefes de estación. De cada uno de ellos hay que guardar:

Maquinistas: nombre completo, dni, sueldo mensual, rango

Mecánicos: nombre completo, tlfn y especialidad (frenos, hidráulica,...)

Jefes estación: nombre completo, dni.

En cuestión de maquinaria podemos encontrar trenes, locomotoras, y vagones. De cada uno hay que considerar:

Vagones: tienen una capacidad máxima de carga (en kilos), una capacidad actual (carga en un momento dado), y el tipo de mercancía con el que están cargados.

Locomotoras:disponen de una matrícula, potencia de su motor y un año de fabricación. Cada locomotora tiene asignado un mecánico que se encarga de su mantenimiento.

Trenes:están formados por una locomotora y un máximo de 5 vagones. Cada tren tiene asignado un maquinista que es responsable de él.

Todas las clases correspondientes al personal serán de uso público. Entre las clases relativas a la maquinaria solo será posible construir, desde clases externas, objetos tipo Tren y de tipo Locomotora. La clase Vagon será solo visible por sus clases vecinas.

Actividad 10: Escribir un programa que lea por teclado una hora cualquiera y un número n que representa una cantidad de segundos. El programa mostrará la hora introducida y la hora resultante de sumar a esa hora, la cantidad de segundos indicada. Para ello hemos de diseñar previamente la clase Hora que dispone de los atributos hora, minuto y segundo. Los valores de los atributos se controlarán mediante métodos set/get.

Estado de la entrega

Estado de la entrega	No entregado
Estado de la calificación	Sin calificar
Fecha de entrega	martes, 31 de diciembre de 2019, 23:59
Tiempo restante	La Tarea está retrasada por: 351 días 16 horas
Última modificación	-