

Actividad de cátedra

ELI147 – Introducción a las Tecnologías de la Información
II Semestre 2018



1. Información general

Semana:	29/04 al 04/05
Contenidos:	Uso de clases y objetos con anidamiento y responsabilidades.
Objetivo:	El objetivo es que el estudiante esta semana sea capaz de realizar sus propios programas interactivos aplicando los conceptos de clases y objetos en problemas de negocios con el manejo de anidamiento y responsabilidades de clases.

2. Ejercicios

En su último viaje a China, Don Adrónico, dueño de Mi Banco, ha llegado con muy buenas ideas que desea aplicar a su sistema bancario. Para ello considere:

Cada **Cuenta** tiene rut, nombre del cliente, número de cuenta, dirección y saldo en la cuenta; hay casos en que una cuenta puede ser creada con un código de seguridad que es un texto el cual ayuda a bloquear la cuenta. Hay que considerar que el **Banco** puede tener clientes con distintas cuentas, para ello maneja una colección compacta sin pLibre de sucursales, en donde cada **Sucursal** tiene un número de sucursal, dirección, región del país, el nombre del gerente y un arreglo no compacto de referencias a instancias tipo Cuenta.

El main contendrá una referencia a un objeto de tipo clase Banco. Implemente las siguientes funcionalidades dónde corresponda:

1. Implementar las clases.
2. Agregar un cliente a una sucursal, para ello deberá verificar que la cuenta y el cliente no existan en el banco.
3. Depositar y girar dinero, para ello deberá realizar las validaciones de los saldos. Esta operación no se podrá realizar en el caso que la cuenta esté protegida por código y el código ingresado no coincida con el de la cuenta.
4. Mover a un cliente de sucursal, para ello deberá indicar sucursal de origen y de destino y mover la cuenta entre las sucursales.
5. Eliminar sucursal, para ello deberá mover a todos los clientes a una nueva sucursal. Considere que las cuentas de los clientes deben moverse a la misma región de la que se va a eliminar. Si esto no es posible mover a cualquier otra sucursal.
6. Sucursal con clientes pudientes: Se deberá determinar la sucursal que acumula mayor saldo en la sumatoria de los saldos de dinero de las cuentas de sus clientes.
7. Eliminar cuenta cliente, para ello se da el número de cuenta y el rut y deberá eliminar la cuenta del sistema.

3. Entregable

El canal de televisión **TVCartoon** ha decidido automatizar ciertas tareas, para ello se solicita a Ud. implementar una aplicación en Java que cumpla con las características que se describen a continuación. TVCartoon tiene un gerente general y un listado de programas, los cuales por presupuesto y horarios no pueden ser menos de 20. Se necesita que los programas sean manejados usando un arreglo compacto que utiliza pLibre. El constructor de TVCartoon recibe su nombre del gerente y la cantidad de programas que se manejarán. La cantidad de programas debe ser validada dentro del constructor.

Cada **programa** tiene un nombre, el cual no se repite con otro programa, un bloque horario el cual se guarda como texto ya que es solo informacional y el rating promedio que tiene el programa (ej: 10.5). Además, se sabe que, debido a la nueva forma de hacer televisión, un programa puede tener más de 1 conductor, es por ello que el programa también debe administrar los conductores, que a lo más serán 5. Se requiere que los conductores se almacenen en un arreglo no

compacto. El constructor de un programa recibe su nombre, su bloque horario y su rating.

Cada **conductor** tiene un rut y un nombre. Se puede dar que un conductor además cuente con horas contratadas y un sueldo, los cuales son condicionados, es decir uno no puede existir sin el otro. Para esto, debe implementar dos constructores: uno que reciba sólo el rut y el nombre, y otro que reciba todos los datos.

Con lo señalado, se le pide escribir los métodos y extractos de aplicación en Java junto a las clases que implementan las entidades descritas. Para ello, escriba su código en base a las siguientes preguntas:

Pregunta 1 – Escribir los siguientes miembros en las clases descritas anteriormente:

- Atributos privados. Debe utilizar como identificador de los atributos los campos resaltados.
- Constructores de las clases respectivas. Estos deben dar respuesta a las posibles formas de creación de los objetos indicadas anteriormente.
- Getters y setters, sólo los que sean estrictamente necesarios para el funcionamiento de los métodos a desarrollar en las preguntas 2 y 3. Además, en la aplicación, debe instanciar un objeto de tipo clase TVCartoon, usando valores literales para invocar el constructor.

Pregunta 2 – Métodos en TVCartoon (y otros métodos que sean necesarios en las otras clases para respetar el encapsulamiento):

- `public boolean agregarConductor(Programa nuevo)`
- `public boolean agregarConductor (String nombrePrograma, String rut, String nombreCond)`
- `public boolean agregarConductor(String nombrePrograma, String rut, String nombreCond, int horasContratadas, int sueldo).`

Debe agregar el conductor al programa indicado. El constructor debe ser implementado acorde a los datos entregados en la sobrecarga respectiva. Considere que:

- Debe validar que exista un programa con el nombre entregado, para poder agregarlo en la instancia respectiva.
- Si existe algún conductor en cualquiera de los programas de TVCartoon con el rut entregado por parámetro, debe descartar los otros parámetros entregados en estos métodos y utilizar el objeto del conductor encontrado para agregarlo al programa respectivo.
- El conductor no puede quedar repetido en el programa donde se pretende agregar. Considere que esta validación debe ser implementada en la clase Programa.
- Debe haber espacio para agregar el conductor al programa indicado.
- En la aplicación, debe especificar un ejemplo de cómo se invocaría cada uno de los métodos sobrecargados, usando la instancia de TVCartoon creada en la pregunta 1 e imprimiendo un mensaje de resultado según lo que retorne la invocación. Use valores literales para la invocación.

Pregunta 3 – Método en TVCartoon (y otros métodos que sean necesarios en las otras clases para respetar el encapsulamiento):

- `public Conductor[] eliminarPrograma(String nombre)`
- Debe quitar de TVCartoon el programa con el nombre entregado por parámetro. Recuerde que el arreglo de programas debe quedar siempre compacto. Considere que:
- Debe retornar un arreglo de largo exacto (es decir, no puede contener valores null) con las referencias a los objetos de los conductores de la instancia del programa eliminado. Si el programa no tenía conductores, retornará null.
- También debe retornar null en caso de que el programa no exista.
- Debe invocar desde el main este método entregando al usuario el resultado. Debe mostrar el nombre de los conductores obtenidos.

ENTREGA: Entrega hasta el día 06 de mayo hasta las 23:00 hrs. El nombre del proyecto en netbeans, la carpeta y el archivo comprimido deben llamarse igual. Debe contener su nombre y dos apellidos, estar escrito en mayúscula y separados por un guion bajo como en el ejemplo:

SEBASTIAN_DAVALOS_PEREZ