



Programación Java



Clases Abstractas e Interfaces

Práctica integradora

Objetivo

El objetivo de esta guía práctica es estudiar a detalle el uso de las clases abstractas e interfaces.

¿Are you ready? 😎👍



Ejercicio 1

Un banco tiene diferentes tipos de transacciones que puede llevar a cabo, entre ellas se encuentran: Depósito, Transferencia, Retiro de Efectivo, Consulta de Saldo, Pago de Servicios. Todas las transacciones tienen dos métodos en común, que son `transaccionOk()` y `transaccionNoOk()`.

A partir de esto existen diferentes tipos de clientes que pueden interactuar con el banco:

- **Ejecutivos:** Realizan Depósitos y Transferencias.
- **Basic:** Realizan consultas de saldo, pagos de servicios y retiro de efectivo.
- **Cobradores:** Realizan retiro de efectivo y consulta de saldos.

Implementar el escenario según corresponda para permitir a los clientes mencionados con anterioridad, el acceso a los diferentes métodos según la operación que deseen realizar.



Ejercicio 2

Se plantea desarrollar un programa que permita mediante una interfaz imprimir diferentes tipos de documentos.

Entre los tipos de documentos se encuentran:

- **Curriculums:** incluye a una persona con todos sus atributos más una lista de sus habilidades.
- **Libros en PDF:** Incluyen atributos como cantidad de páginas, nombre del autor, título y género.
- **Informes:** Incluyen un texto de n longitud, cantidad de páginas, autor, y revisor.



Representar un escenario donde se creen cada uno de estos objetos y que, por medio de un método estático de una interfaz imprimible, se pueda pasar cualquier tipo de documento y sea impreso el contenido.



Ejercicio 3

Se solicita la creación de una clase abstracta denominada *Animal*, de la cual deriven 3 animales: *Perro*, *Gato* y *Vaca*. Los 3 animales son capaces de “emitir sonidos”, para ello será necesario implementar un método que permita mostrar por pantalla el sonido que emite cada animal. Por ejemplo, en el caso del perro “guau”, el gato “miau” y la vaca “muuu”.

A partir de esto, teniendo en cuenta los gustos alimenticios de cada animal (gato y perro son considerados carnívoros y la vaca un hervíboro), incluir las interfaces necesarias para contemplar los métodos *comerCarne* o *comerHierba*.

Una vez realizado lo mencionado, llevar a cabo en el *Main* la creación de diferentes animales y la invocación de sus respectivas implementaciones de métodos.

Como propuesta extra se sugiere llamar a un método *comerAnimal* donde a partir del pasaje de un objeto de cualquier tipo de animal como parámetro, invoque al método para comer según corresponda a dicho animal.