

## Ejercitación

### Ejercicio 1: Clientes y Cuentas

1. Definir una nueva clase **Cliente** que contenga los siguiente atributos:
  - Nombre
  - Apellido
2. Definir una nueva clase **Cuenta** que contenga los siguientes atributos:
  - Número de cuenta
  - Saldo
  - Titular (el titular es un objeto de tipo Cliente).
3. Crear un método llamado **depósito** que tenga como parámetros de entrada **cantidad de dinero**. Al ser un depósito la cantidad de dinero se incrementa al saldo. El método debe imprimir por pantalla el tipo de transacción realizada y el nuevo saldo.
4. Crear un método llamado **extracción** que tenga como parámetros de entrada **cantidad de dinero**. Al ser una extracción, la cantidad se resta del saldo. En caso que el monto de la extracción a realizar sea mayor al saldo disponible, se debe imprimir por pantalla "Fondos insuficientes". En el otro caso, debe imprimir por pantalla el tipo de transacción realizada y el nuevo saldo.
5. Definir una nueva clase llamada **Main** y crear un método main. Dentro del método main crear a dos clientes, seteando los atributos que sean necesarios, y crear una cuenta para cada una de ellas, también seteando los atributos necesarios. Luego, realizar un depósito y una extracción en cada cuenta.

**Aclaración:** definir, para cada clase, los getters y setters necesarios.

### Ejercicio 2: Jugadores y Entrenamiento

1. Definir una nueva clase **JugadorDeFutbol** que contenga los siguiente atributos:
  - Nombre
  - Energía
  - Felicidad
  - Goles
  - Experiencia
2. Definir una nueva clase **SesionDeEntrenamiento** que contenga los siguientes atributos:
  - Experiencia que otorga cuando se hace un entrenamiento
3. Crear un método llamado **hacerGol** en jugador, el cual le cuesta al Jugador 5 de energía y le da 10 de felicidad. Además le aumenta en 1 la cantidad de goles al jugador.. El método debe imprimir por pantalla "GOOOOL!".
4. Crear un método llamado **correr** en jugador, el cual le cuesta al Jugador 10 de energía. El método debe imprimir por pantalla "No me dan mas las piernas".

5. Crear un método llamado **entrenarA** que tenga como parámetros de entrada **un jugador de fútbol**. Al ser un entrenamiento, un jugador tiene que:
  - Correr
  - Hacer un Gol
  - CorrerY luego de hacer estas tres cosas, se debe incrementar la experiencia del jugador sumándole los puntos de experiencia que otorga el entrenamiento.  
El método debe imprimir la experiencia inicial del jugador y la final.
6. Definir una nueva clase llamada **Main** y crear un método main. Dentro del método main crear a dos jugadores de fútbol, seteando los atributos que sean necesarios, y crear una sesión de entrenamiento con los atributos necesarios y que ambos jugadores realicen un entrenamiento.

**Aclaración:** definir, para cada clase, los getters y setters necesarios.

### Ejercicio 3: Atletas y Prueba

1. Definir una nueva clase **Atleta** que contenga los siguientes atributos:
  - Nombre
  - Nivel
  - Energía
2. Definir una nueva clase **Prueba** que contenga los siguientes atributos:
  - Dificultad Requerida
  - Energía Requerida
3. Crear, en la clase **Prueba**, el método **puedeRealizar**, que tenga como parámetros de entrada **un objeto de tipo Atleta**. El método debe devolver **True**, si el atleta puede realizar la prueba o **False** en caso contrario. Un atleta puede realizar la prueba si su nivel es mayor o igual a la dificultad de la prueba y si tiene suficiente energía para afrontarla.
4. Definir una nueva clase llamada **Main** y crear un método main. Dentro del método main crear a dos atletas, seteando los atributos que sean necesarios, y crear tres pruebas para cada una de ellas, también seteando los atributos necesarios. Luego, controlar si los atletas pueden realizar dichas pruebas.

**Aclaración:** definir, para cada clase, los getters y setters necesarios.

### Ejercicio 4: Trípode

El **trípode** es un aparato de tres patas y parte superior circular o triangular, que permite estabilizar una cámara y evitar el movimiento propio de este. Queremos modelar el comportamiento de este objeto.

1. Definir una clase **Trípode** que contenga los siguientes atributos:
  - Plegado: indica el estado actual del trípode (plegado o no)
  - Altura mínima
  - Altura máxima
  - Altura actual.

2. Definir el método **definirAltura(Integer nuevaAltura)**, que reciba una altura y modifique el valor de la altura actual.
3. Definir el método **plegar()**, que permita plegar el trípode.
4. Definir el método **desplegar()**, que permita desplegar el trípode.
5. Definir el método **guardar()**, que permita dejar el trípode listo para guardar. Es decir, debe estar plegado y su altura actual debe ser la mínima posible.
6. Definir el método **listoParaGuardar?()**, que permita controlar si el trípode está listo para guardarse. Es decir, debe estar plegado y con altura actual mínima.
7. Definir el método **usar()**, que permita usar el trípode. Un trípode está listo para usar si está desplegado y con altura mayor a la mitad de la altura máxima.
8. Definir el método **listoParaUsar?()**, que permita controlar si el trípode está listo para ser usado. Es decir, debe estar desplegado y con altura mayor a la mitad de la altura máxima.
9. Definir una nueva clase llamada **Main** y crear un método main. Dentro del método main crear un trípode y hacer pruebas con los métodos definidos anteriormente.

## Ejercicio 5: Vehículos Usados

Se desea desarrollar un sistema informático para una concesionaria que comercializa vehículos usados. Esta concesionaria decidió que solo venderá autos. Todos los autos tienen una marca, un modelo, un año de fabricación, un color y un kilometraje.

Por una estrategia de marketing, la concesionaria quiere llevar registro de las clientes que fueron adquiriendo vehículos. El departamento de marketing de la concesionaria nos informó que lo importante de un cliente es el nombre, el apellido y algún dato de contacto para poder ofrecerle nuevos autos en un futuro.

Por otro lado, el departamento contable de la concesionaria quiere llevar un registro de las ventas realizadas. Cada registro consiste de la siguiente información:

- Monto de la venta
  - Vehículo vendido
  - Cliente
1. Realizar un diagrama de clase que represente el modelo que queremos implementar.
  2. Implementar cada clase y definir sus atributos. Las clases a implementar serán: Concesionaria, Cliente, Venta y Vehículo.
  3. Crear los constructores para la clase Cliente, Vehículo y Venta.
    - Para cliente, recibirá por parámetro un nombre, un apellido y un contacto.
    - Para vehículo, recibirá por parámetro una marca, un modelo, un año de fabricación, un color y un kilometraje.
    - Para venta, recibirá por parámetro un cliente, un vehículo y un monto de venta.

4. Definir, en la clase Concesionaria, el método **registrarVenta(Vehículo vehículo, Cliente cliente, Double monto)**, que agrega al registro de ventas de la concesionaria una nueva venta de un vehículo, cliente por un monto determinado.
5. Definir una nueva clase llamada **Main** y crear un método main. Dentro del método main crear un auto, crear un cliente y registrar una venta de la Concesionaria de dicho vehículo y dicho cliente.

## Adicionales:

### Adicional 1: (Atletas Reloaded)

Extender el ejercicio de atletas para que:

1. Un atleta pueda competir en uno de los estadios de Río. Cada estadio tiene un conjunto de pruebas
2. Dado un atleta quiero saber como le fue en un estadio, es decir qué pruebas paso. El método recibe un atleta y devuelve una lista con las pruebas que pasó.
3. Dados dos atletas y una prueba, quiero saber quién la realiza mejor. Es decir, será aquel atleta que pueda hacerla y tenga mayor nivel atlético.
4. Dados dos atletas y un estadio determinar qué atleta se lleva la medalla de oro. Es decir, el que pueda realizar mejor la mayor cantidad de pruebas.