

PRÁCTICO 3: Intro. a la Programación Orientada a Objetos

Ejercicio 1:

Se desea desarrollar un sistema para un club deportivo el cual tiene deportistas asociados que pueden ser amateurs o profesionales.

De cada deportista se desea registrar su cédula, el id de socio (el mismo deberá ser único) su nombre completo, su fecha de nacimiento, y el total de horas de entrenamiento mensuales.

Además, de los Amateurs se debe conocer la antigüedad en años, y de los Profesionales, se deberá conocer el sueldo.

A) Modelar la siguiente realidad en un sistema orientado a objetos.

Cada clase deberá además tener un método `toString()` que despliegue los datos de cada deportista. En el caso de los deportistas profesionales, deberán tener un método `gananciaMensual()` que retorne la cantidad de dinero que le genere el deportista al club. Esto se calcula estimando que por cada hora de entrenamiento genera un ingreso de 45\$.

B) En la lógica de nuestra aplicación web, desarrolle un script que solicite el ingreso de 3 deportistas (usando `prompt()`). En cada repetición, se deberá consultar si el deportista ingresado es amateur o profesional. En cada caso, solicitar los datos pertinentes.

Se deberá validar que el total de deportistas ingresados no sean mas que tres, y que el id del socio no se pueda repetir. (Ir guardando cada objeto a un array)

Al final del ingreso de datos, se deberá mostrar por pantalla los siguientes datos:

B.A) El deportista con mas horas de entrenamiento

B.B) El amateur mas nuevo en el club

B.C) El profesional que mas ingresos genere, y el que genere menos ingresos.

Ejercicio 2:

Se desea desarrollar un pequeño sistema de gestión para una empresa de alquileres de vehículos. Los vehículos que alquilan pueden ser autos, motos, buses, bicicletas, o camionetas.

Se desea registrar el tipo de vehículo, el id de vehículo (debe ser único), la chapa (también única), el color, marca, modelo, y año de fabricación. También se deseará saber si el vehículo está alquilado, o está disponible para ser alquilado.

A) Modelar la siguiente realidad en un sistema orientado a objetos.

Cada objeto además de sus propiedades deberá tener un método `toString()` que despliegue toda su información, excepto la chapa del vehículo.

PRÁCTICO 3: Intro. a la Programación Orientada a Objetos

Además, deberán tener un método llamado *alquilerInternacional()* que en caso de ser un vehículo fabricado antes de 1980, no podrá salir del país, por ende la función retorna false. En caso contrario retornará true.

- B) Programar la lógica de nuestra aplicación, la cual deberá desplegar todos los datos de todos los vehículos. Dicha información se deberá mostrar en el documento html. Luego de desplegar la info (con datos de prueba creados desde el código de JavaScript), solicitar al usuario (con `prompt()`) si desea alquilar algún vehículo, ingresando su id. En caso de que el id sea válido y no sea de un vehículo alquilado, entonces realizar la función de alquilarlo (modificar la propiedad de alquilado al vehículo en cuestión). Luego de esto, se deberá actualizar el array o el lugar donde se hayan guardado todos los vehículos, y mostrar los nuevos datos actualizados en la consola.

Ejercicio 3:

Se desea desarrollar una pequeña aplicación web que gestione la cuenta de un Cliente. De dicho cliente, se desea registrar su nombre, apellido, y fecha de ingreso. La fecha de ingreso será la fecha actual del sistema desde el cual acceda a la transacción. Por otra parte, se desea registrar la información de las Cuentas, las cuales están asociadas a un Cliente. Además, de dichas cuentas se desea saber el saldo, y el tope. El tope será la cantidad máxima de saldo que podrá tener una cuenta, es decir, si el usuario desea ingresar una cantidad que sumada al saldo actual de la cuenta, supere el tope, entonces no se podrá realizar dicha transacción.

La aplicación deberá solicitar los datos del Cliente, y luego deberá preguntarle al mismo si desea ingresar o retirar dinero.

Se deberán realizar las operaciones pertinentes, y todas las validaciones necesarias. Por ejemplo, si el dinero a retirar supera el saldo, no se podría realizar el retiro.

En este ejercicio se plantea como objetivo secundario, lograr escribir un código limpio, y desarrollar una buena lógica. Se deberá trabajar bajo el paradigma de programación orientada a objetos.

Todo el flujo de datos (entrada/salida) deberá emplearse con las funciones `alert()` y `prompt()`