

Ejercicio 1:

Crea las clases Curso, Estudiante y Profesor para gestionar las asignaturas de un curso. Todos los atributos serán privados.

PROFESOR: Tendrá los datos nombre (String), asignatura impartida (String), DNI (String final), fecha de alta (String). Crea los métodos get y set para todos los atributos y un constructor con parámetros. Además, se creará un método mostrar datos que tendrá el formato: "El profesor de nombre --- con DNI --- imparte la asignatura --- desde *fecha de alta*."

ESTUDIANTE: Tendrá los datos nombre (String), edad (entero) y notas (un array de 10 double). Se deberán crear los métodos get y set para todas las variables y un constructor con parámetros. Además, se creará un método mostrar datos que tendrá el formato: "El alumno de nombre --- con edad --- y notas (se ponen todas las notas seguidas separadas por comas)."

CURSO: Tendrá dos arrays de profesores y alumnos (10 profesores y 20 alumnos como máximo). Ten en cuenta que el orden de las notas será el mismo orden en el que estén los profesores (las notas de Programación estarán todas en la primera posición del array de notas y en la primera posición del array de Profesores estará la información del profesor de Programación). Además, tendrá un String que nombrará al curso.

Tendrá los métodos:

- **Get/set** para los atributos y **constructor** con parámetros que recibirá el nombre del curso.
- **anyadirEstudiante:** Se aumentará el tamaño del array siempre que la longitud de este no sea 20, se creará un nuevo estudiante y se añadirá a esa nueva posición disponible en el array.
- **eliminarEstudiante:** Se le pasará el nombre de un alumno, lo buscará en el array, lo eliminará y reorganizará el array disminuyendo su longitud.
- **anyadirProfesor:** Se aumentará el tamaño del array siempre que la longitud de este no sea 10, se creará un nuevo profesor y se añadirá a esa nueva posición disponible en el array.
- **eliminarProfesor:** Se le pasará el nombre de un profesor, lo buscará en el array, lo eliminará y reorganizará el array disminuyendo su longitud.
- **calcularMediaNotas:** Se mostrará la nota media de cada una de las asignaturas con el formato:

NombreAsignatura : mediaNotas (EJ. Programación : 6,5). La nota media deberá obtenerse calculando la media de la nota de todos los alumnos en esa asignatura.

- **buscarEstudiante:** Buscará el nombre del estudiante en el array y si lo encuentra mostrará sus notas con el formato:
Listado de notas del alumno *nombreEstudiante*:
- *NombreAsignatura : NotaAsignatura* (EJ. Programación : 6,5)
- **mostrarListado:** Recorre ambos arrays y muestra la información de, primero, los alumnos y, después, los profesores.

Ejercicio 2:

Crea las clases Pasajero, Tripulante y Vuelo para gestionar los diferentes vuelos de una compañía de viajes. Todos los atributos serán privados.

PASAJERO: Tendrá los datos nombre (String), apellidos (String), Pasaporte (String), Asiento (int), conVentana? (boolean). Crea los métodos get y set para todos los atributos y un constructor con parámetros. Además, se creará un método mostrar datos que tendrá el formato: "El asiento *asiento con/sin* ventana (en función de conVentana) está ocupado por el pasajero de nombre ---, apellido --- y pasaporte ---.

TRIPULANTE: Tendrá los datos nombre (String) y rol (piloto, servicio o cargamento listado en un enumerado). Se deberán crear los métodos get y set para todas las variables y un constructor con parámetros. Además, se creará un método mostrar datos que tendrá el formato: "El tripulante *nombre* está a bordo con el rol ---.

VUELO: Tendrá dos arrays de tripulantes y pasajeros (5 tripulantes y 20 pasajeros como máximo).

Tendrá los métodos:

- **Get/set** para los atributos y **constructor** sin parámetros.
- **listaTripulacionPorRol:** Listar el nombre de todos los tripulantes que tengan el rol pasado por parámetro.
- **anyadirTripulante:** Se aumentará el tamaño del array siempre que la longitud de este no sea 5, se creará un nuevo tripulante y se añadirá a esa nueva posición disponible en el array.
- **eliminarTripulante:** Se le pasará el nombre de un tripulante, lo buscará en el array, lo eliminará y reorganizará el array disminuyendo su longitud.
- **anyadirPasajero:** Se aumentará el tamaño del array siempre que la longitud de este no sea 20, se creará un nuevo pasajero y se añadirá a esa nueva posición disponible en el array.
- **eliminarPasajero:** Se le pasará el nombre de un pasajero, lo buscará en el array, lo eliminará y reorganizará el array disminuyendo su longitud.

- **obtenerPasajeroPorAsiento:** Se pasará un asiento y mostrará el nombre, el apellido y el pasaporte.

Ejercicio 3:

Crea las clases Banco, Cuenta y Cliente para gestionar las acciones que un usuario puede realizar con su dinero. Todos los atributos serán privados.

CLIENTE: Tendrá los datos nombre (String), apellidos (String), DNI (String final), Usuario (String), Contraseña (String). Crea los métodos get y set para todos los atributos y dos constructores, uno con y otro sin parámetros.

CUENTA: Tendrá los datos número de cuenta que será un número de 8 cifras, el saldo disponible que será un double y un cliente asociado (Cliente). Se deberán crear los métodos get y set para todas las variables y un constructor con parámetros (para crear el cliente se enviarán sus valores de atributos e internamente se llamará al constructor de cliente).

BANCO: Tendrá un array de cuentas, supondremos que como máximo podremos almacenar 10 cuentas al mismo tiempo, un booleano sesionIniciada que comenzará en false y un entero que almacenará la posición sobre la que estaremos trabajando y comenzará en 0.

Tendrá los métodos:

- **Get/set** para los atributos y **constructor** sin parámetros.
- **Iniciar Sesión:** Será IMPRESCINDIBLE que la sesión esté iniciada para realizar cualquiera de las otras acciones. Este método recibirá un usuario y contraseña y comprobará que alguno de sus usuarios tenga esos datos (recorre el array de cuentas y accede a su cliente asociado). Si no encuentra ningún usuario muestra por pantalla “Usuario y contraseña incorrectos”, si encuentra el usuario, pero la contraseña es incorrecta muestra “¿Ha olvidado su contraseña?”, si el inicio de sesión es correcto muestra “Sesión iniciada correctamente” y modifica la variable sesionIniciada y posicion.
- **Revisar estado de cuenta:** Si se ha iniciado sesión entonces se mostrará una lista con los movimientos realizados, empezando por el más reciente (el último del array).

- **Revisar datos de cliente:** Se mostrarán los datos del cliente asociado a esa cuenta con el siguiente formato:
Nombre y Apellido: ---, DNI: ---
Usuario: ---, Contraseña: ---
Numero de cuenta: ---, Saldo disponible: ---
- **Retirar dinero:** Se restará la cantidad pasada por parámetro (debe ser positiva) si esta es inferior al saldo de la cuenta. Si es superior se mostrará “Saldo insuficiente”, si se puede realizar la operación se mostrará “Solicitud realizada”
- **Ingresar dinero:** Se añadirá al saldo disponible la cantidad pasada por parámetro (debe ser positiva).
- **Cerrar Sesión:** Se colocará sesionIniciada a false y posicion a -1. Se mostrará “Se ha cerrado la sesion”
- **Crear cuenta:** Se pedirán por teclado (se deberá pasar el scanner por parámetro) todos los atributos de las clases Cuenta y Cliente y almacenar esa nueva cuenta en el array.