

Imperativo – Ejercicio 1

Un sitio de estadísticas deportivas mantiene la información de jugadores del fútbol argentino. Para ello, se dispone de los datos de los partidos jugados en el año 2022. De cada partido se conoce el estadio donde se jugó, los nombres de los 2 equipos que se enfrentaron, la fecha (ej. '2022/03/28') y la información de los jugadores que participaron del encuentro. De cada jugador se conoce su nombre y apellido, su DNI, posición (arquero, defensa, mediocampo o delantero) y un puntaje otorgado por un crítico deportivo (1..10). Esta información no tiene ningún orden específico.

Se pide:

a) Crear una nueva estructura que almacene para cada jugador, su DNI, su nombre y apellido, posición, y la fecha junto al puntaje obtenido para cada partido del que participó. Esta estructura debe estar ordenada por DNI, y debe ser eficiente para la búsqueda por dicho criterio.

Al finalizar el procesamiento de a), elija las estructuras adecuadas para:

b) Implementar un módulo que **informe por pantalla** los datos de cada jugador y su puntaje total acumulado entre todos los partidos. El listado debe estar ordenado por DNI de manera descendente.

c) Implementar un módulo que **retorne** la cantidad de jugadores cuyo DNI se encuentre entre 30.000.000 y 40.000.000.

d) Implementar un módulo que, dado una posición recibida por parámetro, **retorne** la cantidad de jugadores que hay en el sistema con dicha característica.

e) Implementar un programa principal que utilice los módulos implementados e informe lo que corresponda.

Código fuente disponible en Ejercicio1.pas

Recordar!
Modularizar y resolver
eficientemente

Imperativo – Ejercicio 2

Una aerolínea quiere premiar a sus mejores clientes y para ello, lee la información de los pasajes que ha vendido. De cada venta se conoce un código de vuelo (alfanumérico, ej: AR1620), cantidad de millas recorridas, DNI del pasajero, su nombre y apellido y la clase en que solicitó el boleto. Esta información se lee ordenada por código de vuelo.

Se necesita saber qué cantidad de puntos acumularon los pasajeros de la siguiente manera: cada pasajero que haya viajado en clase ejecutiva suma 100 puntos por milla de vuelo, mientras que aquellos que viajaron en clase turista suman 25 puntos por milla.

Por lo que se pide:

- a) Generar una estructura que almacene para cada pasajero su DNI, nombre y apellido y los puntajes acumulados por cada código de vuelo, de manera tal que resulte **eficiente la búsqueda por DNI del pasajero**.

Al finalizar el procesamiento de a):

- b) Escriba un módulo que **devuelva** el mayor puntaje total (sumando todos sus vuelos) para dar el premio al mejor cliente.
- c) Escriba un módulo que **imprima** el mayor puntaje (entre sus vuelos) de los pasajeros cuyo DNI se encuentre entre 40.000.000 y 50.000.000.
- d) Realizar un programa que simule el llamado a los módulos b) y c), e informe lo devuelto por c).

Recordar!
Modularizar y resolver
eficientemente

Imperativo – Ejercicio 3

Una empresa de telecomunicaciones procesa los datos de varios pedidos de conexión que llegan telefónicamente. Se dispone de una lista con los datos cargados y listos para ser procesados. Cada pedido posee un código único de seguimiento, fecha y hora del turno de conexión, DNI del cliente, código de área, domicilio del cliente y su número telefónico.

Posteriormente al ingreso y almacenamiento de la información, se pide:

a) Crear una nueva estructura que almacene para cada código de área, el total de pedidos de dicha área y los DNI de los clientes que solicitaron el servicio. Esta estructura debe estar ordenada por código de área y debe ser eficiente para la búsqueda por dicho criterio.

Al finalizar el procesamiento de a), elija las estructuras adecuadas para:

b) Implementar un módulo que retorne el código de área con menor cantidad de pedidos de servicio.

c) Implementar un módulo que imprima los DNI de clientes cuyo código de área se encuentre entre 1000 y 2000, ordenados por código de área.

d) Implementar un módulo que, dado un código de área, imprima los DNI de los clientes que solicitan conexión.

e) Implementar un programa principal que utilice los módulos implementados.

Recordar!
Modularizar y resolver
eficientemente

Código fuente disponible en Ejercicio3.pas