

Nombre y apellidos:

20/5/2024

### 3 EVAL PROGRAMACIÓN

Tenemos una base de datos para gestionar las reservas de clases dirigidas en un gimnasio.

Guardamos en tres tablas los datos necesarios: la información de las clases que se imparten en el gimnasio, los usuarios que pueden recibir dichas clases y las reservas de las clases que aún no se han realizado.

Vamos a llevar en memoria una lista de espera (usad el Collection más adecuado) para almacenar las reservas de las clases que a la hora de hacer la reserva están llenas.

Se pide codificar el siguiente supuesto (no es un menú):

1. Los usuarios pueden tener tarifa familiar o normal.

Si es normal el precio es de 45 euros y puede tener además un descuento.

Si tiene tarifa familiar el precio es de 30 euros si son 3 miembros, 20 si son 4 y 12 si son más de 4.

Acceded a la base de datos y cread una ArrayList con la información de todos los usuarios usando herencia. (1 pto)

Recorred el ArrayList y calculad lo que tiene que pagar cada usuario usando polimorfismo.

Mantened este ArraList el resto del programa. (1 ptos)

2. Se han guardado en el archivo de texto reservas.txt las últimas reservas solicitadas ya que la base de datos no funcionaba.

El archivo contiene separado por comas la siguiente información:

usuario, clase dirigida, fecha de la reserva, y un indicador de si es una reserva preferente.

Si no existe el usuario o la clase dirigida se lanza una excepción propia, se muestra un error y se procede con la siguiente reserva. (1,25 ptos)

Si hay plazas para la clase a reservar se inserta la reserva en la tabla de reservas (observar que el número de reserva es autoincremental). (1 pto)

Si, por el contrario, la clase ya está llena se almacena la reserva en la lista de espera.  
(0,5 ptos)

Se actualiza el número de plazas de dicha clase en la base de datos. (0,5 ptos)

3. Ordenar la lista de espera creada en el punto anterior (usando interfaces) primero las reservas más antiguas y luego las más recientes, y a misma fecha, las que tienen preferencia delante. (1,25 ptos)

4. Cancelar reservas confirmadas.

Se anota un usuario, se valida que el código tenga 3 letras en mayúsculas y 2 ó 3 números , y se muestran todas sus reservas de la bbdd. (1 pto)

Se elige la reserva a cancelar y se elimina de la base de datos.

Si fuera el caso, se da de alta en la BBDD una reserva de la lista de espera, que se elimina de ésta, actualizando además el número de plazas ocupadas de la clase. (1,25 ptos)

Repetid el proceso hasta que se anote un usuario que no existe o que no tiene reservas.  
(0,75 ptos)

5. Al crear los usuarios en el punto uno, añadir todas las clases que dicho usuario tiene reservadas o en las que está en lista de espera.

Actualizar dicha información en el resto de puntos siempre que sea necesario. Mostrar por pantalla la información de todos los usuarios al llegar a este punto. (0,5 ptos)

Se tendrá en cuenta el uso de métodos para no repetir código y la creación apropiada de las clases y estructuras de datos.

