

En el gimnasio "NoHayForma" se desea implementar un programa para poder gestionar los socios y la deuda de los mismos.

Del gimnasio se guardará su nombre, su cantidad de socios y los datos de los mismos.

De cada socio se necesita guardar su número de dni, su nombre, su categoría (de 1 a 3), su deuda y su estado (N=normal,M=moroso)

Se necesitarán los siguientes métodos:

- **inscribirSocio:** agregará un socio a la lista de socios del gimnasio. Recibirá los datos del mismo y verificará si el socio no fue ingresado anteriormente. En este caso mostrará un mensaje "el socio ya existe" o en caso contrario "socio agregado con éxito" y sumará uno a la cantidad de socios del gimnasio. Cada socio nuevo que se agregue tendrá inicialmente estado Normal ("N").
- **quitarSocio:** dará de baja un socio. Recibirá el número de dni del mismo y si el mismo existe y no tiene deuda lo eliminará de la lista de socios. En este caso mostrará el mensaje "socio dado de baja" o en caso contrario "el socio tiene deuda o no existe. No puede borrarlo". Si pudo hacerse la baja se deberá restar uno de la cantidad de socios del gimnasio.
- **listarDeudoresCategoría:** muestra los socios cuya deuda supere un monto y sean de una categoría determinada. Tanto el monto como la categoría serán ingresados por teclado y el método los recibirá como parámetros. De los socios que aparezcan en este listado se deben mostrar todos sus datos.
- **ponerMoroso:** recibe un dni de un socio y si el mismo existe y tiene deuda cambiará su estado a "M". También deberá mostrar un mensaje indicando si pudo hacerlo o no.
- **buscarSocio:** busca si existe un socio. Recibe como parámetro su dni.

Los métodos deberán crearse en la clase que corresponda.

Se deberán probar todos los métodos desde la clase Parcial2. En la misma se ingresarán datos de socios hasta que el operador ingrese 0 como número de dni y también deberán probarse una vez los métodos quitarSocio, listarDeudoresCategoría y ponerMoroso.

Para aprobar el examen se debe:

- Desarrollar una lógica que obtenga lo pedido por el enunciado.
- Correcto uso de constructores.
- Correcto diseño de clases y métodos.
- Uso de los ciclos que corresponda.
- Validaciones requeridas.
- Uso de setters y getters necesarios.