UTNMDP
Regional Mar del Plata

Guia #4
Collections

1. Supongamos que estás desarrollando un programa para gestionar una tienda de libros. Cada libro tiene un título, un autor, un precio, y el año de publicación. El programa debe permitir realizar las siguientes operaciones:
    - Agregar un nuevo libro al inventario.
    - Eliminar un libro del inventario.
    - Mostrar todos los libros en el inventario.
    - Buscar un libro por su título.
    - Actualizar el precio de un libro.
    - Calcular el precio total de todos los libros en el inventario.
    - Contar el número total de libros en el inventario.
    - Encontrar el libro más caro y el más barato en el inventario.

    Pensar que estructuras de datos vista (ArrayList, LinkedList, Stack, HashMap, etc.) nos conviene para implementar las operaciones anteriores.

    Consideraciones adicionales:

    - Crea una clase Libro que tenga atributos para el título, autor, precio y año de publicación.

    - Asegúrate de agregar validaciones de entrada para garantizar que los datos ingresados por el usuario sean válidos.

2. Haz una clase llamada Persona que siga las siguientes condiciones:
    - Sus atributos son: nombre, edad, DNI, sexo (H hombre, M mujer), peso y altura. No queremos que se accedan directamente a ellos. Piensa que modificador de acceso es el más adecuado, también su tipo. Si quieres añadir algún atributo puedes hacerlo.
    - Por defecto, todos los atributos menos el DNI serán valores por defecto según su tipo (0 números, cadena vacía para String, etc.). Sexo sera hombre por defecto, usa una constante para ello.
    - Se implantaran varios constructores:
        - Un constructor por defecto.
        - Un constructor con el nombre, edad y sexo, el resto por defecto.
        - Un constructor con todos los atributos como parámetro.

- Los métodos que se implementarán son:
  - ➢ calcularIMC(): calculará si la persona está en su peso ideal (peso en kg/(altura^2 en m)), si esta fórmula devuelve un valor menor que 20, la función devuelve un -1, si devuelve un número entre 20 y 25 (incluidos), significa que está por debajo de su peso ideal la función devuelve un 0 y si devuelve un valor mayor que 25 significa que tiene sobrepeso, la función devuelve un 1. Te recomiendo que uses constantes para devolver estos valores.
  - ➢ esMayorDeEdad(): indica si es mayor de edad, devuelve un booleano.
  - ➢ comprobarSexo(char sexo): comprueba que el sexo introducido es correcto. Si no es correcto, será H. No será visible al exterior.
  - ➢ toString(): devuelve toda la información del objeto.
  - ➢ generaDNI(): genera un número aleatorio de 8 cifras, genera a partir de este su número su letra correspondiente. Este método será invocado cuando se construya el objeto. Puedes dividir el método para que te sea más fácil. No será visible al exterior.
  - ➢ Métodos set de cada parámetro, excepto de DNI.

Ahora, crea una clase ejecutable que haga lo siguiente:

- Pide por teclado el nombre, la edad, sexo, peso y altura.
- Crea 3 objetos de la clase anterior, el primer objeto obtendrá las anteriores variables pedidas por teclado, el segundo objeto obtendrá todos los anteriores menos el peso y la altura y el último por defecto, para este último utiliza los métodos set para darle a los atributos un valor.
- Crea una Collection donde se guardarán los objetos anteriormente creados. Dichos objetos deben estar asociados a su DNI. Pensar que Collection es la más adecuada.
- Para cada objeto, se deberá comprobar si está en su peso ideal, tiene sobrepeso o está por debajo de su peso ideal con un mensaje.

3. Se nos pide armar un programa similar a Spotify que nos permitirá administrar nuestras listas de reproducción. Para ello deberemos generar una clase Canción la cual posee un nombre, la duración de la canción, género (el cual solo podrá ser ROCK, TRAP, JAZZ, HIP HOP, POP, METAL y CLASICA), el Álbum al que pertenece y a veces un artista invitado si es que tiene. Álbum posee el año en que fue publicado, un título y conoce su Artista. Artista a su vez posee un nombre, su edad y la nacionalidad.

UTNMDP
Regional Mar del Plata

1) Armar un ArrayList de canciones.

2) Crear una interfaz llamada Reproducción, que tendrá los siguientes métodos:

- Reproducir: Debe mostrar la canción en reproducción, junto con el nombre del álbum, género y nombre del artista junto con el artista invitado si es que hay.
- AñadirCancion: recibe una canción por parámetro y la guarda en la lista en reproducción.
- EliminarCancion: mostrará las canciones de la lista en reproducción y permitirá elegir una para eliminar.
- VerMiLista: Nos permite ver nuestra lista de reproducción.

3) Crear una clase llamada ListaBasica que implementa la interfaz del punto 2. Posee como atributo nombre, y un objeto de tipo Stack llamado miLista donde se guardarán las canciones. Los métodos eliminarCancion y cambiarCancion no deben permitirle al usuario eliminar o cambiar de canción, sino que deben decirle el siguiente mensaje: "Para acceder a estas opciones, compre el paquete PREMIUM". El método Reproducir solo reproducirá la primera canción del stack y la enviará al final del mismo.

4) Crear una clase llamada ListaPremium que implementa la interfaz del punto 2. Premium posee como atributo nombre y una LinkedList llamada miLista donde se guardarán las canciones. Esta clase debe poder implementar todos los métodos de la interfaz. El método Reproducir mostrará la lista de reproducción donde deberá elegir la canción a reproducir.

5) Crear un menú para interactuar con nuestra lista de reproducción permitiéndonos utilizar la funcionalidad de la interfaz (reproducir, añadir, eliminar, listar). Estando en reproducción, debería apretar la letra "p" para pausar la canción y regresar al menú principal.