

Java Base y estructuras de datos

1. Crea una clase Contador (sí, contador de programación) con métodos que permitan incrementar y decrementar su valor. La clase contendrá un constructor por defecto, un constructor con parámetros, un constructor copia y los setters y getters (métodos de acceso) que corresponda.
2. Crea una clase CuentaCorriente, con los métodos: ingreso, egreso, reintegro y transferencia. La clase contendrá un constructor por defecto, uno con parámetros y otro que recibirá una CuentaCorriente de la cual copiará todos sus datos. Además se deben desarrollar los métodos de acceso (para cada variable de instancia de la clase, desarrollar un método set y otro get. Por ejemplo: si la clase tiene una variable double saldo, sus métodos de acceso serán: double getSaldo() y void setSaldo(double s)
3. Crea una clase Fecha. La clase contendrá además de los constructores que consideres adecuados, métodos de acceso y el método toString, tal como lo explicamos en el ejercicio anterior, un método para comprobar si la fecha es correcta y otro para sumarle un día al valor actual de la fecha. Se debe investigar y utilizar la clase GregorianCalendar para implementar los métodos y constructores de Fecha
4. Crea una clase Fraccion con métodos necesarios para sumar, restar, multiplicar y dividir fracciones. Todos los métodos deben estar sobrecargados de modo que también puedan usarse para operar entre fracciones y números enteros (por ejemplo: $\frac{3}{4}+2$ o $\frac{5}{8}*4$)
5. Crea una clase Libro con los métodos: préstamo, devolución y toString, cuyo prototipo debe ser: public String toString(). Este método debe retornar una cadena que represente al objeto. Por ejemplo: si la clase tiene los atributos: título, isbn y autor, una cadena que represente a un libro podría ser: "Harry Potter, 9780545582889, Rowling, J. K.". La clase contendrá un constructor por defecto, un constructor con parámetros y los métodos de acceso.

Ejercicio de Mapas

Necesitamos una aplicación para una tienda mediante la que queremos almacenar los distintos productos que venderemos y el precio que tendrán. Y se quiere que tenga las funciones básicas, introducir un elemento, modificar su precio, eliminar un producto y mostrar los productos que tenemos con su precio. (todo esto solicitando los datos y las opciones por consola)

(Ejemplo de cómo se vería la pantalla inicial)

```
Introduce el numero de la opción que quieras:  
1.- Introducir producto  
2.- Modificar precio  
3.- Mostrar todos los productos  
4.- Eliminar producto  
5.- Salir
```

Ejercicio List

Programa que lea una serie de valores numéricos enteros desde el teclado y los guarde en un ArrayList de tipo Integer. La lectura de números enteros termina cuando se introduce el valor -99. Este valor no se guarda en el ArrayList. A continuación el programa mostrará por pantalla el número de valores que se han leído, su suma y su media. Por último se mostrarán todos los valores leídos, indicando cuántos de ellos son mayores que la media.

POO

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 acceda directamente a ellos. Piensa que el 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(): calcular si la persona está en su peso ideal (peso en kg/(altura² 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 esta 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 sera 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.

Para cada objeto, deberá comprobar si está en su peso ideal, tiene sobrepeso o por debajo de su peso ideal con un mensaje. Indicar para cada objeto si es mayor de edad.

Por último, mostrar la información de cada objeto.

SQL

Consultas

SELECT

1. Mostrar todos los registros de la tabla de movies.
2. Mostrar el nombre, apellido y rating de todos los actores.
3. Mostrar el título de todas las series.

WHERE Y ORDER BY

4. Mostrar el nombre y apellido de los actores cuyo rating sea mayor a 7,5.
5. Mostrar el título de las películas, el rating y los premios de las películas con un rating mayor a 7,5 y con más de dos premios.
6. Mostrar el título de las películas y el rating ordenadas por rating en forma ascendente.

BETWEEN y LIKE

7. Mostrar el título y rating de todas las películas cuyo título incluya Toy Story.
8. Mostrar a todos los actores cuyos nombres empiecen con Sam.
9. Mostrar el título de las películas que salieron entre el '2004-01-01' y '2008-12-31'.

1. Utilizando la base de datos de movies, queremos conocer, por un lado, los títulos y el nombre del género de todas las series de la base de datos.
2. Por otro lado, necesitamos listar los títulos de los episodios junto con el nombre y apellido de los actores que trabajan en cada uno de ellos.
3. Para nuestro próximo desafío, necesitamos obtener a todos los actores o actrices (mostrar nombre y apellido) que su película favorita sea cualquiera de la saga de La Guerra de las Galaxias.
4. Para nuestro próximo desafío, necesitamos obtener a todos los actores o actrices (mostrar nombre y apellido) que han trabajado en cualquier película de la saga de La Guerra de las Galaxias.
5. Crear un listado a partir de la tabla de películas, mostrar un reporte de la cantidad de películas por nombre de género.

API

si no quieren seguir la entidad que trabajamos en clase hagan una propia

1. CRUD con mysql
2. CRUD con mongoDB
3. Test para los servicios de cualquiera de los de sql