

Ejercicios Orientación a Objetos para Practicar

EJERCICIO1

Enunciado: clase Automóvil

Se requiere un programa que modele el concepto de un automóvil. Un automóvil tiene **los siguientes atributos**:

- **Marca:** el nombre del fabricante.
- **Modelo:** año de fabricación.
- **Motor:** volumen en litros del cilindraje del motor de un automóvil.
- **Tipo de combustible:** String con los posibles valores de gasolina, bioetanol, diésel, biodiésel, gas natural.
- **Tipo de automóvil:** String con los posibles valores de carro de ciudad, subcompacto, compacto, familiar, ejecutivo, SUV.
- **Número de puertas:** cantidad de puertas.
- **Cantidad de asientos:** número de asientos disponibles que tiene el vehículo.
- **Velocidad máxima:** velocidad máxima sostenida por el vehículo en km/h.
- **Color:** valor enumerado con los posibles valores de blanco, negro, rojo, naranja, amarillo, verde, azul, violeta.
- **Velocidad actual:** velocidad del vehículo en un momento dado.

La clase debe incluir los siguientes métodos:

- Un constructor para la clase Automóvil donde se le pasen como parámetros los valores de sus atributos.
- Métodos get y set para la clase Automóvil.
- Métodos para acelerar una cierta velocidad, desacelerar y frenar (colocar la velocidad actual en cero). Es importante tener en cuenta que no se debe acelerar más allá de la velocidad máxima permitida para el automóvil. De igual manera, tampoco es posible desacelerar a una velocidad negativa.
- Si se cumplen estos casos, se debe mostrar por pantalla los mensajes correspondientes.
- Un método para calcular el tiempo estimado de llegada, utilizando como parámetro la distancia a recorrer en kilómetros. El tiempo estimado se calcula como el cociente entre la distancia a recorrer y la velocidad actual.
- Un método para mostrar los valores de los atributos de un Automóvil en pantalla.

Test de la Clase Automóvil

- Un método main donde se deben crear un automóvil, colocar su velocidad actual en 100 km/h, aumentar su velocidad en 20 km/h, luego decrementar su velocidad en 50 km/h, y después frenar. Con cada cambio de velocidad, se debe mostrar en pantalla la velocidad actual.

Ejercicios Orientación a Objetos para Practicar

EJERCICIO 2.-

Enunciado: clase Planeta

Se requiere un programa que modele el concepto de un planeta del sistema solar. Un planeta tiene los siguientes atributos:

- Un **nombre** de tipo String con valor inicial de null.
- **Cantidad de satélites** de tipo int con valor inicial de cero.
- **Masa** en kilogramos de tipo double con valor inicial de cero.
- **Volumen** en kilómetros cúbicos de tipo double con valor inicial de cero.
- **Diámetro** en kilómetros de tipo int con valor inicial de cero.
- **Distancia media al Sol** en millones de kilómetros, de tipo int con valor inicial de cero.
- **Tipo de planeta** de acuerdo con su tamaño, de tipo enumerado con los siguientes valores posibles: GASEOSO, TERRESTRE y ENANO.
- **Observable a simple vista**, de tipo booleano con valor inicial false.

La clase debe incluir los siguientes métodos:

- La clase debe tener un constructor que inicialice los valores de sus respectivos atributos.
- Definir un método que imprima en pantalla los valores de los atributos de un planeta.
- Calcular la densidad un planeta, como el cociente entre su masa y su volumen.
- Determinar si un planeta del sistema solar se considera exterior. Un planeta exterior está situado más allá del cinturón de asteroides. El cinturón de asteroides se encuentra entre 2.1 y 3.4 UA. Una unidad astronómica (UA) es la distancia entre la Tierra y el Sol= 149 597 870 Km.

Test Planeta

En un método main se deben crear dos planetas y mostrar los valores de sus atributos en pantalla. Además, se debe imprimir la densidad de cada planeta y si el planeta es un planeta exterior del sistema solar.

Ejercicios Orientación a Objetos para Practicar

EJERCICIO 3

Enunciado: clase Película

Realizar un programa en Java que defina una clase Película con los siguientes atributos privados:

- Nombre: es el nombre de la película y es de tipo String.
- Director: representa el nombre del director de la película y es de tipo String.
- Género: es el género de la película, un tipo enumerado con los siguientes valores: ACCIÓN, COMEDIA, DRAMA y SUSPENSO.
- Duración: duración de la película en minutos, esta es de tipo int.
- Año: representa el año de realización de la película.
- Calificación: es la valoración de la película por parte de sus usuarios y es de tipo double.

Se debe definir un constructor público para la clase, que recibe como parámetros los valores de todos sus atributos. También se deben definir los siguientes métodos:

- Métodos get y set para cada atributo y con los derechos de acceso privados para los set y públicos para los get.
- Un método imprimir público que muestre en pantalla los valores de los atributos.
- Un método boolean **esPelículaEpica()**, el cual devuelve un valor verdadero si la duración de la película es mayor o igual a tres horas, en caso contrario devuelve falso.
- Un método String **calcularValoración()**, el cual según la tabla 2.5 devuelve una valoración subjetiva.

Tabla 2.5. Valoración de las películas

Calificación	Valoración	Calificación	Valoración
[0, 2]	Muy mala	(7, 8]	Buena
(2, 5]	Mala	(8, 10]	Excelente
(5, 7]	Regular		

- El método **boolean esSimilar(Película otra)** compara la película actual con otra película que se le pasa como parámetro. Si ambas películas son del mismo género y tienen la misma valoración, devuelve verdadero; en caso contrario, devuelve falso.

Tabla 2.6. Objetos películas

Objeto	Nombre	Director	Género	Duración	Año	Calif.
película1	Gandhi	Richard Attenborough	DRAMA	191	1982	8.1
película2	Thor	Kenneth Branagh	ACCIÓN	115	2011	7.0

Probar con una Clase de testing.

- Un método main que construya dos películas; determinar si son películas épicas; calcular su respectiva valoración y determinar si son similares.

Ejercicios Orientación a Objetos para Practicar

EJERCICIO 4

Hacer una Clase con métodos estáticos para esta funcionalidad

Enunciado: clase ConversorMetros

Realizar un programa en Java que permita realizar las siguientes conversiones de unidades de longitud:

- Metros a centímetros.
- Metros a milímetros.
- Metros a pulgadas.
- Metros a pies.
- Metros a yardas.

Hacer clases similares para realizar conversiones de unidades de medición como:

- Medidas de superficie o área: convertir áreas (1 área= 100 m²) a: hectáreas (1 hectárea= 10 000 m²); kilómetros cuadrados (1 kilómetro cuadrado= 1 000 000 m²); fanegas (1 fanega =6460 m²) y acres (1 acre= 4046.85 m²).
- Medidas de volumen: convertir litros a: galones (1 galón=4,41 litros); pintas (1 pinta= 0.46 litros); barriles (1 barril= 158.99 litros), metros cúbicos (1 m³= 1000 litros) y hectolitros (1 hectolitro= 100 litros).

Ejercicios Orientación a Objetos para Practicar

EJERCICIO 5.

Enunciado: clase Cadena

Se desea construir un programa que dado un *String* con valor inicial de: "Programación Orientada a Objetos ", realice las siguientes operaciones:

1. Obtener la longitud de dicho *String*.
2. Eliminar los espacios en blanco del *String* obtenido en el paso anterior.
3. Pasar todos los caracteres del *String* (obtenido en el paso anterior) a mayúsculas.
4. Concatenar al *String* (obtenido en el paso anterior) el *String* "12345".
5. Extraer del *String* (obtenido en el paso anterior), un *Substring* desde la posición 10 al 15.
6. Reemplazar en el *String* (obtenido en el paso anterior) el carácter "o" por "O".
7. Comparar el *String* (obtenido en el paso anterior) con el *String* "Programación".

Después de cada paso, se debe mostrar el resultado en pantalla.