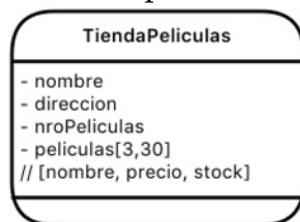


GUIA EJERCICIOS LAB - 121

INTRODUCCIÓN POO

1.- Sea una tienda de películas representada por la clase:



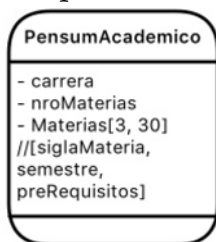
- Mostrar las películas que empiezan con la letra X.
- Sean 2 objetos **TiendaPeliculas** mostrar las películas que tienen en común.
- Insertar k nuevas películas en la Tienda.
- Eliminar la película de nombre X.

2.- Dado el siguiente diagrama de clase **Persona** implementar los métodos `getter` and `setter` para los siguientes atributos y resolver los siguientes incisos:



- Crear o Instanciar 3 personas distintas.
- Comparar si 2 personas tienen la misma edad con métodos `getter` and `setter`
- Comparar si 2 personas tienen la misma edad con un método comparación de la clase
- Mostrar los datos de la persona si tiene el `ci` igual X.
- Buscar la persona con mayor edad entre las personas creando un método de la clase **persona**

3.- Implementar el “**Pensum Académico**” que tiene el siguiente diagrama de clase:

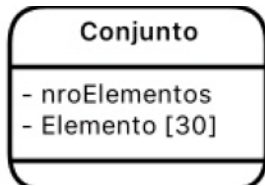


- Comparar si las materias de 2 objetos **Pensum Académico** son iguales
- Buscar si existe la materia X
- Eliminar la materia los pre - requisitos de la materia X
- Crear un método que permite verificar si los 2 objetos son iguales (2 objetos son iguales cuando todos sus atributos son iguales)

4.- Crear una clase **Punto** que modele un punto en un espacio bidimensional. Tendrá 2 atributos, `X` y `Y`, que guardan las coordenadas. Habrá un constructor sin parámetros que creen un punto en (0, 0) y otro al que se le pueden pasar las coordenadas del punto. También habrá método para obtener las coordenadas y imprimir en el formato (X, Y). Otro método que permita calcular la distancia que existe entre 2 puntos y otro que permita mostrar el ángulo entre los 2 puntos.

SOBRECARGA

1.- Sea la clase conjunto:



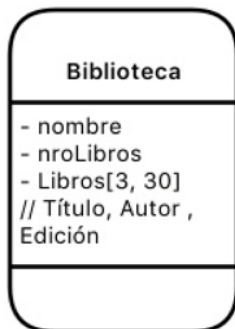
- Instanciar la clase conjunto de 3 diferentes formas.
- Programe la Union, Intersección y diferencia de conjuntos (sobrecargar el operador de su elección)
- Sobrecargar el método verificar
 - Primero verificar si existen datos repetidos en el conjunto
 - Segundo verificar si en el conjunto esta los datos X

2.- Dada la clase **Producto** se pide implementar y sobrecargar los siguientes operadores



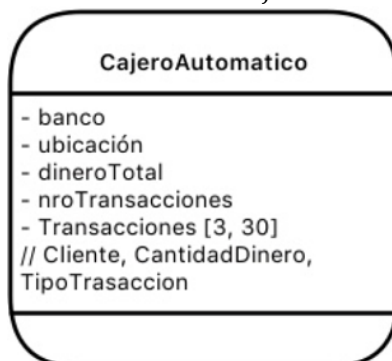
- `++` para leer atributos
- `-` para reducir el precio a la mitad
- `!` para mostrar
- `+` para subir el precio según X
- `&` cambiar el nombre del producto

3.- Dada la clase **Biblioteca** se pide instanciar 2 objetos y realizar las siguientes operaciones



- `+` para trasladar los libros de una biblioteca a otra
- `==` para ver si 2 bibliotecas tienen la misma cantidad de libros del autor X y `!=` para ver si en 2 bibliotecas no tienen ningún libro en común.
- Sobrecargar el método Eliminar
 - Primero eliminar los libros de titulo X.
 - Segundo eliminar los libros del autor X y Edición Y.
- Sobrecargar el método Ordenar
 - Primero ordenar los libros por el Titulo
 - Segundo ordenar los libros por autor.

4.- Dada la clase **CajeroAutomatico**.



- `+` para realizar un deposito de una cantidad X, leer dentro sobrecargar el cliente
- `-` para realizar un retiro de una cantidad X, leer dentro sobrecargar el cliente
- `!` mostrar el informe de depósitos y retiros del cajero
- `*` mostrar los depósitos y retiros del cliente X
- Sobrecargar el método Buscar
 - Buscar todas las transacciones del cliente X
 - Buscar los clientes con mayor cantidad de dinero depositado y buscar al cliente con menor cantidad de retiro de dinero.