1.- Implementa las siguientes clases:

* clase **Nota**. Una nota contiene un identificador numérico y una línea de texto. Define constructor, accedentes, mutadores y toString.
* clase **NotaAlarma**. Una nota que además contiene la hora en la que sonará la alarma. Define constructor, accedentes, mutadores y toString.
* clase **BlocNotas** que modela un bloc de notas en el que se pueden introducir notas, listar todas las notas, eliminar una nota mediante su posición en el bloc de notas o saber cuántas notas contiene el bloc de notas. Utiliza un Array.
* clase **Prueba** que cree un bloc de Notas de ejemplo y pruebe las operaciones que soporta. .

2.- Una persona desea tener una tienda virtual para vender libros y películas en formato *blu-ray*. Para esto es necesario, primeramente, representar con un enfoque orientado a objetos, los tipos de productos que se van a vender. Inicialmente, se están conside­rando las dos clases siguientes:

* Libro, que tiene como atributos el autor y el titulo de tipo String, y el precio de tipo Float.
* Pelicula, cuyos atributos son el título, el protagonista y el di­rector de tipo String, y el precio de tipo Float.
* Las dos clases contienen sus respectivos constructores, métodos *getters* y *setters* y el método toString().
* Implementa las clases Libro y Pelicula considerando los atributos, construc­tores y métodos descritos. Además, escribe una clase principal para crear varios Libros y varias Películas, y saber su título, autor, protagonista, director y pre­cio, según corresponda.

3.- En el ejercicio anterior, el atributo autor de Libro y los atributos protagonista y direc­tor de Película se definieron como de tipo String. Dado que estos atributos contienen el nombre de una persona, esta definición puede no ser homogénea. Por lo tanto, un mejor modelo sería considerar la clase adicional:

* Persona, cuyos atributos son nombre y apellido, de tipo String.

Esta clase, además de contener sus respectivos métodos *getters* y *setters*, considera el método:

* esIgual(Persona p), que devuelve true si el nombre y apellido de un objeto Persona son iguales a las de la persona p y false en caso con­trario.

De esta forma, las clases Libro y Pelicula se tienen que redefinir como sigue:

* Libro, cuyos atributos son autor de tipo Persona, titulo de tipo String y precio de tipo Float.
* Pelicula, con los atributos protagonista y director de tipo Per­sona, titulo de tipo String y precio de tipo Float.

Implementa la clase Persona y la redefinición de las clases Libro y Pelicula, considerando los atributos y métodos descritos.

4.- Vamos a generalizar las clases Libro y Pelicula del ejercicio anterior.

Como nos podemos dar cuenta, las clases Libro y Pelicula tienen dos atributos en común: titulo y precio. Entonces, podemos considerar una generalización de estas dos clases para crear la superclase:

* Producto, cuyos atributos son el titulo y el precio de un producto, de tipo String y Float, respectivamente. Generalmente los productos es­tán relacionados con un identificador único, por lo tanto está clase también tiene el atributo id de tipo Integer. Crear producto como una clase abstracta.

De esta forma, ahora las clases Libro y Pelicula son subclases de Producto y deben heredar de ésta sus atributos y métodos. Además, dado que los productos se van a vender, es necesario contar con un catálogo que los clientes puedan revisar. Por esta razón, se considera la clase:

* Catalogo, que cuenta con un atributo productos, que contiene a los li­bros y películas en venta (por ejemplo, un arreglo suficientemente grande de Productos), y con otro para conocer la cantidad de productos que están disponibles.

Todas las clases anteriores consideran sus respectivos constructores, métodos *getters* y *setters* y el método toString(). Adicionalmente, la clase Catalogo considera los siguientes métodos:

* agregar(Producto p), agrega el Producto p al catálogo.
* eliminar(Integer id), elimina el producto cuyo identificador único es id.
* buscar(String titulo), devuelve, contenidos en un Catalogo, a to­dos los Productos cuyo título es titulo.
* buscar(Persona p), devuelve, contenidos en un Catalogo, a todos los Productos cuyo autor, director o protagonista, según sea el caso, es p.

Implementa las clases Persona, Libro y Catalogo, considerando los atributos, constructores y métodos descritos.

5.- Elaborar una clase RACIONAL que modele los números racionales implementando al menos las operaciones de suma, resta, opuesto e inverso de un número racional a imitación de la suma o resta de los números reales o enteros.

6.- En un puerto se alquilan amarres para barcos de distinto tipo. Para cada ALQUILER se guarda el nombre y DNI del cliente, las fechas inicial y final de alquiler, la posición del amarre y el barco que lo ocupará.

Un BARCO se caracteriza por su matrícula, su eslora en metros y año de fabricación.

Un alquiler se calcula multiplicando el número de días de ocupación (incluyendo los días inicial y final) por un módulo función de cada barco (obtenido simplemente multiplicando por 10 los metros de eslora) y por un valor fijo (300 euros en la actualidad).

Sin embargo ahora se pretende diferenciar la información de algunos tipos de barcos:

* número de mástiles para veleros
* potencia en CV para embarcaciones deportivas a motor
* potencia en CV y número de camarotes para yates de lujo.

El módulo de los barcos de un tipo especial se obtiene como el módulo normal más:

* el número de mástiles para veleros
* la potencia en CV para embarcaciones deportivas a motor
* la potencia en CV más el número de camarotes para yates de lujo.

7.- Codifica los siguientes diagramas de clase y ejecuta en la clase MAIN instanciando los objetos necesarios.

