1) Desarrollar un programa en Java que utilice una clase que se llame ***Indicador*** que sea la representación de un indicador de la vida real (indicadores de nivel, de velocidad, tacómetros, etc).
La clase deberá tener los atributos valorMinimo, valorMaximo, valorActual, unidades y nombre del indicador. En el momento de la creación de cada objeto se asignarán todos los elementos salvo el valorActual. Entre los métodos que se programarán deberán estar el de asignación del valorActual, imprimir el valor, imprimir el indicador e incrementar y disminuir el valorActual con el que cuentan.
En ese programa (el principal), crearás un vector de elementos de tipo Indicador y se podrá escoger el indicador a tratar.

2) Desarrollar un programa en Java que utilice una clase que se llame ***Hora*** con miembros de tipo ***Integer*** para hora, minutos y segundos. Deberá tener un constructor para inicializar la hora a 0 y otro para incializar a una hora determinada (hora, minutos, segundos. La hora deberá ser una hora con valores posibles que hemos de controlar).Se deberá poder sumar y restar horas, así como imprimir una hora, ver la conversión a segundos de una hora dada, sumar segundos a una hora dada.
El formato de impresión y lectura será hh:mm:ss, todo en modo 24 horas.

3) Diseña la clase TragaBolas que tiene los siguientes atributos y métodos:

TragaBolas
- color: **String**. Color del tragabolas. Sólo puede ser azul, amarillo, rojo o verde. **- bolasComidas**: **Integer**. Número de bolas que ha comido hasta el momento. **- maxBolas**: I**nteger**. La cantidad máxima de bolas que puede comer.
+ TragaBolas(String, int): Pide por teclado el color y maxBolas. Las bolasComidas se inicializan a 0. **+ visualizar()**: Muestra los datos del tragabolas por pantalla. **+ comer()**: sólo puede comer si bolasComidas es menor que maxBolas, esta acción sumará 1 a bolasComidas y mostrará por pantalla "He comido una bola"**.** **+ trotar()**: sólo puede trotar si bolasComidas es mayor o igual que 1, esta acción restará 1 de bolasComidas y mostrará por pantalla "Estoy trotando". **+ dormir():** sólo puede dormir si bolasComidas es igual a maxBolas. Mostrará en pantalla “Tripa llena. ZZZZZZ” y rebajará bolasComidas a la mitad. Si no cumple la condición para poder dormir mostrará en pantalla: “No quiero dormir”.

En el método main de la clase Principal hay que mostrar un menú con las siguientes opciones:

1: Crear tragaBolas.
2: Darle de comer.
3: Hacerle dormir.
4: Ver estado.
0: Fin.

4) Vamos a crear la clase CuentaCorriente, con las siguientes propiedades y comportamiento:

CuentaCorriente
- numCuenta: **String**.Será el número de la cuenta corriente. **- saldo**: **Double**. Saldo actual de la cuenta. - **cliente: String**. Nombre de cliente.
+ CuentaCorriente(String cuenta, Double cantidad, String cliente) **+ ingresaEfectivo(Double cantidad)** : el parámetro que recibe se lo suma al saldo. **+ retiraEfectivo(Double cantidad)**: **Boolean** el parámetro indica la cantidad que queremos retirar. Si hay saldo, restará el importe y devolverá true, en caso contrario devolverá false y no realizará ninguna operación. **+ visualiza()** : Mostrará por pantalla la información de la cuenta corriente: Número de cuenta y saldo.

El método main de la clase principal creará dos cuentas corrientes: una con número de cuenta 001 y otra con número de cuenta 002, ambas con 0 € de saldo y clientes distintos, posteriormente mostrará el siguiente menú por pantalla:

1: Ingresar en la cuenta 001
2: Ingresar en la cuenta 002
3: Retirar de la cuenta 001
4: Retirar de la cuenta 002
5: Visualizar ambas cuentas
0: Fin

Dicho menú se ejecutará realizando las operaciones oportunas para cada opción hasta que el usuario elija la opción de fin. El saldo se debe presentar con solo dos decimales.

5) Crea la clase **Fecha**:

Fecha
- dia: **Integer**. Número entero que guarda el día del mes. **- mes**: **Integer**. Número entero que guarda el mes. **- año**: **Integer**. Número entero que guarda el año.
Constructor con tres parámetros uno para el día, otro para el mes y otro para el año, que inicializa los tres atributos del objeto con los parámetros pasados. La fecha debe ser legal. Desarrolla los **getters** y **setters** para todos los atributos. **+ esBisiesto(): Boolean**. Sin parámetros. Devuelve true si el año es bisiesto y false si no lo es. **+ esCorrecta(): Boolean.** Comprueba si la fecha es correcta. Devuelve un valor de tipo boolean indicando si la fecha es correcta o no. Este método a su vez utilizará el método anterior esBisiesto, cuando sea necesario, que calcula si el año es o no bisiesto. **+ getCadenaMes(): String.** Sin parámetros. Devuelve una cadena que contiene el mes con letras: enero, febrero, marzo, etc. **+ getDiasMes(): Integer**. Devuelve el número de días que tiene el mes. Teniendo en cuenta que si el año es bisiesto el mes de febrero tendrá 29 días. **+ getCadenaFecha1()**: **String.** Sin parámetros. Devuelve una cadena con la fecha en formato 'dd-mm-yyyy'. **+ getCadenaFecha2()**: **String.** Sin parámetros. Devuelve una cadena con la fecha en formato '5 de Mayo de 2016'. **+ diferenciaFecha(Fecha fecha2)**: **Integer.** Devuelve un Integer con la diferencia de días entre la fecha del objeto que invoque el método y fecha2. **+ sumaDias(Integer num)**: **Fecha.** Devuelve una Fecha a partir del objeto con el que se invoca el método y el número de días que se pasan como parámetro.

Puedes añadir más métodos si lo consideras necesario.
Crea la clase Principal en la que:
Haya un menú para usar las opciones que se plantean con los métodos indicados.
Sobre todo, que el constructor solo cree una fecha si los valores de día, mes y año se corresponden con una fecha posible, si no, no se debe crear el objeto hasta que los valores sean correctos.