# Programación orientada a objetos

## Guía de ejercicios #1

**Algunos de los siguientes ejercicios propuestos, serán resueltos en la práctica por los instructores de forma expositiva:**

1. Desarrollar una calculadora que permita realizar las operaciones aritméticas básicas (sumar, multiplicar, restar, dividir), la aplicación debe mostrar los resultados en consola, debe solicitar los dos números a utilizar, y la operación a realizar, además se debe mostrar un menú en consola con las 5 siguientes opciones disponibles:
    a. Sumar
    b. Multiplicar
    c. Restar
    d. Dividir
    e. Salir
2. Elaborar una aplicación que le solicite un número por consola y lo multiplique por uno que se genere aleatoriamente. Posteriormente muestre el resultado de la multiplicación y muestre ambos números en consola.
3. Elaborar una aplicación que le permita representar el factorial de un número solicitándole el número deseado y mostrando el resultado usando JOptionPane.
4. Elabore una aplicación que le solicite un número utilizando JOptionPane y le devuelva, en otro JOptionPane, el promedio de la suma de dicho número con todos sus antecesores mayores a cero.
5. Crear una aplicación que haga uso de una clase llamada cuenta que tendrá los siguientes atributos: titular y cantidad (puede tener decimales) y posteriormente cree dos objetos de tipo cuenta y manipúlelos como considere conveniente. Tener en cuenta que:
    a. El titular será obligatorio y la cantidad es opcional (crear dos constructores que cumplan lo anterior).
    b. Crear sus métodos get, set y toString.
    c. Tendrá dos métodos especiales:

i. *ingresar(double cantidad)*: se ingresa una cantidad a la cuenta. Si la cantidad introducida es negativa, no se hará nada.
ii. *retirar(double cantidad)*: se retira una cantidad a la cuenta. Si el resultado de restar la cantidad actual a la que nos pasan es negativo, la cantidad de la cuenta pasa a ser 0.

6. Crear una aplicación que haga uso de una clase llamada Contador que contenga un único atributo entero llamado cont, la clase tendrá los siguientes constructores:
    a. Constructor por defecto.
    b. Constructor con parámetros para inicializar el contador con un valor no negativo. Si el valor inicial que se recibe es negativo el contador tomará el valor cero como valor inicial.
    c. Además de los métodos getter y setter, la clase contendrá los métodos:
        i. *incrementar( )*: incrementa el contador en una unidad.
        ii. *decrementar( )*: decrementa el contador en una unidad. El contador nunca podrá tener un valor negativo. Si al decrementar se alcanza un valor negativo el contador toma el valor cero.

**Ejercicios para que los estudiantes desarrollen en la práctica de laboratorio (al menos uno de la siguiente lista)**

1. Crear una clase Libro que contenga los siguientes atributos: ISBN, Titulo, Autor, Número de páginas. Crear los respectivos métodos get y set correspondientes para cada atributo además crear el método *toString( )* para mostrar la información relativa al libro con el siguiente formato:
   "El libro con ISBN creado por el autor tiene páginas". En el fichero *main*, crear dos objetos Libro (los valores que se quieran) y mostrarlos desplegando un JOptionPane, por último, indicar cuál de los dos tiene más páginas usando JOptionPane.

2. Crear una clase llamada Password que siga las siguientes condiciones:
    a. Que tenga los atributos longitud y contraseña. Por defecto, la longitud será de 8.
    b. Los constructores serán los siguientes:
        i. Un constructor por defecto.

ii. Un constructor con la longitud que nosotros le pasemos. Generará una contraseña aleatoria con esa longitud (usando random).

c. Los métodos que implementa serán:

i. *esFuerte( )*: devuelve un booleano que indique si la contraseña es fuerte o no. Para que sea fuerte debe tener más de dos mayúsculas, más de una minúscula y más de cinco números (se puede apoyar de ciclos, recordar que las cadenas se pueden recorrer por posición al igual que los arreglos).

ii. generarPassword (): genera la contraseña del objeto con la longitud que tenga.

iii. Método get para contraseña y longitud.

iv. Método set para longitud.

Instanciar un objeto de la clase, generar una contraseña y desplegar en pantalla, usando JOptionPane, si la contraseña es fuerte o no.

3. Elaborar una clase llamada Productos, que almacene nombre y precio. Incluir todos sus getters y setters, además incluir un constructor que reciba los parámetros descritos, y otro que no reciba ningún parámetro. Posteriormente, en el main cree un programa que le solicite la cantidad de productos que desea agregar y posteriormente le permita agregar la cantidad de productos que el usuario considere pertinentes dentro de un arreglo de objetos de tipo Productos y al finalizar muestre la suma de los precios de todos ellos utilizando un JOptionPane.
4. Crear una clase llamada Persona, con los atributos: nombre, edad, altura y construir los siguientes métodos para la clase:
    a. Un constructor, donde los datos pueden estar vacíos.
    b. Los setters y getters para cada uno de los atributos.
    c. *mostrar( )*: muestra los datos de la persona.
    d. *esMayorDeEdad( )*: Devuelve un valor lógico indicando si es mayor de edad.

    Considerar validar las entradas de datos, posteriormente solicitar la cantidad de personas que se desean ingresar dentro de un arreglo de objetos, y solicitar los datos necesarios usando JOptionPane mostrando finalmente los detalles de las personas que son mayores de edad.