

Ejercicio 1: En este ejercicio vamos a crear una clase java que represente las características y comportamientos básicos de una calculadora que opera con dos números enteros, es decir que solo puede realizar la suma, resta, multiplicación y división entre 2 números enteros (operandos).

Dentro del paquete `ar.edu.unju.fi.ejercicio1`, crear una clase java que tenga como nombre **Calculadora**.

En esta clase declare los atributos representativos de una calculadora. Ejemplo: color, marca, tamaño, `operando1`, `operando2`. Seleccione los tipos de datos adecuados para cada uno de los atributos.

Declare dos constructores para la clase **Calculadora**, uno que recibe parámetros y otro que no recibe parámetros.

Declare los métodos que representan el comportamiento de la calculadora: sumar, restar, multiplicar, dividir, cada método debe retornar el resultado de la operación matemática, determine el tipo de dato primitivo adecuado a retornar (byte, short, int, long, float, double).

Declare los métodos accesoros para poder acceder a los atributos desde otras clases.

Sobreescriba el método **toString()** para poder visualizar el estado de los objetos de la clase **Calculadora**.

En el mismo paquete cree la clase denominada **Principal**. Observe detenidamente las opciones del asistente del IDE y seleccione la opción que permita agregar el método **main()** a la clase.

Dentro del método **main()** de la clase **Principal** declare y cree un objeto de la clase **Calculadora** utilizando los dos constructores de la clase **Calculadora**, luego muestre el estado de los objetos y el resultado de cada método por consola.

Ejercicio 2: En este ejercicio vamos a crear una clase java que represente las características y comportamientos de una cuenta bancaria. El nombre de la clase: **Cuenta**, algunos atributos de la clase: número cuenta (entero grande), dni de cliente (entero grande), saldo actual (decimal). Entre sus métodos podemos declarar: `depositar(double monto)`, `extraer(double monto)` aquí se debe verificar si existe saldo sino mostrar un mensaje que no se puede realizar la extracción, el método `depositar` aumenta el saldo actual con el valor `monto` y el método `extraer` disminuye el saldo actual según el `monto` enviado como parámetro.

Crear dos constructores de clase, con parámetros y sin parámetros.

Agregue los métodos accesoros (getters y setters) para los atributos.

Sobreescriba el método **toString()** para mostrar el estado del objeto por consola.

Crear la Clase **Principal** incluyendo en ella el método **main()**.

Genere el código necesario para ingresar por consola los datos que permitan crear 2 cuentas con sus respectivos datos, luego realice depósitos y extracciones en la cuentas, mostrando siempre el estado del objeto por consola.

Ejercicio 3: En este ejercicio vamos a crear una clase java denominada **ProcesoFecha**, la cual define características y comportamientos relacionados al tratamiento de una fecha.

Los atributos para la clase: día, mes y año, de tipo numérico entero.

Agregue en la declaración de la clase:

Los métodos accesorios (getters y setters).

Un constructor predeterminado (sin parámetros) con el valor 1 para día, valor 1 para mes y valor 1900 para el año.

Un constructor parametrizado con día, mes y año.

Un método llamado **validar()** que realizará la validación de los valores de los atributos: el día (1 al 31), el mes (1 al 12) y el año (1900 a 2021). Si el día no es correcto, asignarle el valor 1, si el mes no es correcto, asignarle el valor 1 y si el año no es correcto entonces asignarle el valor 1900.

Debe considerar qué mes se ingresó para determinar cuántos días tiene el mes y también verificar si el año es bisiesto para permitir una fecha como esta: 29/02/2000. El método **validar()** devuelve un String que es el resultado de invocar al método **mostrarFormatoSimple()**

Un método llamado **verificarBisiesto()**, este método devuelve un valor booleano, verdadero si el año ingresado es bisiesto y falso si el año no es bisiesto.

Un método llamado **devolverCantidadDiasMes()**, devolverá la cantidad de días del mes.

Un método llamado **mostrarFormatoSimple()**, retornará un String con la siguiente cadena armada con día, mes y año. Ej: "21-11-2010"

Un método llamado **mostarFormatoReducido()**, retornará un String con la siguiente cadena armada con día, mes y año. Ej: "21/11/10". Utilice los métodos de la clase **String** para manipular las cadenas.

Un método llamado **mostrarNombreMes()**, retornará un String que corresponda con el nombre del mes ingresado, utilice alguna estructura de control de selección para determinar el nombre del mes.

Ej: 12 = Diciembre, 5 = Mayo, 3 = Marzo

Luego agregue dentro del mismo paquete la clase **Principal** y construya un objeto de la clase **ProcesoFecha** utilizando el constructor sin parámetros, invoque todos los métodos para el objeto creado y muestre por consola los resultados.

Construya un nuevo objeto de la clase **ProcesoFecha** ahora utilizando el constructor con parámetros, solicite al usuario que ingrese por consola los valores para día, mes y año. Luego muestre por consola todos los resultados de los métodos.

Ejercicio 4: Cree una clase denominada **CalculoMatematico**.

Los atributos de clase son: n valor numérico entero.

Agregue en la declaración de la clase:

Los métodos accesorios (getters y setters).

Un constructor parametrizado.

Un método llamado **calcularSumatoria()** que devuelva el resultado del siguiente cálculo:

$$\sum_{k=1}^n \left[\frac{n * (n + 1)}{2} \right]^2$$

$$\sum_{k=1}^n \left[\frac{n * (n + 1)}{2} \right]^2 = \left[\frac{1 * (1 + 1)}{2} \right]^2 + \left[\frac{2 * (2 + 1)}{2} \right]^2$$

Utilice la estructura de control iterativa **“for”**

Un método llamado **calcularProductoria()** que devuelva el resultado del siguiente cálculo:

$$\prod_{k=1}^n k * (k + 4)$$

$$\prod_{k=1}^n k * (k + 4) = 1 * (1 + 4) * 2 * (2 + 4)$$

Utilice la estructura de control iterativa **“for”**

Luego agregue la clase **Principal** con el método **main()** y solicite al usuario que ingrese por consola el valor de n para construir un objeto de la clase **CalculoMatematico**.

Invoque los métodos del objeto y muestre el resultado por consola.

Ejercicio 5: Idem al ejercicio 4 pero utilizando la estructura iterativa **“while”**

Ejercicio 6: Idem al ejercicio 4 pero utilizando la estructura iterativa **“do while”**