

PRÁCTICA 1 – Java: Conocimiento del IDE – Sintaxis JAVA

1. Escribir un programa que en el método *main*, solicite un nombre y muestre “Hola <nombre ingresado>” y debajo, la fecha y hora actual con el siguiente formato de ejemplo “viernes 20/04/2018 18:00”. Validar que el mensaje no se muestre hasta tanto no se haya ingresado un nombre.
2. Escribir un programa en el que se ingresen 2 cadenas e invocando a un método privado y estático de la clase, devuelva si son iguales o no. De acuerdo al resultado del método, mostrar en consola “Son iguales” o “Son distintas”.
¿Qué sucede si el método que hace la comparación no se declara como estático?
3. Escribir un programa que imprima tres **argumentos** tomados desde la **línea de comando**. Para hacer esto, necesitará usar los índices del arreglo de strings que recibe como parámetro. Asegurarse de que se reciban al menos tres parámetros antes de imprimirlos.
4. Escribir un programa que muestre los números del 1 a N uno al lado del otro, separados por un espacio. N es un valor entero ingresado por teclado.
5. Definir una clase que posea un método al que se le pasen 2 **números** enteros como **parámetros** y devuelva el resultado de la suma, el producto y la resta de los mismos.
6. Escribir un programa que imprima los **factoriales** de los primeros diez números naturales. ($n! = n * (n-1) * (n-2) * \dots * 2 * 1$)
7. Escriba un programa que utilice el método **static getProperty()** de la clase **System** para obtener información del sistema. La información que se desea obtener es: versión de Java, sistema operativo y su versión.
8. Crear una sentencia **switch** dentro de un programa que imprima un mensaje para cada caso. Colocar un break en cada caso y testearlo. Luego remover el break y verificar que sucede.
9. Escribir un programa que calcule la **raíz cuadrada** de un número real y muestre su parte entera. ¿Cómo modificaría el programa para que el resultado se redondee al entero más próximo?
10. Definir la clase **ClienteBanco** con tres datos miembro: nombre, PIN, saldo. Implementar funciones para mantener los atributos, depositar y extraer dinero, y mostrar el estado de la cuenta. Para operar, el cliente debe ingresar un número, y éste debe coincidir con el PIN particular.
11. Crear una clase **Hora** que contenga datos miembro de tipo entero para *horas*, *minutos* y *segundos*.
Definir dos constructores: un constructor inicializará los datos en 0 y otro a valores variables. Una función miembro mostrará la hora en formato hh:mm:ss. Otra función sumará dos objetos de tipo hora pasados como argumento.
Al iniciar la aplicación, crear tres objetos de la clase **Hora** inicializando con valores a 2 de ellos, que deberán sumarse almacenando el resultado sobre el tercero. Mostrar el resultado.
12. Haciendo uso del concepto de **Composición**, crear una clase **Automovil** con atributos patente, dueño y motor. El motor es un atributo de tipo **Motor**. La clase motor tiene dos atributos: cilindrada y tipo (naftero, diesel, eléctrico).
Crear 3 Automóviles y agregarlos en un vector de 3 elementos de tipo Automovil.
Recorrer el arreglo y, para cada objeto de la clase Automóvil, mostrar toda la información disponible del mismo.