**Tarea Backend**

1)    Investigar el uso y significado de 5 palabras reservadas (Incluir fragmentos de código si es posible).

**exports**

Se utiliza en java modular para exportar un paquete con un módulo. Esta palabra clave sólo está disponible en Java 9 y versiones posteriores.

Ejemplo:

module com.tutorialspoint.model {

exports tp.com.tutorialspoint.model;

}

**byte**

La palabra clave byte se utiliza para declarar un campo que puede contener un entero de complemento de dos con signo de 8 bits. Esta palabra clave también se utiliza para declarar que un método devuelve un valor del byte tipo primitivo.

Ejemplo:

byte b = 100;

**enum (agregada en J2SE 5.0)**

Una palabra clave de Java utilizada para declarar un tipo enumerado. Las enumeraciones amplían la clase base Enum.

Ejemplo:

enum Level {

LOW,

MEDIUM,

HIGH

}

Y se accede de la siguiente forma:

Level myVar = Level.MEDIUM;

**finally**

Se utiliza para definir un bloque de instrucciones para un bloque definido previamente por la palabra clave try. El bloque finally se ejecuta después de que la ejecución sale del bloque try y de cualquier cláusula catch asociada sin importar si se lanzó o capturó una excepción, o si se ejecutó el método de izquierda en medio de los bloques try o catch usando la palabra clave return.

**native**

Se usa en declaraciones de métodos para especificar que el método no se implementa en el mismo archivo fuente de Java, sino en otro idioma.

Ejemplo:

// Native method

public native void test(){

static {

// We will be loading body from DLL file

// It has to be present in DLL file

System.loadLibrary("NameOfDLLFile");

// Above C code in loaded in the JVM

}

}

2)    Investigar ¿cómo compilar un archivo .java y como ejecutarlo usando solamente la terminal de comandos (Terminal/CMD).?

En la ventana consola escribiremos el comando “cd (dirección donde guarde el programa java)” y pulsaremos enter. Esto nos traslada a la carpeta donde se ubica nuestro archivo (“JavaApp.java”).  Al escribir el comando dir y pulsar enter nos aparece el archivo JavaApp.java. A continuación, daremos la instrucción para que se realice **el proceso de compilación del programa**, para lo que escribiremos “javac JavaApp.java”, dando como resultado un nuevo archivo “JavaApp.class” que se creará en este mismo directorio

Finalmente, vamos a pedirle al intérprete que ejecute el programa. Para ello escribiremos en la ventana consola: java JavaApp. El resultado será que se nos muestra la cadena “Hola Mundo”.

Texto

Descripción generada automáticamente

Video Explicativo: <https://youtu.be/UFfLe4fBngY>

3)    Investigar ¿Cómo capturar datos introducidos desde nuestro teclado y elabora un pequeño programa utilizándolo?

Texto

Descripción generada automáticamente

Video Explicativo: <https://youtu.be/n711TCzD9Vg>

Link de código(repositorio): <https://github.com/JalilMartinez/Capacitacion-Infotec/blob/main/JAVA/capturaDatos/src/Main.java>

4)    Elabora un programa usando los diferentes operadores aritméticos vistos en clase, asigna los resultados de estas operaciones a distintas variables e imprima en consola los resultados usando estas variables.

Texto

Descripción generada automáticamente

Video Explicativo : <https://youtu.be/zSoTV6QeTfk>

Link de código (repositorio) : <https://github.com/JalilMartinez/Capacitacion-Infotec/blob/main/JAVA/capturaDatos/OperadoresAritmeticos/src/Main.java>

5)    Elabora un fragmento de código dónde realices 6 conversiones de tipos de datos ( 3 de manera implícita y 3 de manera explícita) e imprime en consola los resultados.

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media

Video Explicativo : <https://youtu.be/R_uuQwkRbAs>

Link de código (repositorio) : <https://github.com/JalilMartinez/Capacitacion-Infotec/blob/main/JAVA/capturaDatos/Conversion/src/Main.java>