

MatLib

Realiza una librería con las siguientes operaciones matemáticas que se detallan al final del documento.

Por cada función realiza la versión para valores long y otra para valores int

Las funciones a implementar son:

- **suma** (Devuelve la suma de dos números) **(1 puntos)**
- **inc** (Devuelve el incremento en una unidad de un valor dado) **(2 puntos)**
- **resta** (Devuelve la resta de dos números) **(2 puntos)**
- **dec** (Devuelve el decremento en una unidad de un valor dado) **(2 puntos)**
- **esMayor** (Devuelve si un número es mayor que otro) **(4 puntos)**
- **esMenor** (Devuelve si un número es menor que otro) **(4 puntos)**
- **esIgual** (Devuelve si un número es igual a otro) **(5 puntos)**
- **multiplica** (Devuelve el resultado de multiplicar dos números) **7 puntos)**
- **divide** (Devuelve el cociente de una división entera) **(10 puntos)**
- **resto** (Devuelve el resto de una división entera) **(9 puntos)**
- **potencia** (Devuelve el resultado de elevar un número a un valor) **(6 puntos)**
- **cuadrado** (Devuelve el resultado de elevar un número a 2) **(3 puntos)**
- **cubo** (Devuelve el resultado de elevar un número a 3) **(3 puntos)**
- **esMultiplo** (Indica si un número es múltiplo de otro dado) **(6 puntos)**
- **esDivisor** (Indica si un número es divisor de otro dado) **(4 puntos)**
- **esPrimo** (Indica si un número es primo) **(12 puntos)**
- **esCuadradoPerfecto** (Indica si un número es el resultado del cuadrado de otro) **(20 puntos)**

Existen una serie de **reglas** a la hora de implementar las funciones.

- La única operación aritmética que se permite usar es '+', no se permite usar '++'
- Se puede usar el operador '-' siempre y cuando sea para cambiar el signo de una variable pero no para operar entre dos variables. Por ejemplo $a = -b$ está permitido pero $a = b - c$ no lo está. Tampoco se puede utilizar el operador '--'.
- Solo se puede usar el operador de comparación '>' (los comparadores '<' e '==' no están permitidos) y no se permite comparar dos variables: Por ejemplo: $(a > 0)$ se permite, $0 > a$ se permite y $a > b$ (no se permite).
- Se pueden usar todos los operadores lógicos '||', '&&', '!'.

Descarga el archivo zip adjunto a la tarea y abre la carpeta _Examen1312 en Visual Studio. Dentro de la carpeta _Examen1312, crea la carpeta donde va estar tu librería y llámala matematicas

Luego dentro de esa carpeta crea el archivo Operaciones.java que contendrá la clase con todas las funciones implementadas.

Recuerda que el paquete se debe llamar matematicas y la clase se debe llamar Operaciones

A continuación, en el archivo Examen13122022.java importa la librería.

Verás que hay un programa que realiza pruebas de tu librería para ver que la librería funciona correctamente. Si ves la salida siguiente:

Pruebas unitarias

Función suma: correcto

Función inc: correcto

Función resta: correcto

Función dec: correcto

Función esMayor: correcto

Función esMenor: correcto

Función esIgual: correcto

Función multiplica: correcto

Función divide: correcto

Función resto: correcto

Función potencia: correcto

Función cuadrado: correcto

Función cubo: correcto

Función esMultiplo: correcto

Función esDivisor: correcto

Función esPrimo: correcto

Función esCuadradoPerfecto: correcto

Significa que las funciones dan el resultado esperado aunque puede ser que no estén bien implementadas ya que se deben cumplir las reglas establecidas anteriormente y no hacer trampas a la hora de dar directamente los resultados esperados en las pruebas unitarias.

Si alguna función indica fallo, quiere decir que no está correctamente implementada.

Glosario de funciones detalladas

```
/** suma
 * Devuelve la suma de dos números.
 * <p>
 * @param a Sumando primero
 * @param b Sumando segundo
 * @return La suma de a y b
 */
```

```
/** inc
 * Incrementa en uno el valor recibido y lo devuelve
 * <p>
 * @param a Valor a incrementar
 * @return El valor recibido incrementado en uno
 */
```

```
/** resta
 * Devuelve la resta de dos números
 * <p>
 * @param a Minuendo
 * @param b Sustraendo
 * @return La resta de los dos números
 */
```

```
/** dec
 * Decrementa en uno el valor recibido y lo devuelve
 * <p>
 * @param a Valor a decrementar
 * @return El valor recibido decrementado en uno
 */
```

```
/** esMayor
 * Comprueba si un número es mayor que otro
 * <p>
 * @param a Valor a comparar si es mayor
 * @param b Valor a comparar si es menor
 * @return true si a es mayor que b y
 *         false si es igual o menor
 */
```

```
/** esMenor
 * Comprueba si un número es menor que otro
 * <p>
 * @param a Valor a comparar si es menor
 * @param b Valor a comparar si es mayor
```

```
* @return true si a es menor que b y
*         false si es igual o mayor
*/
```

```
/** esIgual
 * Comprueba si un número es igual que otro
 * <p>
 * @param a Primer valor a comparar
 * @param b Segundo valor a comparar
 * @return true si son iguales y false si son distintos
 */
```

```
/** multiplica
 * Devuelve la multiplicación de dos números usando sumas
 * <p>
 * @param a Multiplicando
 * @param b Multiplicador
 * @return El resultado de sumar b veces a
 */
```

```
/** divide
 * Devuelve el cociente de la división entera de dos números usando restas
 * <p>
 * @param a Dividendo
 * @param b Divisor
 * @return Devuelve el cociente de la división entera de a entre b
 *         realizando restas
 */
```

```
/** resto
 * Devuelve el resto de la división entera
 * <p>
 * @param a Dividendo
 * @param b Divisor
 * @return Devuelve el resto de la división entera de a entre b
 */
```

```
/** potencia
 * Devuelve la potencia de un número elevado a otro
 * <p>
 * @param base    base
 * @param exponente exponente
 * @return        Devuelve la potencia de un número elevado a otro usando
multiplicaciones
 */
```

```
/** cuadrado
 * Devuelve la potencia de un número elevado a 2
 * <p>
 * @param a número a calcular su cuadrado
 * @return Devuelve la potencia de un número elevado a 2
 */
```

```
/** cubo
 * Devuelve la potencia de un número elevado a 3
 * <p>
 * @param a número a calcular su cubo
 * @return Devuelve la potencia de un número elevado a 3
 */
```

```
/** esMultiplo
 * Indica si un número es múltiplo de otro dado
 * <p>
 * @param a valor a comprobar si es múltiplo
 * @param b valor con el que comprobar si a es múltiplo de él
 * @return true si a es múltiplo y false en caso contrario
 */
```

```
/** esDivisor
 * Indica si un número es divisor de otro dado
 * <p>
 * @param a valor a comprobar si es divisor
 * @param b valor con el que comprobar si a es divisor de él
 * @return true si a es divisor y false en caso contrario
 */
```

```
/** esPrimo
 * Indica si un número es primo
 * <p>
 * @param a número a comprobar si es primo
 * @return true si a es primo y false en caso contrario
 */
```

```
/** esCuadradoPerfecto
 * Indica si un número es cuadrado perfecto
 * <p>
 * Un número es cuadrado perfecto si es el resultado del cuadrado de otro número
 * @param a número a comprobar si cuadrado perfecto
 * @return true si a es cuadrado perfecto y false en caso contrario
 */
```