

CREACIÓN DE FUNCIONES Y USO DE PARÁMETROS

ENUNCIADO

En esta práctica aprenderemos a crear funciones para poder aprovechar y reutilizar el código creado hasta el momento. De esta forma, podremos volver a utilizar estas funciones más adelante en nuestras aplicaciones, si fuera necesario.

El nombre del proyecto deberá contener las iniciales de vuestro nombre y primer apellido. Por ejemplo para el nombre José Luis Berenguel el nombre del proyecto sería: **jlberenguel_funciones**.

El proyecto consta de dos partes, en la primera parte escribiremos el código de cada una de las funciones, adaptando el código que ya hemos creado en los ejercicios previos. En la segunda parte, reescribiremos el código de los programas para que hagan uso de la llamada a las funciones.

Primera parte

Las funciones se incluirán en un fichero **Operaciones.java**. Todas las funciones deben incluir en su declaración los modificadores **public static**. La lista de funciones es la siguiente:

- **public static double areaTriangulo(double base, double altura)** (ejercicio 0 unidad 1)
- **public static double volumenCono(double radio, double altura)** (ejercicio 4 unidad 1)
- **public static double ecuacionPrimerGrado(double a, double b)** (ejercicio 12 unidad 1)
- **public static double tiempoCaida(double altura h)** (ejercicio 13 unidad 1)
- **public static boolean esBisiesto(int año)** (ejercicio 8 relación I)
- **public static double potencia(double base, int exponente)** (ejercicio 9 relación I)
- **public static int numeroCifras(int numero)** (ejercicio 10 relación I)
- **public static boolean esCapicua(int numero)** (ejercicio 12 relación I)
- **public static double costeLlamadaTelefonica(int segundos)** (ejercicio 16 relación I)
- **public static boolean esPrimo(int numero)** (ejercicio 20 relación I)
- **public static boolean esPerfecto(int numero)** (ejercicio 3 relación II)
- **public static String conversorDigitoPalotes(int digito)** (ejercicio 11 relación II)

Para implementar estas funciones, debes reutilizar el código creado en los ejercicios de las relaciones de la unidad 1 y que has escrito en Java en esta unidad.

Las funciones deben comprobar sus parámetros de entrada, pero estas no son responsables de volver a pedir nuevos datos. Si los datos recibidos no son válidos, el problema de validación está en la clase que ha realizado la llamada a la función, puesto que esta llamada puede realizarse desde numerosos lugares (interfaz gráfico, consola, etc.).

La forma de validar los parámetros de una función es la siguiente: en caso de que las funciones reciban valores no permitidos se deberá lanzar la excepción ***IllegalArgumentException***. Por ejemplo, en el caso del método `areaTriangulo()`, si la base o la altura no son positivos:

```
if(base<=0){  
    throw new IllegalArgumentException("La base debe ser mayor que cero");  
}  
if(altura<=0){  
    throw new IllegalArgumentException("La altura debe ser mayor que cero");  
}
```

Más información sobre cómo validar los argumentos de una función en el siguiente enlace.

Java Practices. Validate method arguments.
<http://www.javapractices.com/topic/TopicAction.do?Id=5>

Segunda parte

En esta segunda parte debes crear nuevos programas principales (main) para cada uno de los ejercicios correspondientes a las funciones que has creado. Recuerda que debes solicitar los datos de entrada al usuario y validarlos correctamente.