


| | |
|--------------------------------------|---|
| Programación |  |
| <i>1º Evaluación</i> | |
| Tema 03. Programación modular | |

RELACIÓN DE PROBLEMAS 4. Programación modular

Resolver los siguientes problemas en Java. Realizarla en un sólo proyecto llamado Relacion4, con un paquete por cada ejercicio. En toda la relación se usarán métodos **public static**.

1. Realizar un método llamado **solicitarNumeroEnRango**. El método devuelve un entero y tendrá como parámetros de entrada:
 - límite inferior
 - límite superior

El método solicitará un número comprendido entre esos dos valores y lo devolverá. Si no está en el rango lo volverá a solicitar. Si el límite inferior es mayor al límite superior, los intercambia.

Probarlo desde el **main**.

2. Realizar un método que dado un número positivo devuelva su factorial. Si el número es negativo se mostrará un mensaje de error y se devolverá un -1.

Utilizar este método desde el **main** para escribir los factoriales desde 0 hasta un número límite que se pida por teclado.

La salida en pantalla será de esta forma:

```
Introduce un número límite para calcular factoriales: 4
0! = 1
1! = 1
2! = 2
3! = 6
4! = 24
```

3. Realizar un método que informe de si un año es o no bisiesto. Un año es bisiesto si es múltiplo de 4, a excepción de los que, siendo múltiplos de 4 son múltiplos de 100 y no de 400. Probarlo desde el **main**.
4. Realizar un método llamado **diasDeUnMes** que tenga como entrada un número de mes (entre 1 y 12) y un año y devuelva el número de días de ese mes y año. Usar el método anterior para cuando el año sea bisiesto, febrero tenga 29 días.

5. Realizar un método llamado **pintarNuevaHora** al que se le pasen 4 parámetros enteros:

- horaActual
- minutosActuales
- segundosActuales
- segundosASumar

El método no devuelve nada y debe mostrar dos mensajes:

```
La hora anterior era H: M: S
La nueva hora es H: M: S
```

El método supondrá que los datos de entrada son correctos. Esta validación de datos se realizará desde el **main**.

Probar el método desde el **main**.

6. Realizar un método llamado **siguienteFecha** al que se le pasen 3 parámetros enteros:

- diaActual
- mesActual
- annoActual

El método no devuelve nada y debe mostrar dos mensajes:

```
La fecha de hoy es D-M-A
La fecha de mañana es D-M-A
```

El método supondrá que los datos de entrada son correctos. Esta validación de datos se realizará desde el **main**.

Probar el método desde el **main**.

7. Realizar un método llamado **diaDeLaSemana** al que se le pasen 3 parámetros enteros:

- dia
- mes
- anno

El método debe devolver en un String el día de la semana que era esa fecha. Para ello se tendrá en cuenta el siguiente algoritmo:

```
a = (14 - mes) / 12
y = anno - a
m = mes + 12 * a - 2
d = (día + y + y/4 - y/100 + y/400 + (31*m)/12) mod 7
```

Si el resultado de d es un cero (0) era domingo, 1 para el lunes... 6 para el sábado.