Aspectos avanzados en Groovy - Ejercicios

Índice

1. Creación de closures (I)

Implementar un closure que realice la operación matemática factorial del número pasado como parámetro. El archivo debe llamarse ej31.groovy.

2. Closures sobre listas y mapas

Crear una lista de números enteros y utilizar algunos de los métodos vistos hasta ahora para recorrer la lista, generar el número factorial de cada uno de sus elementos con el closure que hemos creado en el ejercicio 1 y mostrarlo por pantalla. El archivo debe llamarse ej32.groovy.

3. Creación de closures (II)

Crear un closure para comprobar si un número pasado por parámetro es primo o no. Posteriormente crear un mapa o una lista de números para comprobar cuales de esos son primos y cuales no. El archivo debe llamarse ej33.groovy.

4. Creación de closures (III)

Crear dos closures llamados `ayer` y `mañana` que devuelvan las fechas correspondientes a los días anterior y posterior a la fecha pasada. Podemos crear fechas mediante la sentencia `new Date().parse("d/M/yyyy H:m:s","28/6/2008 00:30:20")`.

Posteriormente, crear una lista de fechas y utilizar los closures recién creados para obtener las fechas del día anterior y posterior a todos los elementos del mapa. El archivo debe llamarse ej34.groovy.

5. Clase completa en Groovy

Crear una clase llamada *Calculadora* que permita la realización de cálculos matemáticos con dos operadores. Estos cálculos serán la suma, resta, multiplicación y división.

Los parámetros serán solicitados como entrada a la aplicación. Para leer parámetros por línea de comandos se pueda utilizar el siguiente closure:

```
System.in.withReader {
        print 'Introduzca operador: '
        op1 = it.readLine()
        println(op1)
}
```

El sistema nos debe pedir que introduzcamos los operandos así como el tipo de operación que queremos realizar y mostrará por pantalla el resultado de la operación. El archivo debe llamarse ej35.groovy.