

EJERCICIOS SINCRONIZACIÓN DE HILOS

EJERCICIO 1: Desarrolla un juego multijugador simple donde varios jugadores intentan recoger monedas de un cofre compartido. Cada jugador es representado por un hilo, y el cofre tiene un número limitado de monedas (10).

Cuando un jugador recoge una moneda, se debe asegurar que otro jugador no pueda tomar la misma moneda al mismo tiempo. Para simular el proceso de recoger moneda puedes usar el siguiente código:

```
long startTime = System.nanoTime();

while (System.nanoTime() - startTime < 500_000_000)

{

    double x = Math.sqrt(Math.random());

}
```

Cuando las monedas se acaben, se mostrará un mensaje por consola indicándolo y finalizará el programa.

EJERCICIO 2: Necesitamos obtener el hash de una serie de ficheros. Como es una operación lenta hay que emplear threads.

El algoritmo para obtener el hash de un fichero llamado prueba.txt es el siguiente:

```
MessageDigest manejadorHash = null;
FileInputStream manejadorFichero = null;
File ficheroAHashear=
    new File("C:\\ficheros\\prueba.txt");
try {
    manejadorHash = MessageDigest.getInstance("SHA-256");

    manejadorFichero = new FileInputStream(ficheroAHashear);
    byte[] dataBytes = new byte[1024];
    int numBytesLeidos;
    while ((numBytesLeidos = manejadorFichero.read(dataBytes)) != -1) {
        manejadorHash.update(dataBytes, 0, numBytesLeidos);
    }
    byte[] hashComoBytes = manejadorHash.digest();
    // Convert the byte array to a hexadecimal string
    StringBuilder hashComoString = new StringBuilder();
    for (byte unByte : hashComoBytes) {
        hashComoString.append(Integer.toString((unByte & 0xff) + 0x100,
```

```

16).substring(1));
    }
    System.out.println( hashComoString);

    } catch (Exception e) {
        System.out.println( "error");
    }
    finally{
        if (manejadorFichero!=null) {
            try {
                manejadorFichero.close();
            } catch (IOException e) {
                System.out.println( "error");
            }
        }
    }
}

```

Elabora para ello:

- una clase Java que contendrá el hilo que realiza el hash y que controle que si se produce una excepción. En caso de que todo vaya bien se guardará en el fichero de resultado la información siguiente: **nombre del fichero - Hash: XX**, donde XX es el valor del hash del fichero. Si se ha producido una excepción se guardará en el fichero de resultado la información siguiente: **nombre del fichero - no se ha podido realizar el hash**.
- una clase Java con el método main que:
 - Accede a la carpeta C:\ficheros y obtiene los nombres de los ficheros que hay en dicha carpeta
 - Se asegure de que los hashes de los ficheros anteriores sean guardados en el fichero resultado.txt
 - El método main en el que te basarás para realizar lo anteriormente indicado será éste:

```

public static void main(String[] args) {
    ProcessBuilder builder = new ProcessBuilder("cmd.exe", "/c", "dir", "c:\\ficheros",
"/B");
    Process process;
    try {
        process = builder.start();
    } catch (IOException e) {
        throw new RuntimeException(e);
    }

    BufferedReader reader = new BufferedReader(new
InputStreamReader(process.getInputStream()));

    String line = " ";

```

```
while (line != null) {  
    try {  
        line = reader.readLine();  
    } catch (IOException e) {  
        throw new RuntimeException(e);  
    }  
    if (line != null) {  
  
    }  
}
```

○