2 Ejercicios:

5 puntos Contador de letras en ficheros

5 puntos Reporta divisores primos

Tiempo de examen 2 horas

Ejercicio 1

El programa recibirá como parámetros de línea de comandos varias rutas de ficheros. El programa arrancará un Thread por cada fichero, leerá el contenido contando cada letra (Da igual si son mayúsculas o minúsculas; ‘a’ y ‘A’ contarán como ‘a’, no tengas en cuenta la ‘ñ’).

El programa principal esperará a que terminen los Threads, sumará el número de letras de cada Thread y lo escribirá por pantalla.

Ejemplo:

java Contador ruta/fichero1.txt ruta/fichero2.txt … runta/ficheroN.txt

El número de letras es:

a: 330

b: 265

…

z: 102

NOTA: Cada Thread recibirá la ruta del fichero en el constructor

5 Puntos

1,0 puntos Recoger la información de línea de comandos y pasarla a los Threads

1,0 puntos Lectura de la información correspondiente en cada Thread, de forma concurrente.

1,0 puntos Cada Thread cuenta las letras de forma concurrente.

1,0 puntos El programa principal espera a los Threads, extrae la información y la mezcla

1,0 puntos Reporta la información correcta, extraída de cada thread ejecutado de forma concurrente.

Ejercicio 2

Este programa recibe por parámetro 2 números. El primero indica el número de trabajo. El segundo indica el número de threads. El primer número debe ser mayor a 30000. El segundo número debe estar entre 2 y 10. En caso contrario el programa emitirá un error y no hará nada más.

Descripción del trabajo

El programa escribirá por pantalla todos aquellos números por los que se puede dividir el número de trabajo y que sean primos (número de trabajo = primer número pasado por parámetro de línea de comando). Ejemplo:

java Reporta 31233 4

Escribe:

1, 3, 29, 359

Ya que 31233 puede ser dividido por estos números y además son primos.

Distribución de trabajo

Para hacer el proceso de verificación de estos números el programa creará N Threads, siendo N el segundo número pasado por parámetro. Para ello el programa principal deberá distribuir el trabajo de forma equitativa. (31233 / 4 = 7808; 31233 % 4 = 1)

- el primer Thread verificará los números del 1 al 7808

- el segundo Thread verificará los números del 7809 al 15616

- el tercero Thread verificará los números del 15617 al 23424

- el cuarto Thread verificará los números del 23424 al final

NOTA: al ser el resto 1 el último hará 1 número más. No importa.

NOTA: Los Threads verificadores obtendrán la información sobre el rango en el que deben trabajar a través de su constructor. Piensa sobre si estos Threads necesitan alguna información más.

5 Puntos

0,5 puntos Verificación de parámetros de entrada

1,0 puntos Distribución del trabajo

1,5 puntos Cálculos correctos de números que cumplen los requisitos

1,0 puntos Reportar por pantalla la información (Correcta y concurrente)

1,0 puntos Tras la ejecución correcta y concurrente, el programa principal espera la finalización de los Threads