PSP - U3 Actividades

Descargar estos apuntes

Índice

> **Nombre del proyecto**
>
> En estas actividades seguimos usando la nomenclatura de clases y proyectos explicada en la hoja de actividades de autoevaluación de la U2

12. Clase psp.actividades.U3A12_Controlador

Haz un programa que calcule

$$\sum_{i=1}^{n} \sqrt{i}$$

El programa controlador recibe como parámetro el valor de n y, de forma opcional, también se le puede pasar la cantidad de operaciones a realizar por cada hilo (por defecto, si no se indica nada, 100).

Va a dividir equitativamente el reparto entre (n/100)+1 hilos.

- Al primer hilo le asignará el cálculo desde 0 hasta 99,
- al 2º desde 100 hasta 199,
- al 3º desde 200 hasta 299,
- ...
- al último le asignará los que queden hasta n.

El programa recogerá el resultado que calcule cada hilo, los irá sumando y, cuando todos hayan acabado, mostrará el resultado final.

```
La suma total de las raíces cuadradas desde 0 hasta 10875 es: 756105.62767
```

> **Recuperar el resultado**
>
> ¿Qué debe hacer el hilo principal para obtener el resultado de cada hilo?

Clase psp.actividades.U3A12_Calculador (hereda de Thread)

Esta clase recibirá como argumentos el número de hilo que es, el valor inferior del cálculo que debe realizar y el valor superior.

Con el número de hilo, se pondrá a sí misma un nombre (Calculador-X)

Comunicación entre hilos

¿Cómo podemos hacer que el hilo devuelva la información al hilo principal?

Calculad el tiempo que tarda en realizar los cálculos variando la cantidad de hilos creados.

Tiempo de ejecución

Para calcular el tiempo que tarda en ejecutarse un proceso / hilo podemos usar el método System.currentTimeMillis() de la siguiente forma

```
long t_comienzo, t_fin;
t_comienzo = System.currentTimeMillis();
// Código que queremos monitorizar
t_fin = System.currentTimeMillis();
System.out.println("El proceso ha tardado " + (t_fin - t_comienzo) + "ms");
```

13. Clase psp.actividades.U3A13_Contador

Implementa un programa que reciba como argumentos una lista de ficheros de texto y que cuente el número de caracteres, palabras y líneas que hay en cada uno.

Para hacerlo, creará un hilo por cada archivo recibido que se encargará de procesar la información del archivo que reciba como parámetro.

Junto con la información obtenida, cada hilo mostrará el tiempo que ha tardado en completar la tarea.

14. Clase psp.actividades.U3A14_Cronometro

Crea una aplicación gráfica con Swing que simule un cronómetro.

En el diseño tendremos en la parte central de la ventana unas etiquetas representando hh:mm:ss

Tendremos un botón `Start / Resume` que comenzará / reanudará la cuenta del cronómetro. El valor mostrado se actualizará cada segundo.

Piensa que no podemos dejar al proceso principal bloqueado, por lo que *alguien* tendrá que encargarse de actualizar el valor cada segundo.

Tendremos un botón `Pause / Stop` que detendrá la cuenta del cronómetro pero mantendrá el valor actual. Si se vuelve a pulsar una segunda vez el botón reseteará el valor del cronómetro al valor inicial 00:00:00

Un flujo de ejemplo del programa podría ser este

> Cronómetro a 00:00:00 → Start / Resume → Empieza cuenta → Pause / Stop →
> Cronómetro a 00:00:37 → Start / Resume → Reanuda cuenta → Pause / Stop →
> Cronómetro a 00:01:05 → Pause / Stop → Cronómetro a 00:00:00