PSP - U3 Actividades auto evaluación

Descargar estos apuntes

Índice

Warning

Para todas las aplicaciones, **las excepciones deben controlarse en el código del programa**. Si no se indica lo contrario, en el bloque catch siempre se mostrará la pila de llamadas usando el método `printStackTrace()` de la excepción capturada junto con la versión traducida del mensaje de error asociado, método `getLocalizedMessage()`.

Nombre del proyecto

Al contrario que en el resto de actividades, en este caso vamos a crear un único proyecto para todas las actividades. Para ello vamos a utilizar una nomenclatura diferente a la que os pedía en la Unidad 2.
Estas actividades son muy sencillas con clases que no tienen mucho contenido. Para evitar tener que crear un proyecto por cada actividad, vamos a crear un proyecto con varias clases, una por cada actividad. De esta forma, podremos ejecutar cada actividad de forma independiente.

- Crea un proyecto con nombre U3Actividades_Autoevaluables
- Crea una/varias nuevas clases dentro del paquete psp.u3. para cada ejercicio.
- Para probar cada actividad puedes seleccionar la clase y pulsar Shfit + F6 (botón derecho sobre la clase > Run file)

Uso de clases anónimas o con lambdas

Si quieres evitar la creación de tantas clases, cuando el código que tenga que ejecutar un hilo sea sencillo, puedes optar por el uso de lambdas, siempre que sea posible.

Ponerle nombre a los hilos

Es importante darle nombre a los threads y mostrar ese nombre en las salidas por consola. De esta forma se puede identificar qué hilo es el que está produciendo esa información.

uso de colores en la consola

Para diferenciar las salidas de los diferentes hilos, siempre que sea posible, podéis consultar el (Anexo I de la Unidad 2).

1. Clase psp.u3.A1_Numeros

Haced un programa que contenga tres hilos en ejecución, dos de ellos realizaran la suma de los números divisibles entre 5 que se encuentran en un rango de 100 al 1000, y el hilo restante, la impresión de los 10 primeros múltiplos de 10.

Hacedlo extendiendo de la clase Thread.

Como hay varios hilos y no todos hacen lo mismo, debemos crear más de una clase para crear los threads. A la primera llámala A1_Numeros1, a la segunda A1_Numeros2, ...y a la clase que se encarga de crear y lanzar todos los hilos es a la que vas a llamar como la actividad A1_Numeros.

2. Clase psp.u3.A2_Caracteres

Haced un programa que contenga dos hilos en ejecución, un hilo imprimirá 5 caracteres aleatorios, y el otro hilo pedirá una cadena, e imprimirá solo las vocales.

Hacedlo extendiendo de la clase Thread.

3. Clase psp.u3.A3_Inversa

Haced un programa que contenga dos hilos en ejecución:

- Uno de ellos realizará la impresión de la transformación de una cadena a su forma inversa. Por ejemplo la cadena hilo se iría imprimiendo: o, ol, oli, olih).
- El otro hilo hará la suma interna de los números impares comprendidos del 500 al 1000, y mostrará únicamente el resultado de la suma.

Hacedlo implementando la interfaz Runnable

4. Clase psp.u3.A4_Divisores

Haced un programa que contenga tres hilos en ejecución que presenten cada uno de estos, la impresión de los números divisibles entre 4, que se encuentran en el rango del 1 al 200

Hacedlo extendiendo de la clase Thread.

5. Clase psp.u3.A5_Argumentos

Haced un programa que reciba como argumento una cadena de texto.

La clase partirá la cadena en palabras (por los espacios en blanco) y creará un hilo para tratar cada palabra. Cada hilo se encargará de imprimir solo las vocales.

Hacedlo implementando la interfaz Runnable

> Cuando una clase recibe argumentos, estos le llegan en el parámetro args del main, un array de Strings que contiene todos los parámetros que se le han pasado.

6. Clase psp.u3.A6_Sumatorio

Haced un programa que contenga dos hilos en ejecución, los cuales harán la suma interna de los números pares comprendidos del 100 al 1000, imprimir sólo el resultado de la suma.

Hacedlo implementando la interfaz Runnable

7. Clase psp.u3.A7_ParImpar

Haced un programa que contenga dos hilos en ejecución, uno imprimirá números impares del 1 al 10 y el otro imprimirá números pares del 0 al 10.

Hacedlo heredando de Thread para el primer hilo e implementando la interfaz Runnable para el segundo hilo.

8. Clase psp.u3.A8_Series

Haced un programa que contenga tres hilos en ejecución, cada hilo pedirá al usuario un valor de tipo entero, e imprimirá la suma de la serie $2^1+2^2+2^3+...+2^n$, donde n es el numero que introdujo el usuario

Hacedlo extendiendo de la clase Thread.

9. Clase psp.u3.A9_Factorial

Crea una clase que reciba una cantidad indeterminada de números como argumentos. Crear un hilo por cada número recibido que se encargue de calcular el factorial del dicho número y mostrarlo.

> n! = n * (n-1) * (n-2) * (n-3) * ... * 1;

Hacedlo implementando la interfaz Runnable