**PROGRAMACIÓN AVANZADA**

**Alumno:** Máximo Abel Paute Jumbo

**Fecha:** 8 de julio del 2020

**HILOS**

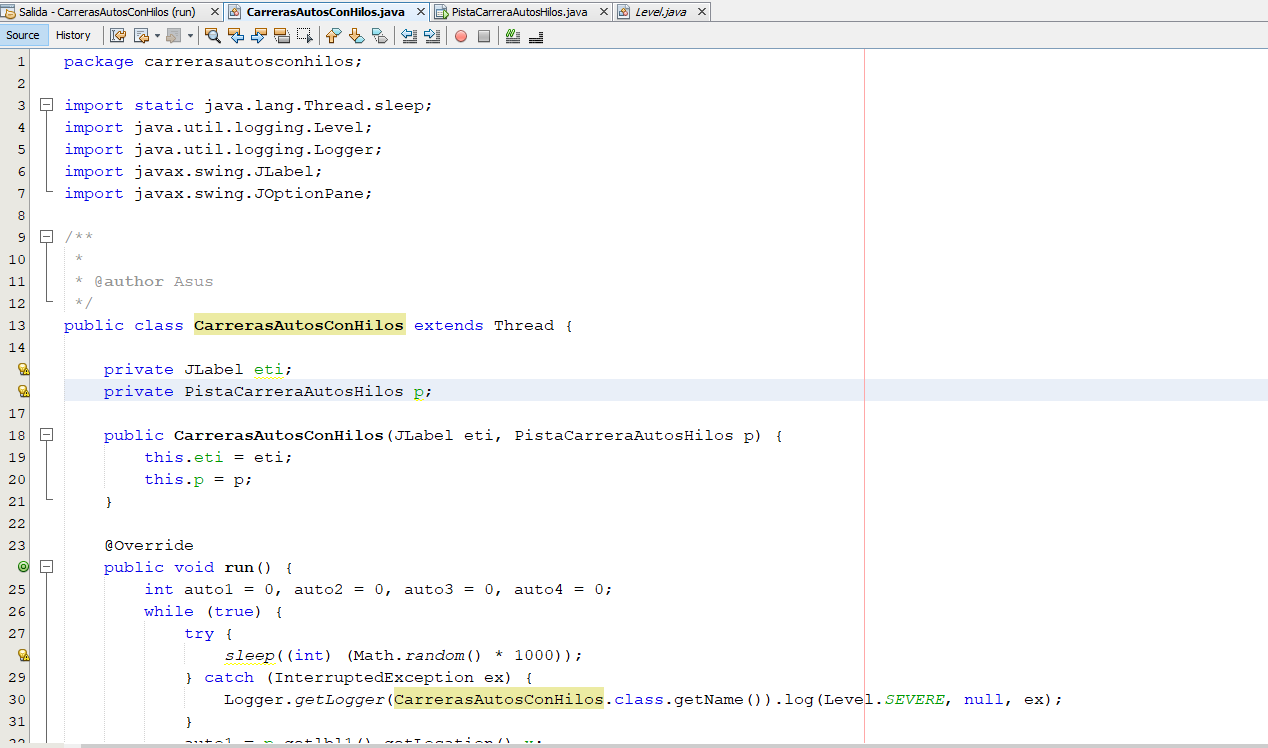
La forma más directa para hacer un programa multihilo es extender la clase Thread, y redefinir el método run(). Este método es invocado cuando se inicia el hilo (mediante una llamada al método start() de la clase Thread). El hilo se inicia con la llamada al método run() y termina cuando termina éste.

**Simulación de carrera de autos en pista**

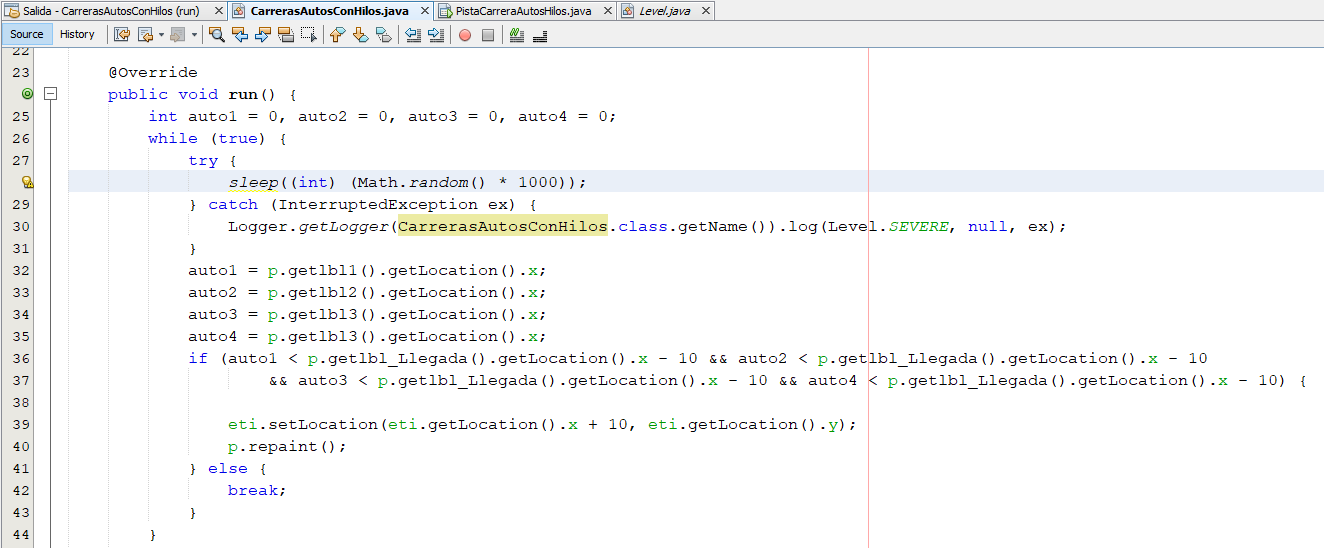
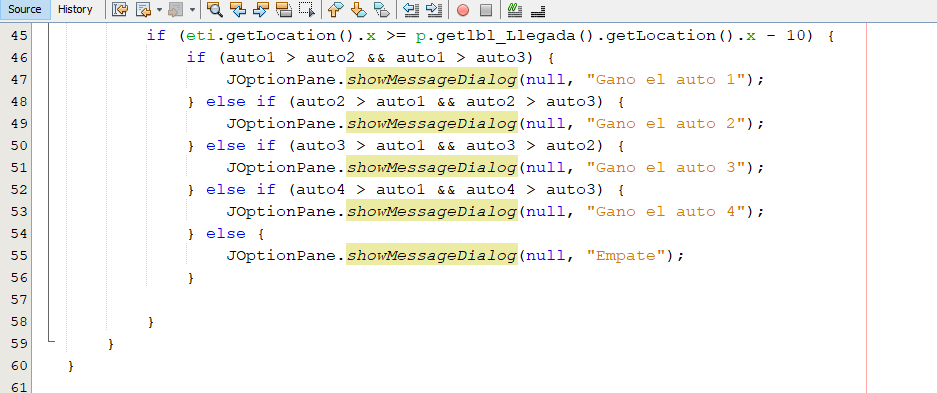
* La simulación consiste la competencia de 4 carros ejecutados a la misma vez, mediante hilos, obteniendo ganador o empate.

**Proceso**

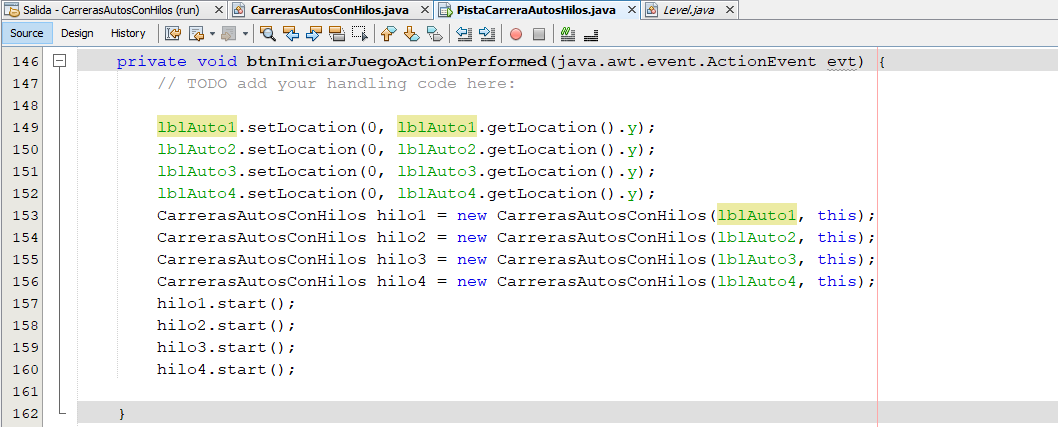
1. Clase CarreraAutosConHilos

Se declara dos atributos de tipo JLabel y PístaCarrerasAutosHilos

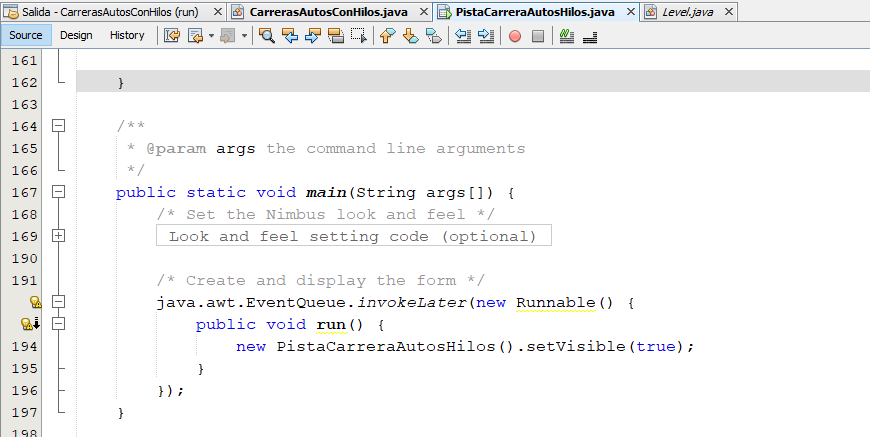
1. **Metodo run():** El hilo se inicia con la llamada al método run() y termina cuando termina éste.

* Se declara variables para los carros inicializadas en 0.
* Para calcular el tiempo que tarda en ejecutar cada hilo utilizamos el método **Math.random**, multiplicado por 1000 que es el timpo que representa 1s en milisegundos.
* Se utiliza **getLocation()** Se utiliza para ubicar en el jframe cada carrito
* **repaint():** Metodo para avisar a la máquina virtual que ese componente necesita repintado. El método en si mismo no borra ni dibuja nada.
* En el if cada auto se va posicionando hacia la derecha de 10 en el eje x, se va acercando a la meta.

1. **Botón Iniciar**
   * Se crea una instancia de la clase CarreraAutosConHilos con new, luego llamamos al método start() de la thread para hacer que ejecute el método run().



1. **Main**
   * En este caso solo será necesario implementar el método "***run()***" para que los procesos implementados en ese método se ejecuten en un hilo diferente.



1. **Resultados**



La carrera finaliza con un ganador o un empate, demostrando que se puede realizar varios procesos utilizando hilos que básicamente son multitarea ayudando a optimizar tiempo de ejecución del programa.