

Práctica 1 - PCTR

Francisco Javier Molina Rojas

10 de Octubre de 2023

1 Metodo `Math.random()`

Método **estático** (compartido en memoria por todos los hilos). Sin embargo, esta adecuadamente sincronizado para que no cause problema al usarlo en diferentes hilos.

- La sincronización se realiza con una cola de llegada de hilos.
- Esta condición causa que haya mucha contención.
- Si hubiesen muchos hilos, esto sería ineficiente.

2 Clase `Random` de `java.util`

Genera números aleatorios de periodo 2^{48} . Posee dos tipos de constructores (con o sin semilla — `Random()` o `Random(long seed)`). En este caso tendríamos que usar un objeto de la clase para cada hilo.

- Ya no poseemos contención.
- Ocupan mas memoria (en principio, no es un problema).
- El único problema a solucionar es que deben tener diferente semilla.
- Puedes cambiar de semilla usando el metodo `setSeed(long seed)`.

3 Clase `ThreadLocalRandom`

Genera números aleatorios de periodo 2^{64} . Se puede usar la clase desde el método `run()` de cualquier hilo. Hay que tener en cuenta que la semilla se genera automáticamente, y **NO se podra cambiar**.

- Ya no poseemos contención.
- No ocupan más memoria que la necesaria para lanzar los hilos.
- El único problema es que la semilla del generador permanecerá constante.
- Evidentemente, los hilos son necesarios para usarlo.

4 Clase Scanner

La usaremos como método de entrada de datos. Para crear una instancia (u objeto) de la clase Scanner haremos lo siguiente:

"Scanner s = new Scanner(System.in);", dándole el control del flujo de entrada a la instancia Scanner s. Posteriormente podemos usar los siguientes métodos:

- next() Nos permite obtener un objeto String desde el flujo de entrada.
- nextDouble() lo mismo pero con Double.
- nextFloat() lo mismo pero con Float.
- nextInt() lo mismo pero con Int.
- nextLong() lo mismo pero con Long.
- nextBoolean() lo mismo pero con Boolean.

Aún existen más metodos, pero los más importantes (y los que más pinta tienen que usaremos) son estos.

5 Argumentos por línea de comando

Cuando vayas ejecutar un programa .java (que tenga definido un main) puedes pasarle por la línea de comandos distintos argumentos. Para que puedas probarlo, crea un fichero test.java con el siguiente código:

```
1 public class test{
2     public static void main(String[] args)
3     throws Exception{
4         int argumento1 = Integer.parseInt(args[0]);
5         String argumento2 = args[1];
6         System.out.println(argumento1 + " " + argumento2
7             );
8     }
```

Listing 1: test.java

Compilalo y ejecutalo con el siguiente comando:
"java test 10 javivi"

Podras ver como lo que incluimos en la llamada a la ejecución por la línea de comandos se nos añade a cada una de las posiciones del vector args que recibimos en el main como argumento. Nótese que los argumentos SIEMPRE seran Strings, por lo que si queremos introducir un número, deberemos "parsearlo" para poder usarlo correctamente.