Taller 1B

Parte 1: Incremento de un Contador

**Ejemplo 1: Aplicación monothread para el incremento de un Contador**

Efectivamente, la aplicación regresa el valor esperado el cual es 1000x1000 ya que simplemente aumenta el contador que tiene de atributo.

**Ejemplo 2: Aplicación multithread para el incremento de un contador**

**2. ¿Al ejecutar el programa, el resultado corresponde al valor esperado? Explique.**

No, no retorna el valor esperado ya que ejecuta 1000 threads diferentes, cada uno con su propio contador (Que inicia en 0). De esta manera, sólo tendríamos 1000 threads con sus contadores en 1000.

**3. Ejecute cinco veces el programa y escriba el resultado obtenido en cada ejecución.**

Ejecutando 1 vez: 1000

Ejecutando 2 veces: 2000

Ejecutando 3 veces: 3000

Ejecutando 4 veces: 4000

Ejecutando 5 veces: 5000

**4. ¿Hay acceso concurrente a alguna variable compartida? Si es así, diga en dónde**

Yo diría que no, ya que sólo se instancian 1000 objetos diferentes de la clase ContadorThreads, cada uno con su variable diferente. Entonces no.

**Parte 2: Elemento mayor en una matriz de enteros**

**Ejemplo 3: Aplicación multithread para encontrar el elemento mayor de una matriz de enteros**

No funcionaría, ya que imprimiría varias veces el mayor número de la fila de la matriz. En ese caso, en la primera ejecución imprime el valor más grande de la primera columna, en la segunda imprime el valor más grande de la segunda y así sucesivamente.

**¿Hay acceso concurrente a alguna variable compartida? Si es así, diga en dónde.**

Yo diría que no, ya que cada vez que se instancia un nuevo MaximoMatriz, esta inicia con todas sus variables en el valor original.

**¿Puede obtener alguna conclusión?**

A mi manera de entender los threads, creo que con cuando uno crea varios objetos diferentes estos NO son afectados entre si. Simplemente, cada uno se afecta a sí mismo ya que no comparten variables ni sus atributos.