

## Taller 1 Parte B.

Juan David Bautista Parra

### 1. ¿Al ejecutar el programa, el resultado corresponde al valor esperado?

Se ha logrado el valor esperado por el contador valor esperado = 1000000.

```
1 package TalleresResueltos;
2
3 public class Taller1ParteB
4 {
5     private int contador = 0;
6
7     public void incrementar()
8     {
9         for(int i=0; i<1000;i++)
10         {
11             contador++;
12         }
13     }
14
15     public int getContador()
16     {
17         return contador;
18     }
19
20     public static void main(String[] args) {
21         Taller1ParteB t = new Taller1ParteB();
22
23         for (int i = 0; i<1000;i++)
24         {
25             t.incrementar();
26         }
27         System.out.println(t.getContador());
28     }
29 }
30
```

Problems Javadoc Declaration Console x

<terminated> Taller1ParteB [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (30)

1000000

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

A medida que itero genero un nuevo thread. Y la máquina ejecuta tan rápido el for y la misma variable es modificada por varios threads al tiempo.

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

8426637  
8470005  
8602904  
8972429  
8791285

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

Si el acceso a la variable nombrada “contador” es modificada a medida que la los threads se han inicializando.

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

| EJECUCIÓN | VALOR OBTENIDO | VALOR ESPERADO |
|-----------|----------------|----------------|
| 1         | 86109          | 101047         |
| 2         | 67426          | 98704          |
| 3         | 17170          | 48181          |
| 4         | 68046          | 92562          |
| 5         | 74748          | 74748          |

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

Si el acceso concurrente está en las variables compartidas de Máximo local y máximo global.

**7. Conclusión.**

Según este laboratorio podemos concluir que para el uso de threads que se ejecuten simultáneamente, debe existir una precaución en la sobre-escritura de variables compartidas. Ya que esto genera que el programa no pueda ejecutarse efectivamente.