**TALLER 1**

**1A.**

1. Complete la siguiente tabla, con respecto a la creación de threads usando la extensión de la clase **Thread** y la implementación de la interface **Runnable**.

|  |  |
| --- | --- |
| **Se parecen** | **Se diferencian** |
| * Manejan los mismos métodos y funcionan para implementar soluciones con más de un hilo. | * **Al implementar Runnable, muchos subprocesos pueden compartir el mismo objeto de subproceso mientras que al extender la clase Thread cada subproceso tiene un único objeto asociado.** |

**1B.**

**PARTE 1**

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

R/ Si, el programa deja la variable contador en 10.000.000

1. **¿Al ejecutar el programa, el resultado corresponde al valor esperado? Explique.**  
   R/ No, el resultado del código en todas las ejecuciones es ligeramente menor al resultado esperado. La discrepancia se debe a que varios threads están aumentando concurrentemente el valor del contador, se está entrelazando la ejecución de operaciones de cada thread.
2. **Ejecute cinco veces el programa y escriba el resultado obtenido en cada ejecución**

|  |  |
| --- | --- |
| **EJECUCION** | **VALOR OBTENIDO** |
| **1** | 9.862.834 |
| **2** | 9.876.672 |
| **3** | 9.591.680 |
| **4** | 9.737.912 |
| **5** | 9.658.210 |

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

R/ Si hay acceso concurrente a la variable contador, en el for donde se crea cada thread y se le da start.

**PARTE 2**

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

|  |  |  |
| --- | --- | --- |
| **EJECUCION** | **VALOR OBTENIDO** | **VALOR ESPERADO** |
| **1** | 55.151 | 86.159 |
| **2** | 47.673 | 103.548 |
| **3** | 96.483 | 96.483 |
| **4** | 63.425 | 101.605 |
| **5** | 67.216 | 74.988 |

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

R/ Si, en la variable valor global.

1. **¿Puede obtener alguna conclusión?**

R/ El programa se queda con el máximo global de un thread. Si otro thread encuentra un valor superior, no se actualiza. Puede que el máximo que encuentre el thread sea el máximo global o puede que no lo sea, es random.