Juliana Velasco

201822088

Taller 1

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
En ambas se invoca el método start() ya que con el método new() solo se crea la estructura para representar el Thread. Solo se puede dar un único start() a un Thread.	En una se implementa una interfaz (Runnable), mientras que en la otra se extiende una clase (Thread). Implementación de Runnable: En el main se crea un objeto tipo Thread. Extensión de la clase Thread: En el main se crea un objeto de la misma clase.

Taller 1b

1. Al ejecutar el programa el resultado sí es el esperado.
2. Al ejecutar el programa el resultado no es el esperado, esto ocurre ya que los Threads no cumplen el método FIFO. Es decir, ya que hay múltiples Threads puede que algunos terminen antes que otros, resultando en que la variable compartida tome el resultado erróneo, lo que genera que el programa arroje un número menor a 10000000.
3.

Ejecución	Valor obtenido
1	9941046
2	9980816
3	9980177
4	9962947
5	9989920

4. Sí, hay acceso concurrente a la variable compartida “contador” (variable tipo int estática). Esto es evidente en el aumento de “contador” en el método run ().

1.

Ejecución	Valor obtenido
99399	98840
96851	93262
105000	105000
94763	87649
88517	87684

2. Sí, hay acceso concurrente a una variable compartida. En este caso es la variable “mayor” (variable tipo int estática), a la cual se accede en el método run ().

3. En este método se puede concluir que, debido a que se utilizan múltiples Threads, es posible que se imprima el resultado erróneo teniendo en cuenta que estos comparten una variable y no siguen un orden secuencial (FIFO), lo que resulta en modificaciones sin un orden que no aseguran el éxito del programa.