

Taller 1

Sara Calle Lopez

201820801

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 el run solo se imprime - En ambas implementaciones de invoca el método start. - Solamente se puede invocar un método start por cada objeto. - Con el new solo se crea la estructura y no se activa el thread.	-En la extensión Thread se crea un objeto de la misma clase, mientras que en la implementación de la interface Runnable se crea un objeto de tipo Thread. - En una se implementa una interface runnable mientras que la otra extiende de la clase Thread.

Taller 1B

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

- Si, el valor si corresponde con el esperado. Pues el valor es de 10 millones.

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

- Si, el valor si corresponde con el esperado. Pues el valor es de 10 millones.

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

No, el resultado no corresponde con el esperado ya que se crean varias líneas de ejecución (Threads). Por esto, al acabar unos threads antes que otros, el resultado se imprime parcialmente y no alcanza a llegar a los 10 millones. Esto debido a que los threads no van al tiempo y no están en el orden.

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

Ejecución	Valor obtenido
1	9,972,710
2	9,968,293
3	9,990,000
4	9,966,961
5	9,962,223

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

Si, en la variable contador es recurrente en este caso, pues es consultada varias veces en la ejecución del programa en el método run.

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

Ejecución	Valor obtenido	Valor esperado
1	81290	86335
2	82686	88265
3	85258	97534
4	93354	96556
5	77242	77242

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

Si, hay acceso concurrente a la variable mayor. Pues esta al tener en su declaración la sentencia static, es compartida por varios threads en el método run.

¿Puede obtener alguna conclusión?

Si, cuando se utilizar un monothread el resultado de una ejecución con variables concurrentes es correcta. Sin embargo, cuando se utilizan varios threads con variables compartidas, estos toman diferentes líneas de ejecución que afectan el valor de la variable compartida e imprimiendo así un número erróneo.