

Laboratorio 1 – Infraestructura Computacional

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:

- Ambas implementaciones cumplen el mismo fin, crear un nuevo Thread e imprimir un mensaje por consola
- Ambas implementaciones utilizan tanto la clase Thread y la interfaz Runnable, solo que de manera diferente.
- En ambas se llama el método start de un objeto de la clase Thread
- En ambas, lo que hace el Thread es ejecutar lo que se encuentre dentro del método run()

Se diferencian:

- Cada una utiliza una implementación diferente del constructor de la clase Thread. La primera, el constructor sin parámetros, y la segunda el constructor que recibe como parámetro un objeto que implemente la interfaz Runnable
- En la primera implementación, la clase que creamos hereda directamente de la clase Thread, mientras que en la segunda, nuestra clase implementa la interfaz Runnable, que es la interfaz que a la vez implementa la clase Thread

Nota: En el repositorio, la clase Taller1_Ext_Thread corresponde a la implementación del ejercicio heredando de la clase Thread, y la clase Taller1_Runnable corresponde a la implementación implementando la interfaz Runnable

Taller 1b

Parte 1: Incremento de un contador:

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

Rta: Sí, el resultado es 10000000, que es el resultado de multiplicar 1000×10000

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

Rta: No, en cada ejecución el resultado es diferente, ya que los threads no se encuentran sincronizados, y el Sysout se ejecuta antes de que todos los threads hayan ejecutado su método run()

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

Ejecución	Valor obtenido
1	9409966
2	9760000
3	9320000
4	9816176
5	9235284

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

Rta: Sí, a la variable contador. Ésta es compartida ya que en su declaración está "static".

Parte 2: Elemento mayor en una matriz de enteros:

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

Ejecución	Valor obtenido	Valor esperado
1	41961	104067
2	66035	95412
3	63665	104576
4	88080	94491
5	48169	97623

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

Rta: Sí, la variable "mayor".

3. ¿Puede obtener alguna conclusión?

Rta: Al hacer que todos los threads puedan modificar una variable compartida en cualquier momento, un thread puede hacer que su mayor local sea ahora el mayor global, incluso si esto es un error. Para solucionar esto, la acción de verificación del mayor y el cambio por el mayor local debe ser una acción sincronizada.