*Ingeniería de Sistemas y Computación*

*ISIS 2203 Infraestructura Computacional*

Daniel Londoño 201821363

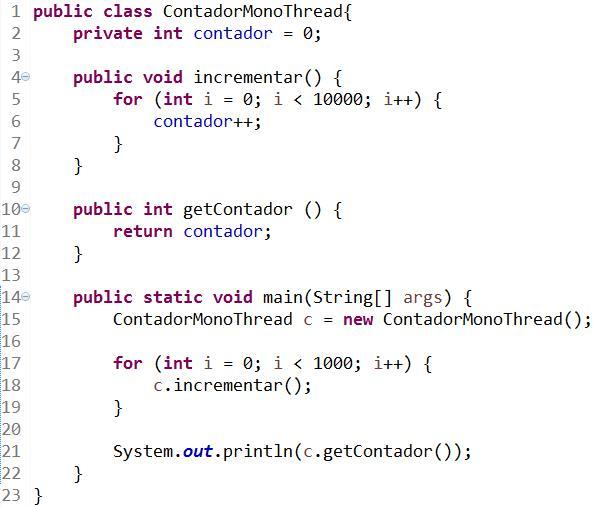
**Taller 1b – Manejo de Threads**

El propósito de este taller es entender la forma como se manejan los threads para implementar aplicaciones concurrentes en Java, e identificar la necesidad sincronización para controlar el acceso concurrente a variables compartidas. El taller tiene dos partes. En la primera parte se va a incrementar un contador un número determinado de veces utilizando dos programas: monothread y multithread. En la segunda parte se seleccionará el mayor de los elementos de una matriz de enteros iniciada al azar.

**Parte 1: Incremento de un contador**

**Ejemplo 1: Aplicación monothread para el incremento de un contador**

El ejemplo a continuación muestra cómo manipular un contador en una aplicación monothread. El ejemplo consiste en llamar 1000 veces un método que incrementa 10000 veces un contador. Este programa es realizado utilizando únicamente el thread principal de la aplicación.



**Responda:**

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

El resultado obtenido es 10.000.000. Este es es resultado esperado ya que el método debería incrementar el contador en 10000 por cada iteración. Incrementar es llamado 1000 veces, i.e 10000\*1000 = 10000000

***Carlos Eduardo Gómez Montoya***

***Jairo Emilio Bautista Mora***

***Geovanny Andrés González*** 1

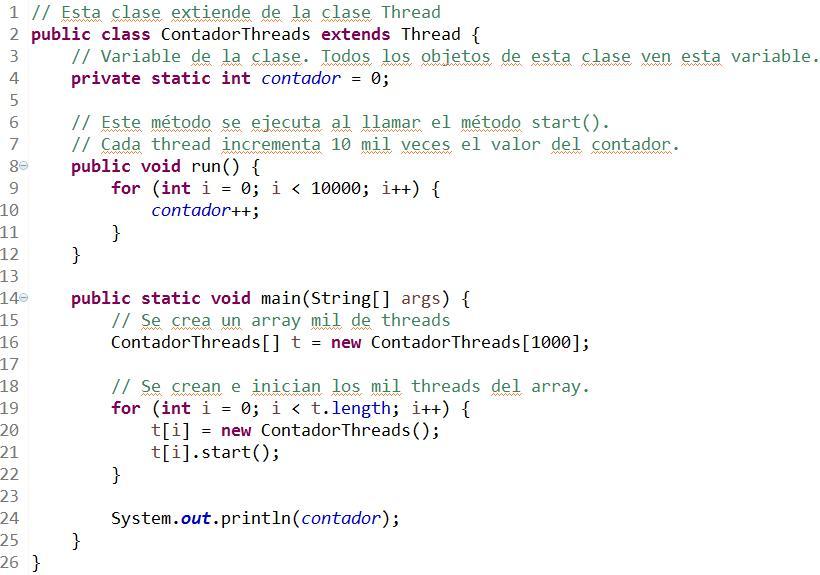
*Ingeniería de Sistemas y Computación*

*ISIS 2203 Infraestructura Computacional*



**Ejemplo 2: Aplicación multithread para el incremento de un contador**

El ejemplo a continuación muestra un ejemplo de una aplicación multithread para la manipulación de un contador. El ejemplo consiste en crear 1000 threads que al ejecutarse, incremente 10000 veces un contador.



**Responda:**

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

El resultado obtenido no corresponde con el esperado. Esto es debido a que mientras el programa esta ocupandose de un Thread, la variable contador (que cabe resaltar que es un atributo de la clase y no de una instancia de ella) está siendo modificada por procesos adyacentes.

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



|  |  |
| --- | --- |
| Ejecución | Resultado |
| 1 | 9978933 |
| 2 | 9992898 |
| 3 | 9990000 |
| 4 | 9972152 |
| 5 | 9925577 |

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

Si. La variable contador, al tener el prefijo static, hace referencia a que el atributo contador, es parte de la clase, más no de una instancia de ella, es decir, de un objeto de clase ContadorThreads. La variable compartida en este caso sería el contador.

***Carlos Eduardo Gómez Montoya***

***Jairo Emilio Bautista Mora***

***Geovanny Andrés González*** 2

*Ingeniería de Sistemas y Computación*

*ISIS 2203 Infraestructura Computacional*



**Parte 2: Elemento mayor en una matriz de enteros**

**Ejemplo 3: Aplicación multithread para encontrar el elemento mayor de una matriz de enteros**

El ejemplo a continuación muestra cómo utilizar threads para que de manera concurrente se pueda encontrar el mayor de los elementos de una matriz de enteros.



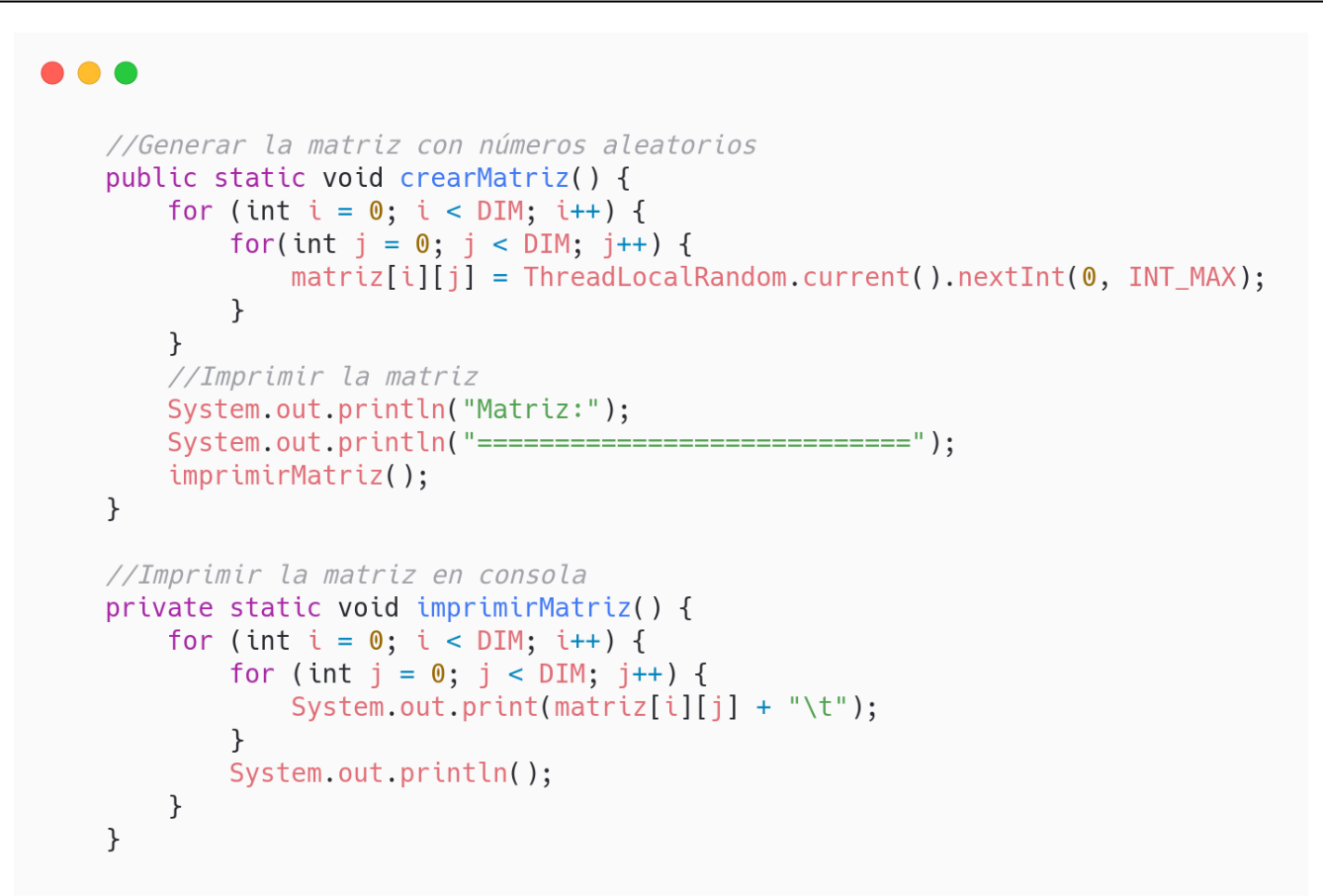
***Carlos Eduardo Gómez Montoya***

***Jairo Emilio Bautista Mora***

***Geovanny Andrés González*** 3

*Ingeniería de Sistemas y Computación*

*ISIS 2203 Infraestructura Computacional*



***Carlos Eduardo Gómez Montoya***

***Jairo Emilio Bautista Mora***

***Geovanny Andrés González*** 4

*Ingeniería de Sistemas y Computación*

*ISIS 2203 Infraestructura Computacional*



***Carlos Eduardo Gómez Montoya***

***Jairo Emilio Bautista Mora***

***Geovanny Andrés González*** 5

*Ingeniería de Sistemas y Computación*

*ISIS 2203 Infraestructura Computacional*



**Responda:**

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



**Ejecución Valor obtenido** **Valor esperado**

1

2

3

4

5



|  |  |  |
| --- | --- | --- |
| Ejecución | Valor | Valor esperado |
| 1 | 102433 | 102433 |
| 2 | 88482 | 95878 |
| 3 | 103872 | 103872 |
| 4 | 68613 | 97558 |
| 5 | 98265 | 98265 |

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

Si. La variable *mayor* al tener el prefijo static es un atributo de la clase más no de una instancia de la misma. Es decir que todos los procesos que usan esas variable la estan compartiendo. Esta está presente justo después de la inicialización de *matriz* en las líneas de código.

1. ¿Puede obtener alguna conclusión?

El hecho de que múltiples procesos tengan acceso, o en su defecto dependan y simultaneamente operen sobre una variable compartida por la clase, es decir, que tenga el atributo static, conlleva a errores en el hardware que afectan posteriormente al software como tal.

***Carlos Eduardo Gómez Montoya***

***Jairo Emilio Bautista Mora***

***Geovanny Andrés González*** 6