**María Camila Terán**

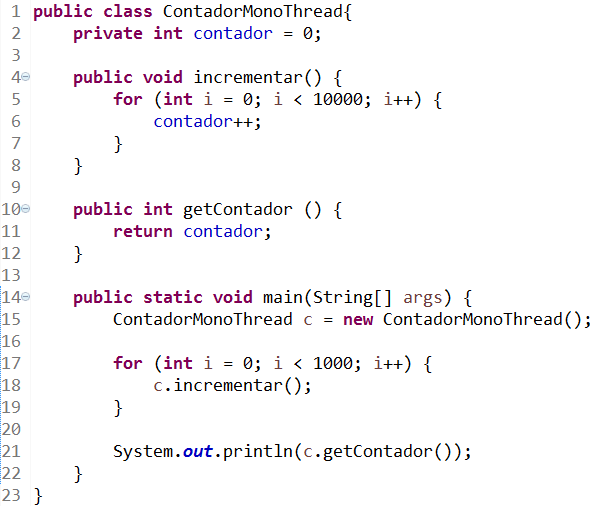
**Taller 1b – Manejo de Threads**

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.



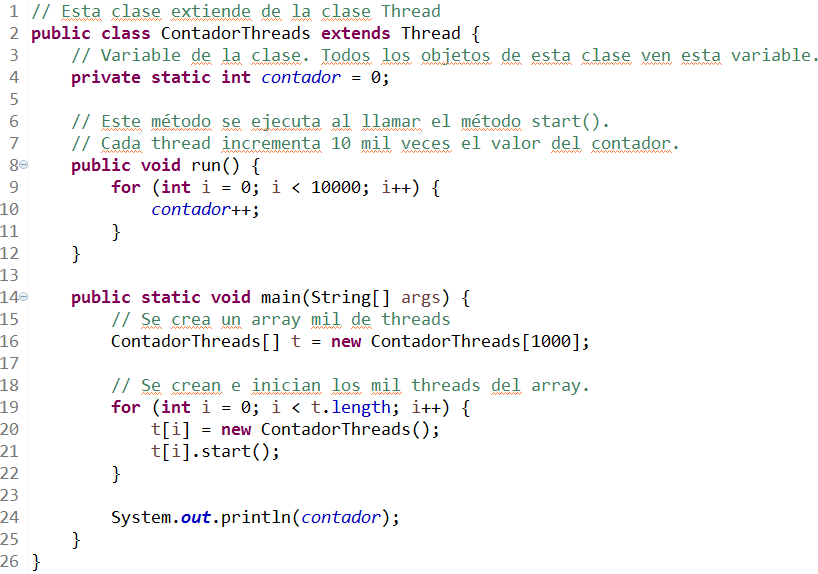
## Responda:

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

El resultado no es el esperado pues se espera que sea 10,000,000 y el resultado es 1,000,000.

## 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.

Efectivamente el valor es esperado debido a que, a diferencia del anterior, se usan hilos de ejecución lo que permite ejecutar el proceso varias veces dependiendo del valor del contador.

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

|  |  |
| --- | --- |
| **Ejecución** | **Valor obtenido** |
| 1 | 9580329 |
| 2 | 9604943 |
| 3 | 9568024 |
| 4 | 9627234 |
| 5 | 9653294 |

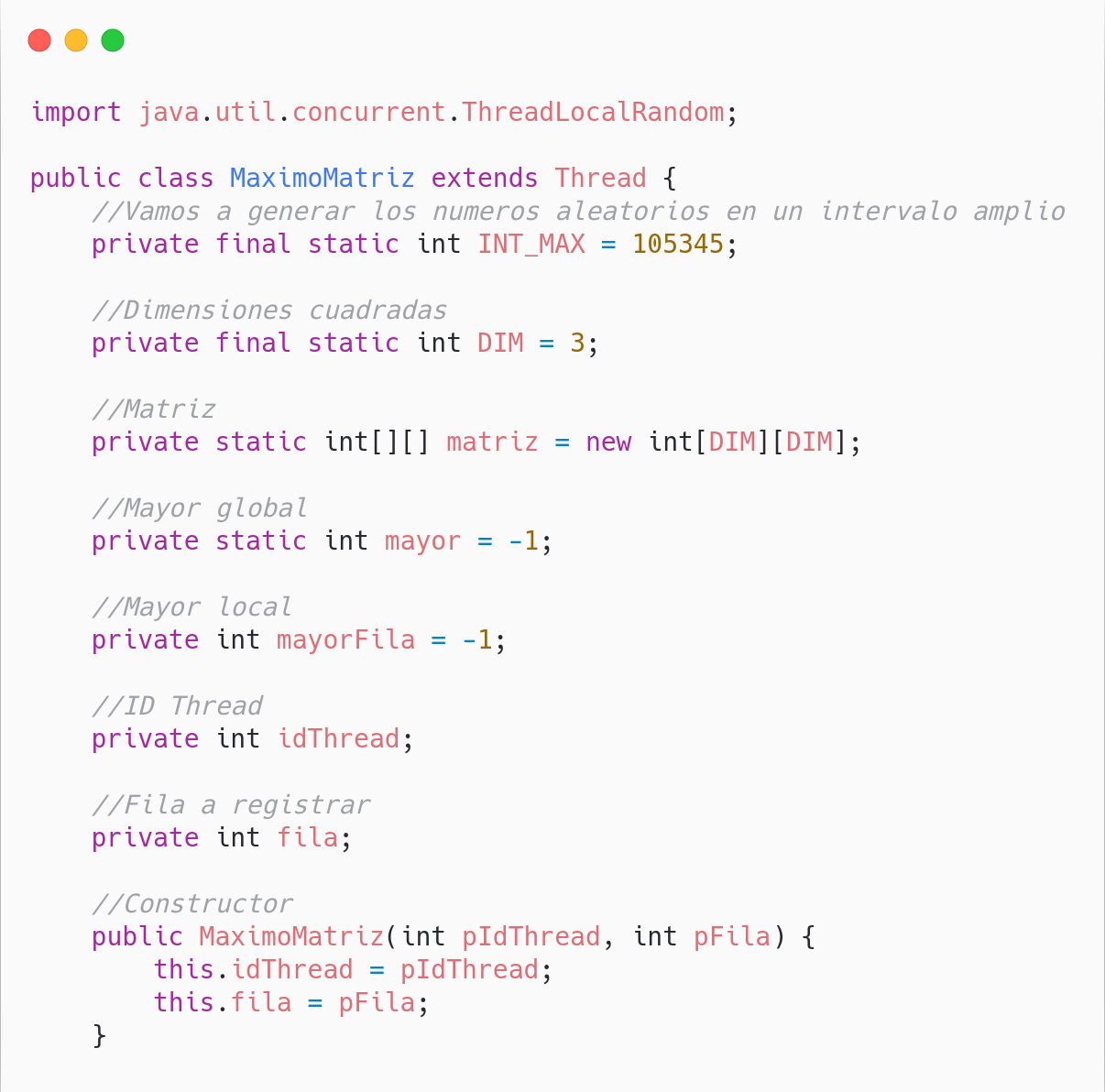
1. ¿Hay acceso concurrente a alguna variable compartida? Si es así, diga en dónde.

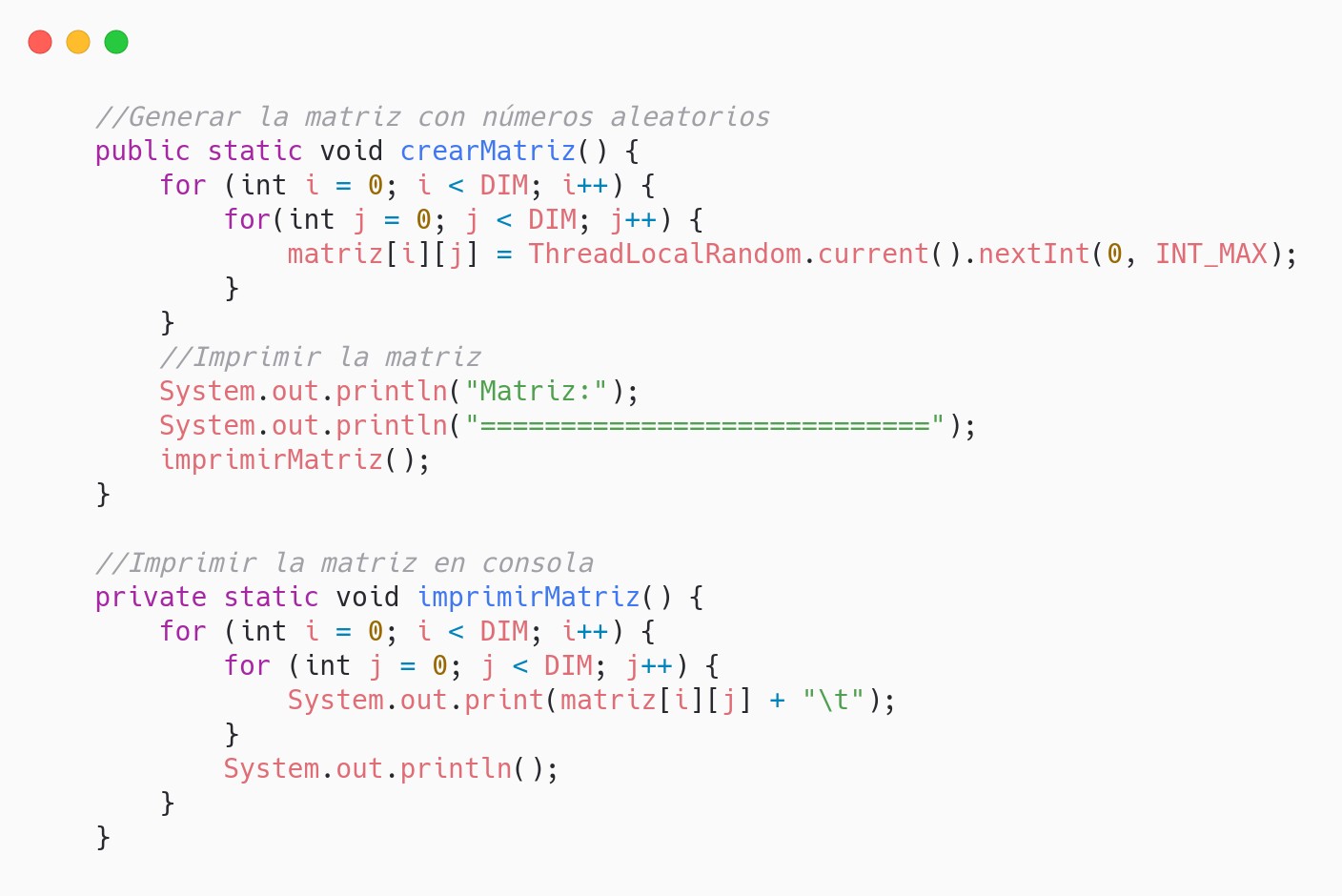
Si hay acceso concurrente al contador.

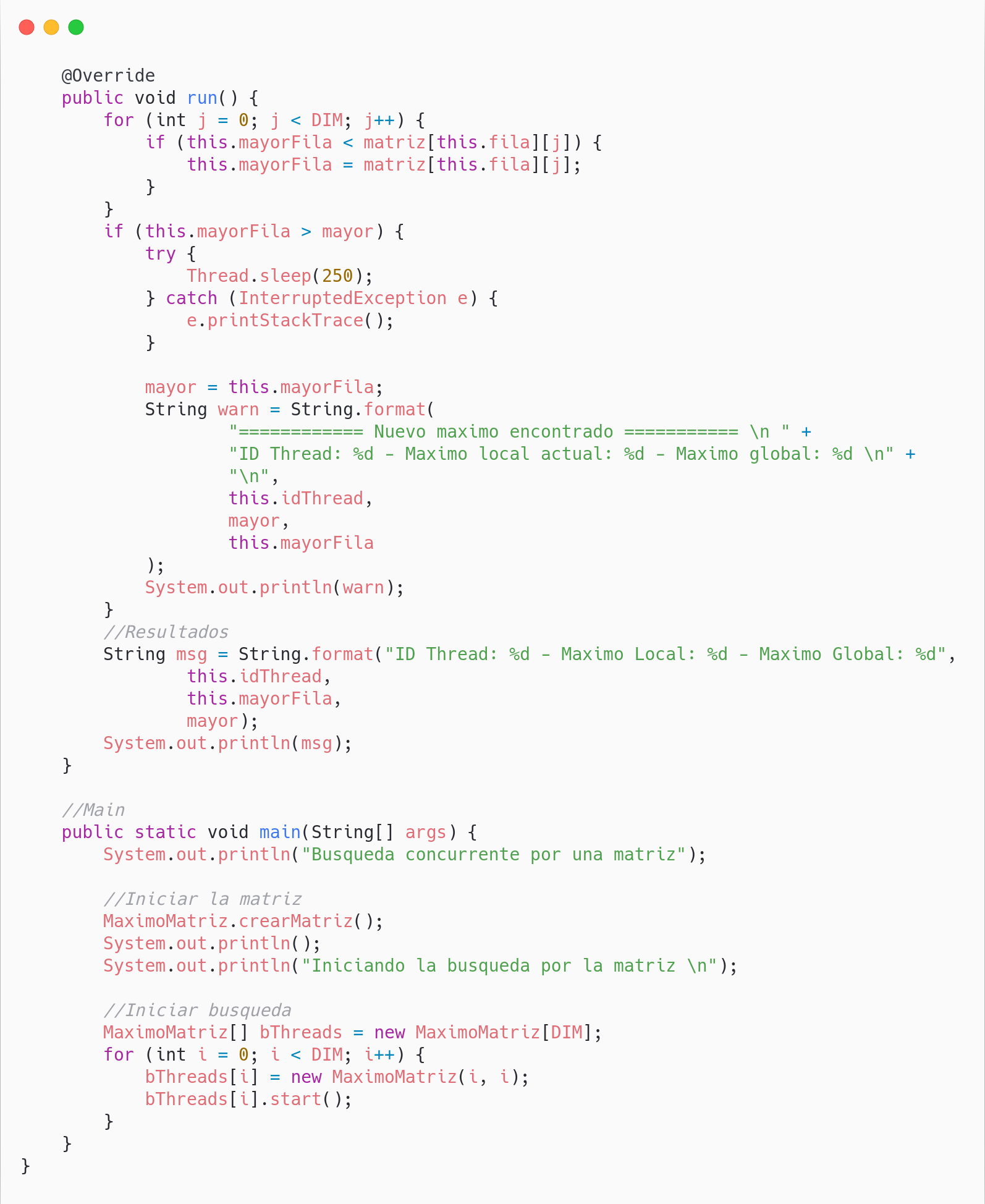
# Parte 2: Elemento mayor en una matriz de enteros

## 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.







## Responda:

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

|  |  |  |
| --- | --- | --- |
| **Ejecución** | **Valor obtenido** | **Valor esperado** |
| 1 | 61263 | 95373 |
| 2 | 84745 | 84745 |
| 3 | 78324 | 95964 |
| 4 | 61234 | 66384 |
| 5 | 91340 | 9134 |

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

Efectivamente la variable compartida se refiere a mayor pues se puede acceder a ella desde cualquier punto del programa.

1. ¿Puede obtener alguna conclusión?

En cuanto a una conclusión se puede afirmar que el máximo actual no coincide en varias ocasiones con el máximo global pero se están realizando comparaciones en paralelo por lo que se genera un “multi thread”. Cada punto de ejecución tenía su máximo global y su máximo actual y multi thread puede ser más rápido que “single thread”.