CTU training solutions

0861 100 395 | www.ctutraining.co.za | enquiry@ctutraining.co.za

Gabriella Rakgotsoka

20232605

JD522_Formative 3 **16/05/2024**

Table of Contents

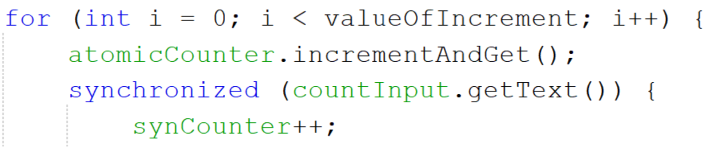
Application Questions

Question 1

Create a Java program that simulates a basic counter. The counter should be incremented by multiple threads. Two (2) threads per implementation.

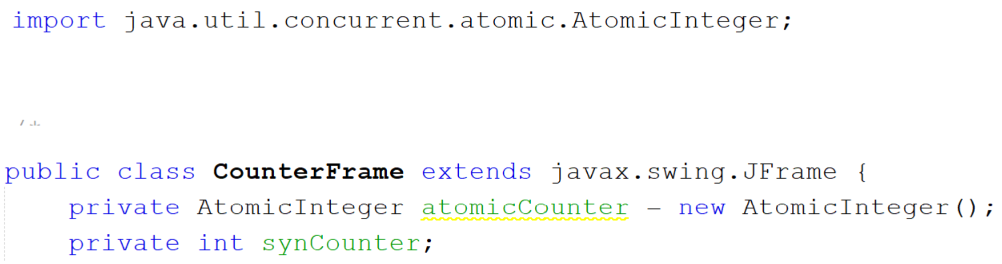
Implementations:

1. Use synchronized keyword to increment the counter and to display the value of the counter.



```
for (int i = 0; i < valueOfIncrement; i++) {
    atomicCounter.incrementAndGet();
    synchronized (countInput.getText()) {
        synCounter++;
```

2. Use AtomicInteger from java.util.concurrent.atomic package to increment the counter and display its value.



```
 import java.util.concurrent.atomic.AtomicInteger;



public class CounterFrame extends javax.swing.JFrame {
    private AtomicInteger atomicCounter = new AtomicInteger();
    private int synCounter;
```

Ensure that the value of the counter is correctly displayed when incremented by multiple threads in both implementations.

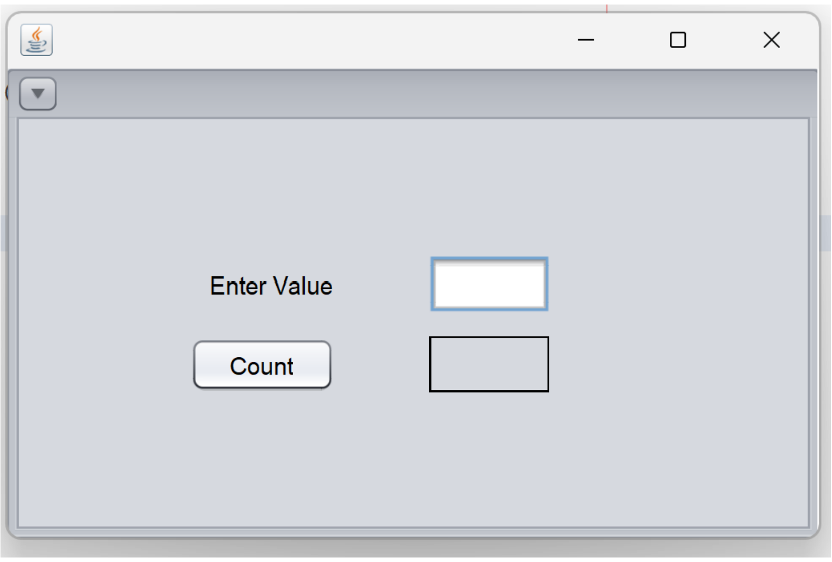
Expected Output:

The output can vary because of thread interleaving, but the final count value must be consistent. For example, if you have two threads and each one increments the counter 1000 times, the final counter value should be 2000.

Design an interface that will prompt the user to insert a value that will be incremented using both threads.

Output

Below is a design of the GUI counter created in java. The counter prompts the user to enter a value in the provided text box and increment the value once the user clicks the count button.


Enter Value
Count

.

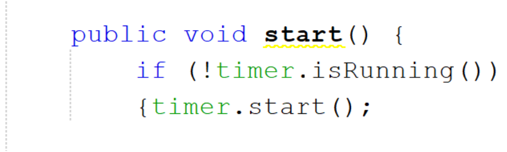
Scenario Questions

Question 2

Create a Java application for CTU Sports Day called “CTU Timer”. The application must consist of a graphic user interface similar to the one below:

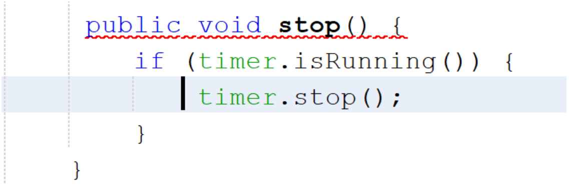
- It must consist of two (2) buttons, labels and an uneditable text field to display the counts. The application must make use of a thread to do the counting. The counter should move at intervals of one (1) second.

- When the ‘Start timer’ button is clicked, the timer should start counting.



```
public void start() {
    if (!timer.isRunning())
    {timer.start();
```

- When the ‘Stop timer’ button is clicked, the timer should stop.



```
public void stop() {
    if (timer.isRunning()) {
        timer.stop();
    }
}
```

- When the ‘Start timer’ button is clicked again, the timer must continue counting where it left off.

Completed Declaration of Authenticity

I Gabriella Rakgotsoka _ hereby
(FULL NAME)
declare that the contents of this assignment JD522_FA3 is entirely my own work except for the following documents: (List the documents and page numbers of work in this portfolio that were generated in a group)

Activity	**Date**

Signature: Date: 2024/05/16