UFAZ OOP2 MULTITHREADING MINI-PROJECT REPORT

October 8, 2018

Prepared by: Orxan Hashimli and Afrasiyab Khalili
University: UFAZ

TASK:

1. Write a **Counter** class that inherits from the **Thread** class; it has a String type attribute; its run() method counts from 1 to n by making a random pause of 0 to 5 seconds between two increments, it displays each incremented value with its name and then displays an end message. Test this class in a **TestCounter** class that starts several **Counter** objects.

SOLUTION:

We have 3 classes:

(a) **Counter** - it is a class which extends the Thread.
So, we have two methods in our class:

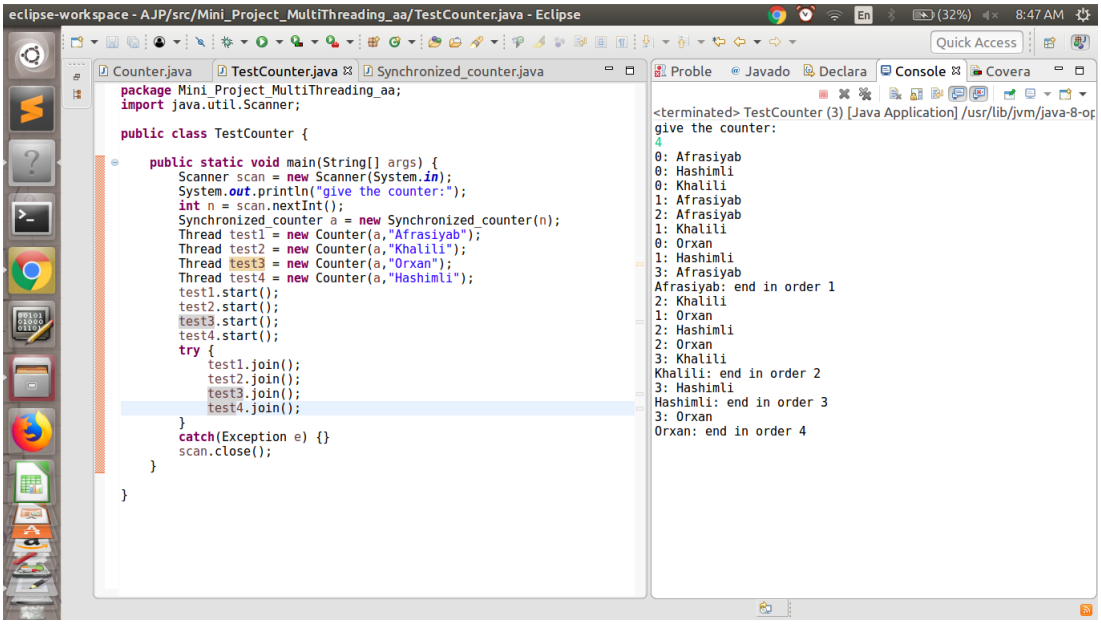
i. **Counter()** - it is a constructor that takes two parameters(name of String and counter).

ii. **run()** - it is a start point of Thread after calling **Thread.start()** method.

(b) **TestCounter** - it is a main Class. We instantiate as many instance as we can. after instantiation we can awake our Thread objects just using **Thread.start()** method. For simplicity, we use **Thread.join()** method because we want our threads to wait each other. Of course we do this in **try, catch** block to know that there is mistake or not.

(c) **Synchronized_Counter** - It is doing the main task(displaying the string and incremented value).
We use **for** loop inside the run() method then **try, catch** blocks to do that. To make a pause between increments we use **Thread.sleep()** method.
Output will be like:

18-10-08 08-47-31.png



2. Modify the **run()** method of the Counter class so that the thread displays the end message with its order of arrival.Test the change.

SOLUTION:

Generally, we have 3 classes in this exercise.

(a) **Synchronized_Counter** - it is our additional class that takes **counter** from the user.

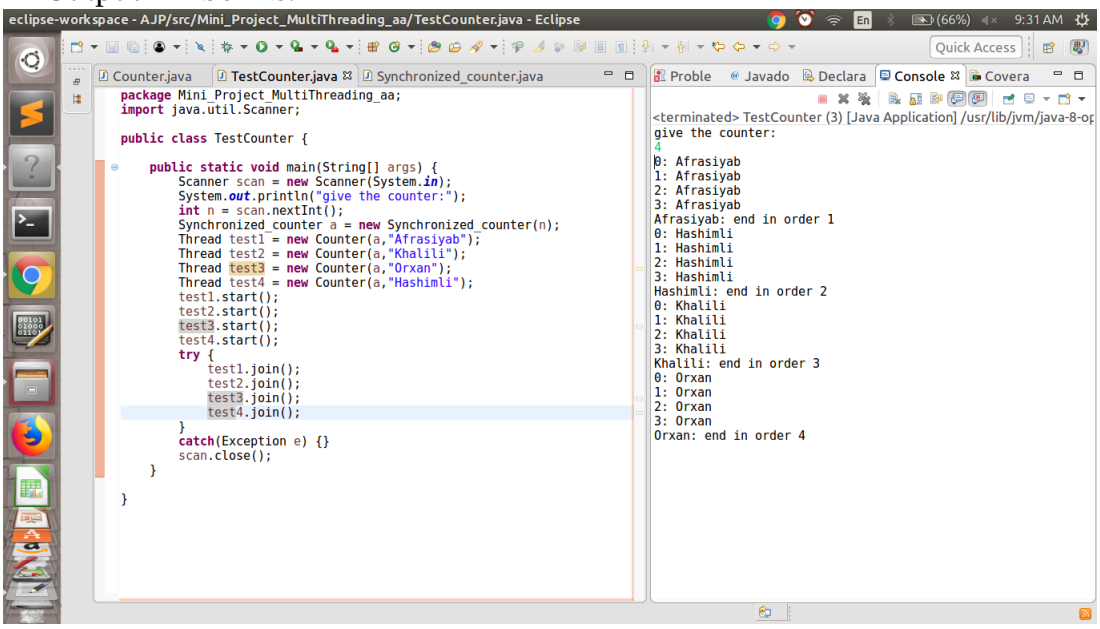
(b) **Counter** - it is a class that inherits Thread class.

(c) **TestCounter** - it is our main class.

Procedure:
First, we are passing a value to the input of **constructor of Synchronized_Counter**.Then I have another method in this class which is called **counter()** and it is **synchronized**. The main difference between (1) and (2) is that we are not using **synchronized** in our method(Picture.1), therefore we get output without order.So in the **Counter** class we create an object of Synchronized_ Counter class and we have the run() method that calls counter() method each time. Finally,In the Main, We instantiate an object which is instance of Synchronized_Counter class, then several objects which are the instances of

Counter class and we give as a input instance of Synchronized_Counter class when we instantiate Counter objects and then calling Thread.start(),Thread.join() with try,catch block.

Output will be like:



-08 09-31-17.png