Homework 7 about Generics, Multithreading, Networking (5 tasks)

Due Tuesday 04/21/2020 11:59 PM

[20 pts] Programming Task 1 – Difficulty Level: Easy

(Distinct elements in ArrayList) Write the following method that returns a new ArrayList. The new list contains the nonduplicate elements from the original list.

Public static <E> ArrayList<E> removeDuplicates(ArrayList<E> list)

[20 pts]Programming Task 2 – Difficulty Level: medium

(Parallel array initializer) Implement the following method using the **Fork/Join** Framework to assign random values to the list.

public static void parallelAssignValues(double[] list)

a. Implement a sequential algorithm to create a list with 9,000,000 elements.
b. Implement a parallel algorithm to create a list with 9,000,000 elements and invokes parallelAssignValues to assign random values to the list.

   Math.random(), your parallel code execution time will be worse than the sequential code execution time because Math.random() is synchronized and cannot be executed in parallel. To fix this problem, create a Random object for assigning random values to a small list.

c. Run ten times of a, ten times of b, and compare the result using the following table

	Run a	Run b
Time of Round 1		
Time of Round 2		
Time of Round 3		
Time of Round 4		
Time of Round 5		
Time of Round 6		
Time of Round 7		
Time of Round 8		
Time of Round 9		
Time of Round 10		

[20 pts] Programming Task 3 about multithreading – Difficulty Level: medium

(Parallel quick sort)

1.1 Implement the following method in parallel to sort a list using quick sort https://liveexample.pearsoncmg.com/html/QuickSort.html

    public static void parallelQuickSort(int[] list)

1.2 Write a test program (you can put it within the main method within class QuickSort) – Run the following experiment for ten times
   a. randomly generate a list of size 9,000,000 integers (each integer is in [0, 9,000,000); you may reuse your program in programming Task 2.
   b. record the time of sorting these integers using **QuickSort(int[] list)**;
   c. record the time of sorting the same integers using parrallelQuickSort(int[] list);

How many times, the time in c is smaller than the time in b?

[20 pts]Programming Task 4 - (Synchronize threads. Difficulty level: medium)

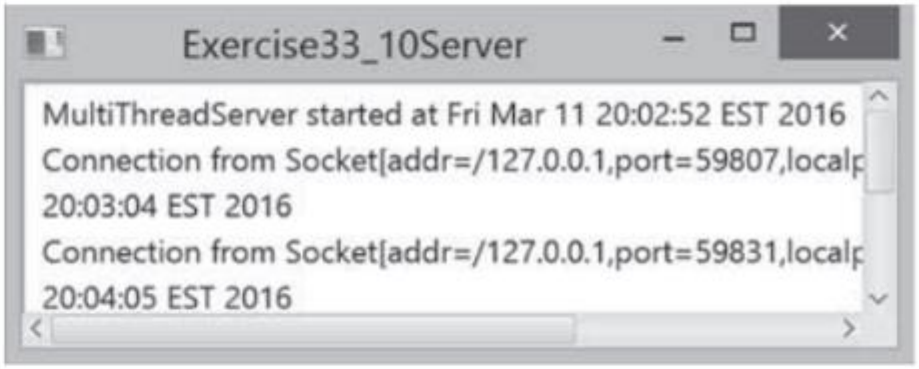
Write a program that launches 1,000 threads. Each thread subtracts 5 from a variable NumOfGold that initially is 5,000. You need to pass NumOfGold by reference to each thread. In order to pass it by reference, define an Integer wrapper object to hold NumOfGold.

Run the program **for ten times** – show the last five lines of the result of each run.

Add synchronization to the program and run the program **for ten times** – show the last five lines of the result of each run.

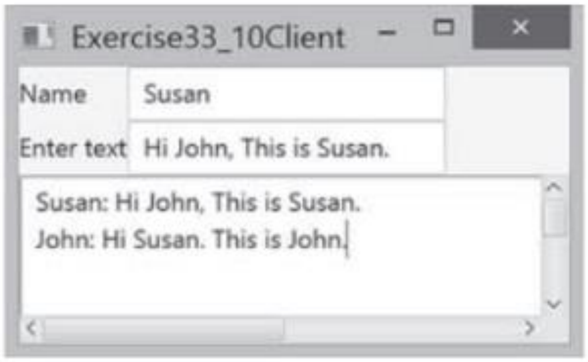
[20 pts] Programming Task 5 (Multiple Client Chat – difficulty level: hard)

(Multiple client chat) Write a program that enables any number of clients to chat. Implement one server that serves all the clients, as shown in Figure 33.22. Name the client Exercise33_10Client and the server Exercise33_10Server.



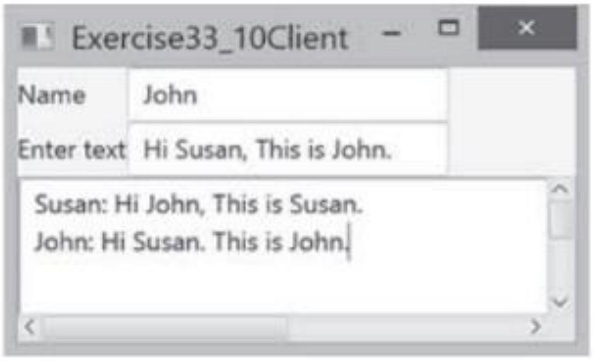
```
MultiThreadServer started at Fri Mar 11 20:02:52 EST 2016
Connection from Socket[addr=/127.0.0.1,port=59807,localp
20:03:04 EST 2016
Connection from Socket[addr=/127.0.0.1,port=59831,localp
20:04:05 EST 2016
```

(a)



```
Susan: Hi John, This is Susan.
John: Hi Susan. This is John.
```

(b)



```
Susan: Hi John, This is Susan.
John: Hi Susan. This is John.
```

(c)

Figure 33.22 The server starts in (a) with three clients in (b) and (c).