CSE 7393

Assignment 3

Due: Sep 14 (in class) / Sep 26 (distance)

Part A. Observer GaugeTracker

Load, run and study the GaugeTracker program from Chapter 6 of Knudsen. Read the javadoc for the Gauge class.

Use your com.smu.util from Assignment 2 to implement the following.

Create a class called ThreadCounter that extends Observable and implements Runnable. The run method should loop and

- sleep for 300 ms.
- wake up and increment an integer variable that is initialized to zero
- after incrementing its counter it should notify any Observers that the value has changed and report the new value using notifyObservers(…)
- terminate the loop when the value of counter exceeds 100

Write a Midlet called ThreadGauge.java that displays a gauge that tracks the value of an observable ThreadCounter instance. You should define the Midlet as an Observer so that each time it gets notified of a change via the update method, it updates the gauge.

Part B. Jargoneer Version 3 Extension

In your previous Jargoneer2 program (that caches definitions in a Hashtable) entries retrieved from the server are stored in locally so if the same request occurs again, the data can be obtained locally rather than from the network.

This works fine for definitions that are on the server, but if a definition is requested and not found on the server, the server response is: [No definition found.] which should be stored in your Hashtable to avoid network connection for a word not on the server.

However, while this works fine, it may happen that the server gets updated with the new definitions and your user will not see the new definition due to your cache. To anticipate this possibility, your task in this assignment is to determine if a requested word appears locally in the cache with the definition [No definition found.]. If so, your user should be presented with a screen that gives the option of connecting once again to the server to recheck the definition. If the user accepts the option, the program should again connect to the server and ask for the word definition. If the definition changes, the new definition should be stored in the cache. [note: this is unlikely, but possible]

Thus, your task in Jargonner3 is to determine if the requested word has already been requested with a definition of [No definition found.] – and if so, give the user the option
of reconnecting to the server. If a new definition is found, add it to the HasTable, but if [No definition found.] is returned once again, display this to the user and do not add this again to the HasTable (thereby saving power).

Submit:
- submit a zip file entitled a3<lastname>.zip to Blackboard using the Assignment 3 link
- the zip should contain:
  - observer.jar (with com.smu.util package)
  - ThreadGauge jar
  - ThreadGauge jad
  - source code
  - jargonner3.jar
  - jargoneer3.jad
  - revised source code: Jargonner3.java