March 31, 2013

Spring 2013 MP3

Develop a *Java SE NetBeans* project that implements a product distribution simulation using the *Producer-Consumer* parallel design pattern.

Create classes for a single *ProductProducer* and four *ProductConsumer.* Develop a *Product* class that encapsulates the fields from the *PRODUCT_data.txt* file. Develop a *ProductConsumer* class that encapsulates a region identifier (*N, S, E and W*) and a list to collect consumed *Product* objects.

Provide a *ProductMessage* class that encapsulates a *Product* object, the current timestamp and a random region identifier of a *ProductConsumer*.

The *ProductProducer* produces a *Product* for consumption by the *ProductConsumers*. It randomly selects a *Product* for product distribution to the *ProductConsumers* (i.e. the *ProductProducer* prepares a *ProductMessage* by populating a *ProductMessage* with a random *Product*, the current timestamp and the *ProductConsumer's* region). The *ProductProducer* then pushes the *ProductMessage* object on its internal queue.

Each *ProductConsumer* consumes only their respective *ProductMessage* objects from the *ProductProducer* and maintains an internal list of its collected products.

Provide the following functionality in the driver:
• begins the simulation by creating and starting the *ProductProducer* and *ProductConsumer* objects
• provides a capability to terminate the simulation by a single keystroke
• displays real-time queue change status in the *ProductProducer*
• displays real-time consumption data per *ProductConsumer*
• writes all *Product* objects per *ProductConsumer* to a file based on region
• displays all *Product* objects per *ProductConsumer* on completion
• displays the total elapsed time of the simulation

Provide a portable test script that executes the application and writes all display data to a file called ***mp3out.txt*** in a portable file system location (i.e. project top-level)

Comment your code and provide a detailed ***README*** (PDF only) file that includes:

- project description/abstract
- installation, compile and run-time requirements
- insights and expected results
- screen captures demonstrating all application capabilities
- generate project *Javadocs* API and move from the *dist* folder to a *docs* folder

Submit a compressed file called ***mp3.zip*** (zip only) of all project code and documentation by 04/014/13, 23:59/CDT. Late mini-projects will lose points. (**50 points**)