Tejal Gajare
A20287489

ITM 411 Mini Project 3: Read Me

1. Abstract

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

Objective: The objective of this mini project 3 was to help us understand the fundamental JAVA Concept of Multithreaded Programming using Threads.

Basic Understanding:

Creation of a thread:

An application that creates an instance of Thread must provide the code that will run in that thread. There are two ways to do this:

- *Provide a Runnable object (Implement a Runnable Interface):*

  The Runnable interface defines a single method, run, meant to contain the code executed in the thread.

- *Subclass Thread (Extend a Thread Class):*

  The Thread class itself implements Runnable, though its run method does nothing.

  We use thread. Start() in order to start the new thread.

Producer/Consumer parallel pattern:

It is one of the more common patterns in threaded programming. The idea is to process data asynchronously by partitioning requests among different groups of threads. The producer is a thread (or group of threads) that generates requests (or data) to be processed. The consumer is a thread (or group of threads) that takes those requests (or data) and acts upon them. This pattern provides a clean separation that allows for better thread design and makes development and debugging easier.

Synchronization of multi-threaded application:

When two or more threads need access to a shared resource, they need some way to ensure that the resources will be used by only one thread at a time. The process by which this is achieved is called synchronization.

Tejal Gajare
A20287489

2. System Requirement

OS: Windows 7
JDK: 1.7.0
IDE: NetBeans IDE 7.2.1

3. Architecture or System Flow

3.1 Creation of Java Application:
Launch NetBeans and click on new Project to choose a new Java Application, name it as 'MP3'and click on Finish

3.2 Creation of packages:
Along with the default package, mp3, create new packages 'domain', 'Product' , 'Producer' and 'Consumer' at the same level

3.3 Creation of classes/interfaces (Java files):

- Utilities: This Utility Class contains the methods to read the file, generate random product and create product message
- Product: This class is used to store the contents of the file in various declared member fields
- ProductMessage: This class encapsulates the Product object, timestamp and regionID
- ProducerProduct: This class in used to create the producer object
- ProductProducer: This is the Producer Base class that encapsulates the common methods. It maintains an internal queue and an array list to store the product list in order to generate random products and generate a product message to be stored in the queue.
- ProducerChild: This is the Producer class sub class. It implements the run() method.
- ProductConsumer: This class represents the main consumer base class. Each ProductConsumer Consume only their respective ProductMessage objects from the ProductProducer and maintains an internal list of its collected products.
- NorthRegionConsumer: This class represents the north region. It implements the getRegion() Method declared as abstract in its parent ProductConsumer Class.
- EastRegionConsumer: This class represents the east region. It implements the getRegion() Method declared as abstract in its parent ProductConsumer Class.

- SouthRegionConsumer: This class represents the south region. It implements the getRegion() Method declared as abstract in its parent ProductConsumer Class.

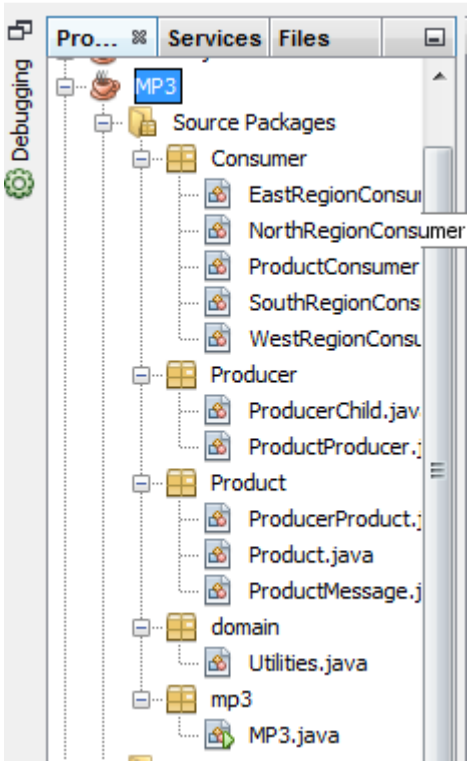
- WestRegionConsumer: This class represents the west region. It implements the getRegion() Method declared as abstract in its parent ProductConsumer Class.

- MP3: This class represents the Driver class containing the main method.

3.4 Script file:
Created a 'script.bat' file located in the MP3 folder. This file is responsible to convert the MP3.jar file (This file is created after every successful build of the java code) into mp3out.txt file (located at the root level of the main project).

3.5 Javadoc API:
This document depicts in detail in HTML format, all the comments written in the code. It can be created by right clicking on the java project and choosing the 'Generate Javadocs' option. This file folder gets automatically created inside the main project folder's 'dist' folder. It can be moved to a suitable location if needed.

4. Screen Shots



4.1 Structure of the Java Project: The above screen capture shows the basic folder structure for the MP3 JAVA application

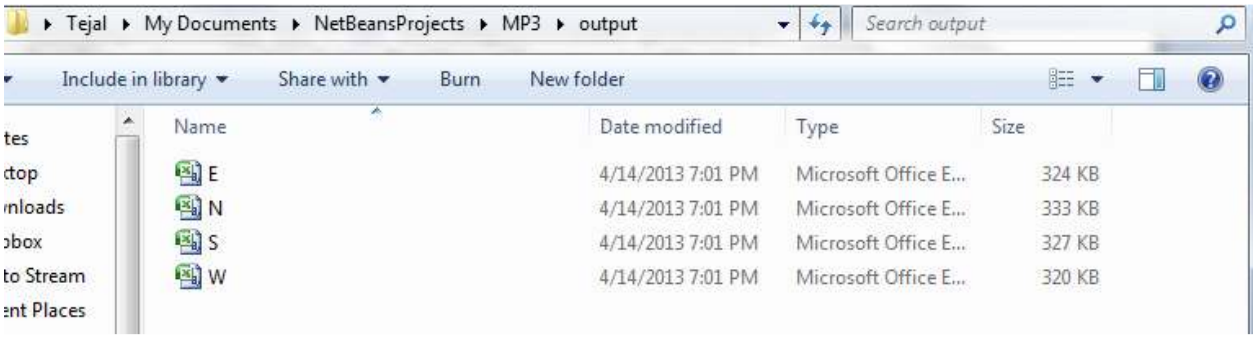
Tejal Gajare
A20287489

```
//Creation of Producer object
ProducerChild prodChild = new ProducerChild(productList);

//Creation of Consumer objects
NorthRegionConsumer nrc = new NorthRegionConsumer(prodChild);
EastRegionConsumer erc = new EastRegionConsumer(prodChild);
SouthRegionConsumer src = new SouthRegionConsumer(prodChild);
WestRegionConsumer wrc = new WestRegionConsumer(prodChild);

//To accept the keystroke to end the simulation... (PRESS ENTER)
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
System.out.println("");
System.out.println("Press 'Enter' to terminate simulation!");
```

4.2 Start the simulation by creating and starting the *ProductProducer* and *ProductConsumer* objects and provides a capability to terminate the simulation by a single keystroke – **'ENTER'**

```
Simulation begins...
-------------------------------------------------------------------------------
class Producer.ProducerChild added => Product{PRODUCT_ID=980032, MANUFACTURER_ID=19978451, PRODUCT_CODE=FW, PURCHASE_COST=39.95, QUANTITY_ON_HAND=50, MARKUP=25
class Consumer.WestRegionConsumer consumed the product message
```

4.3 Displays real-time queue change status in the *ProductProducer and* displays real-time consumption data per *ProductConsumer*



4.4 Writes all *Product* objects per *ProductConsumer* to a file based on region in the 'output' folder

Tejal Gajare
A20287489

```
class Consumer.NorthRegionConsumer Product Total:  1651
class Consumer.EastRegionConsumer Product Total:  1604
class Consumer.SouthRegionConsumer Product Total:  1618
class Consumer.WestRegionConsumer Product Total:  1589

Simulation Time in Seconds: 1
```

4.5 Displays the total number of products of each region and the total elapsed time of the simulation

5. Conclusion

Thus, the code was implemented successfully. Please go through the mp3out.txt in order to view the output of the code

6. Submission Package

It includes following folders:

1. MP3:
This includes the main JAVA application
- It has a script.bat file which is responsible to create the mp3out.txt file at the project top level.
- It has a data folder which contains the input file 'PRODUCT_data.csv'
- It has a output folder that contains the individual region files N.csv, E.csv, S.csv and W.csv

2. MiniProject3 Docs:
This includes the JavaAPI docs and the README file