## Sales Processing : Table Of Contents:

[1.1. Sales Processing : Table Of Contents: 1](#_Toc478038539)

[2. Project: Building And Execution 2](#_Toc478038540)

[2.1. From Command Prompt: Running the program 2](#_Toc478038541)

[2.2. From Command Prompt: Executing the Junit 2](#_Toc478038542)

[2.3. From Eclipse: Running the program 2](#_Toc478038543)

[2.4. From Eclipse: Running the JUnits 2](#_Toc478038544)

[3. Configuring the Project 2](#_Toc478038545)

[3.1. Configuring the NumberOfMessages, Sales log count, Sale Adjustment Log count 2](#_Toc478038546)

[4. Project: Sample run with configuration: 3](#_Toc478038547)

[4.1. Confiruation :{totalMessagesToBeSent:16,logProdAdjstOnMsgsReceived:4,logProdOnMsgsReceived\_count:2 } 3](#_Toc478038548)

[4.2. Confiruation :{totalMessagesToBeSent:100,logProdAdjstOnMsgsReceived:50,logProdOnMsgsReceived\_count:10} 3](#_Toc478038549)

[5. Main Files 3](#_Toc478038550)

[5.1. Main Class: 3](#_Toc478038551)

[5.2. MessageGenerator: 3](#_Toc478038552)

[5.3. MessageQueueProvider: 3](#_Toc478038553)

[5.4. MessageSender: 3](#_Toc478038554)

[5.5. MessageReceiver: 3](#_Toc478038555)

[5.6. MessageSenderReceiver: 3](#_Toc478038556)

[5.7. ModelClasses: MProducts: 3](#_Toc478038557)

[5.8. ModelClasses: MProduct: 3](#_Toc478038558)

[5.9. ModelClasses: MProduct: 3](#_Toc478038559)

[5.10. ModelClasses: MSaleHist: 4](#_Toc478038560)

[6. Help to interpret User Prompts 4](#_Toc478038561)

[6.1. Logging the Configuration when the program starts 4](#_Toc478038562)

[When the program is started Configuration is displayed: 4](#_Toc478038563)

[6.2. Understanding Product Sale Log 4](#_Toc478038564)

[Sale recorded will show all the sales for all the products. The will be shown as per the configuration of 4](#_Toc478038565)

[6.3. Validating the Product Sale Log 4](#_Toc478038566)

[6.4. Understanding the Adjustment Log: (pls look from bottom to up for each adjustment) 5](#_Toc478038567)

[1) It shows the totalAdjustments in Orange 5](#_Toc478038568)

[2) The bottom text shows the first message that was received as part of sale. Shown in RED 5](#_Toc478038569)

[7. Verification of Adjustments 5](#_Toc478038570)

[8. Log showing that after logProdAdjstOnMsgsReceived, messeages would not be received and adjustments are logged. 5](#_Toc478038571)

# 

# Project: Building And Execution

## From Command Prompt: Running the program

1. Download the project from git hub (and unzip)
2. Install Java (tested and executed using java version "1.8.0\_121")
3. Install maven (tested on with mvn:version:3.3.9)
4. Open the command prompt (cmd.exe) and change directory to the downloaded project directory [ from step1].
5. $ mvn clean compile exec:java

## From Command Prompt: Executing the Junit

1. Open the command prompt (cmd.exe) and change directory to the downloaded project directory [ from step1].
2. $ mvn clean test

## From Eclipse: Running the program

1. Import the project into eclipse
2. Install Java (tested and executed using java version "1.8.0\_121")
3. Install maven (tested on with mvn:version:3.3.9)
4. Open the file: **com.mprocessor.sales.exc.SalesProcessingExec.java** and rt click and RunAs -> Java Application

## From Eclipse: Running the JUnits

1. All the Junits are present in the folder src/test/java . Rt click and RunAs -> Junit TestCases

# Configuring the Project

## Configuring the NumberOfMessages, Sales log count, Sale Adjustment Log count

|  |
| --- |
| In File : -> package com.mprocessor.sales.cfgSalesProcessingConfig  *// changes to change the count for logging the Products Adjustment* **public static int** *logProdAdjstOnMsgsReceived*= 4; *// changes to change the count for logging the Sales Recorded* **public static int** *logProdOnMsgsReceived\_count*= 2; *// change the number of messages that has to be sent for RandomMessageGenerator* **public static int** *totalMessagesToBeSent* = 16; |

# Project: Sample run with configuration:

## Confiruation :{totalMessagesToBeSent:16,logProdAdjstOnMsgsReceived:4,logProdOnMsgsReceived\_count:2 }

Please check the attached file: *sampleRun\_output\_msgs16\_log2\_adj4.txt*

## Confiruation :{totalMessagesToBeSent:100,logProdAdjstOnMsgsReceived:50,logProdOnMsgsReceived\_count:10}

Please check the attached file: *sampleRun\_output\_msgs100\_log10\_adj50.txt*

# Main Files

## Main Class:

|  |  |
| --- | --- |
| ClassName: | com.mprocessor.sales.excSalesProcessingExec |
| Description | This is the main class that starts the processor: |

## MessageGenerator:

|  |  |
| --- | --- |
| ClassName: | RandomMessageGenerator |
| Description | This class generates the Random messages. Please not there are 3 products that are configured to send the message.  Messagetypes- This class generates all the 3 kinds of messages randomly. |

## MessageQueueProvider:

|  |  |
| --- | --- |
| ClassName: | com.mprocessor.sales.cfg.SalesProcessingConfig.MessageQueueProvider |
| Description | Class that hass a Queue object to hold the messages |

## MessageSender:

|  |  |
| --- | --- |
| ClassName: | com.mprocessor.sales.cfg.SalesProcessingConfig.MessageSender implments Runnable. |
| Description | This class sends the messages into the queue |

## MessageReceiver:

|  |  |
| --- | --- |
| ClassName: | com.mprocessor.sales.cfg.SalesProcessingConfig.MessageReceiver implments Callable |
| Description | This class receives the messages from the queue. Once the message is recived A Sale is recorded. |

## MessageSenderReceiver:

|  |  |
| --- | --- |
| ClassName: | com.mprocessor.sales.cfg.SalesProcessingConfig.MessageSenderReceiver |
| Description | Static methods that facilitates the message Sending and Receiver. |

## ModelClasses: MProducts:

|  |  |
| --- | --- |
| ClassName: | com.mprocessor.sales.model.MProducts: |
| Description | Contains list of all the products processed. |

## ModelClasses: MProduct:

|  |  |
| --- | --- |
| ClassName: | com.mprocessor.sales.model.MProduct |
| Description | Contains a descripiton of single product.It contains the list of all MSales belonging to the product. |

## ModelClasses: MProduct:

|  |  |
| --- | --- |
| ClassName: | com.mprocessor.sales.model.MSale |
| Description | Contains details about the sale. totalVal, indivVal,ocdurence, History of Sale ie after adjustments |

## ModelClasses: MSaleHist:

|  |  |
| --- | --- |
| ClassName: | com.mprocessor.sales.model. MSaleHist |
| Description | This contains the history of sale after the sale is adjusted. |

# Help to interpret User Prompts

## Logging the Configuration when the program starts

## When the program is started Configuration is displayed:

|  |
| --- |
| --------------------------  <<START>> SALES PROCESSING  Configuration{totalMessagesToBeSent:16,logProdAdjstOnMsgsReceived:4,logProdOnMsgsReceived\_count:2}  -------------------------- |

## Understanding Product Sale Log

## Sale recorded will show all the sales for all the products. The will be shown as per the configuration of

SalesProcessingConfig.logProdOnMsgsReceived\_count setting variable.

|  |
| --- |
| ---------------------------------------  LOG Product Sales: ,MsgsProcessed:4  ---------------------------------------  MProducts {  [Product{name:prod1,totalOfAllSales:23, sales:{  [MSale{totalSaleValue:6,indivSalValue:3, occurrence:2,totalAdjsts:0  ,FirstMsg:Message{msgType:'MSG\_TYPE\_2',productType:'prod1', value:3,occurrence:2}}],  [MSale{totalSaleValue:9,indivSalValue:3, occurrence:3,totalAdjsts:0  ,FirstMsg:Message{msgType:'MSG\_TYPE\_2',productType:'prod1', value:3,occurrence:3}}],  [MSale{totalSaleValue:5,indivSalValue:5, occurrence:1,totalAdjsts:0  ,FirstMsg:Message{msgType:'MSG\_TYPE\_1',productType:'prod1', value:5}}],  [MSale{totalSaleValue:3,indivSalValue:1, occurrence:3,totalAdjsts:0  ,FirstMsg:Message{msgType:'MSG\_TYPE\_2',productType:'prod1', value:1,occurrence:3}}]}}]}  -------------------------  END : LOG Product Sales:  --------------------------------------- |

## Validating the Product Sale Log

1)Always the totalSaleValue = individualSaleValue \* occurrence. Ie in a row yellow value is equal to multiplication of Blue values.

MSale{totalSaleValue:5,indivSalValue:5, occurrence:1,totalAdjsts:0

1. Sum of totalSaleValue is equal to the product : totallAllSales.

23=(6+9+5+3);

## Understanding the Adjustment Log: (pls look from bottom to up for each adjustment)

Following shows a sample adjustment ...

## It shows the totalAdjustments in Orange

## The bottom text shows the first message that was received as part of sale. Shown in RED

1. Yellow shows the ***totalSaleValue*** and ***individualSaleValue*** after adjustment is made
2. Blue: Shows the ***operation*** {like subtraction, add, multiplication} and the ***adjustedVal***.

|  |
| --- |
| [MSale{totalSaleValue:9,indivSalValue:3, occurrence:3,**totalAdjsts:5**  **<<ADJUSTMENT : 5>>:**  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'add', adjVal:1, value:5,occurrence:1  <PREV-MSale{totalSaleValue:6,indivSalValue:2, occurrence:3}>  **<<ADJUSTMENT : 4>>:**  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'subtract', adjVal:1, value:4,occurrence:1  <PREV-MSale{totalSaleValue:9,indivSalValue:3, occurrence:3}>  **<<ADJUSTMENT : 3>>:**  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'subtract', adjVal:1, value:4,occurrence:1  <PREV-MSale{totalSaleValue:12,indivSalValue:4, occurrence:3}>  **<<ADJUSTMENT : 2>>:**  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'multiply', adjVal:2, value:4,occurrence:1  <PREV-MSale{totalSaleValue:6,indivSalValue:2, occurrence:3}>  <<ADJUSTMENT : 1>>:  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'add', adjVal:1, value:4,occurrence:1  <PREV-MSale{totalSaleValue:3,indivSalValue:1, occurrence:3}>  -FirstMsg-:Message{msgType:'MSG\_TYPE\_2',productType:'prod1', value:1,occurrence:3}}], |
| $YOUR INPUT:---------> |

# Verification of Adjustments

1)Always the totalSaleValue = individualSaleValue \* occurrence

2) Two consequitive PREV-MSale value should be derived from the adjustment made from the \_DUE2\_MSG.

|  |
| --- |
| <PREV-MSale{totalSaleValue:12,indivSalValue:4, occurrence:3}>  **<<ADJUSTMENT : 2>>:**  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'multiply', adjVal:2, value:4,occurrence:1  <PREV-MSale{totalSaleValue:6,indivSalValue:2, occurrence:3}> |

# Log showing that after logProdAdjstOnMsgsReceived, messeages would not be received and adjustments are logged.

|  |
| --- |
| -------------------------------------------  LOG Product Adjustments: ,MsgsProcessed:8  -------------------------------------------  MProducts {  [Product{name:prod1,totalOfAllSales:78, sales:{  [MSale{totalSaleValue:14,indivSalValue:7, occurrence:2,totalAdjsts:3  <<ADJUSTMENT : 3>>:  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'subtract', adjVal:1, value:4,occurrence:1  <PREV-MSale{totalSaleValue:16,indivSalValue:8, occurrence:2}>  <<ADJUSTMENT : 2>>:  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'multiply', adjVal:2, value:4,occurrence:1  <PREV-MSale{totalSaleValue:8,indivSalValue:4, occurrence:2}>  <<ADJUSTMENT : 1>>:  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'add', adjVal:1, value:4,occurrence:1  <PREV-MSale{totalSaleValue:6,indivSalValue:3, occurrence:2}>  -FirstMsg-:Message{msgType:'MSG\_TYPE\_2',productType:'prod1', value:3,occurrence:2}}],  [MSale{totalSaleValue:21,indivSalValue:7, occurrence:3,totalAdjsts:3  <<ADJUSTMENT : 3>>:  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'subtract', adjVal:1, value:4,occurrence:1  <PREV-MSale{totalSaleValue:24,indivSalValue:8, occurrence:3}>  <<ADJUSTMENT : 2>>:  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'multiply', adjVal:2, value:4,occurrence:1  <PREV-MSale{totalSaleValue:12,indivSalValue:4, occurrence:3}>  <<ADJUSTMENT : 1>>:  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'add', adjVal:1, value:4,occurrence:1  <PREV-MSale{totalSaleValue:9,indivSalValue:3, occurrence:3}>  -FirstMsg-:Message{msgType:'MSG\_TYPE\_2',productType:'prod1', value:3,occurrence:3}}],  [MSale{totalSaleValue:11,indivSalValue:11, occurrence:1,totalAdjsts:3  <<ADJUSTMENT : 3>>:  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'subtract', adjVal:1, value:4,occurrence:1  <PREV-MSale{totalSaleValue:12,indivSalValue:12, occurrence:1}>  <<ADJUSTMENT : 2>>:  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'multiply', adjVal:2, value:4,occurrence:1  <PREV-MSale{totalSaleValue:6,indivSalValue:6, occurrence:1}>  <<ADJUSTMENT : 1>>:  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'add', adjVal:1, value:4,occurrence:1  <PREV-MSale{totalSaleValue:5,indivSalValue:5, occurrence:1}>  -FirstMsg-:Message{msgType:'MSG\_TYPE\_1',productType:'prod1', value:5}}],  [MSale{totalSaleValue:9,indivSalValue:3, occurrence:3,totalAdjsts:3  <<ADJUSTMENT : 3>>:  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'subtract', adjVal:1, value:4,occurrence:1  <PREV-MSale{totalSaleValue:12,indivSalValue:4, occurrence:3}>  <<ADJUSTMENT : 2>>:  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'multiply', adjVal:2, value:4,occurrence:1  <PREV-MSale{totalSaleValue:6,indivSalValue:2, occurrence:3}>  <<ADJUSTMENT : 1>>:  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'add', adjVal:1, value:4,occurrence:1  <PREV-MSale{totalSaleValue:3,indivSalValue:1, occurrence:3}>  -FirstMsg-:Message{msgType:'MSG\_TYPE\_2',productType:'prod1', value:1,occurrence:3}}],  [MSale{totalSaleValue:7,indivSalValue:7, occurrence:1,totalAdjsts:2  <<ADJUSTMENT : 2>>:  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'subtract', adjVal:1, value:4,occurrence:1  <PREV-MSale{totalSaleValue:8,indivSalValue:8, occurrence:1}>  <<ADJUSTMENT : 1>>:  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'multiply', adjVal:2, value:4,occurrence:1  <PREV-MSale{totalSaleValue:4,indivSalValue:4, occurrence:1}>  -FirstMsg-:Message{msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'add', adjVal:1, value:4,occurrence:1}}],  [MSale{totalSaleValue:3,indivSalValue:3, occurrence:1,totalAdjsts:1  <<ADJUSTMENT : 1>>:  <ADJ\_DUE2\_MSG>:msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'subtract', adjVal:1, value:4,occurrence:1  <PREV-MSale{totalSaleValue:4,indivSalValue:4, occurrence:1}>  -FirstMsg-:Message{msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'multiply', adjVal:2, value:4,occurrence:1}}],  [MSale{totalSaleValue:4,indivSalValue:4, occurrence:1,totalAdjsts:0  -FirstMsg-:Message{msgType:'MSG\_TYPE\_3',productType:'prod1',operation:'subtract', adjVal:1, value:4,occurrence:1}}],  [MSale{totalSaleValue:9,indivSalValue:3, occurrence:3,totalAdjsts:0  -FirstMsg-:Message{msgType:'MSG\_TYPE\_2',productType:'prod1', value:3,occurrence:3}}]}}]}  -------------------------------------------  END :LOG Product Adjustments:  --------------------------- |