Özlem Ataş

08.02.2017

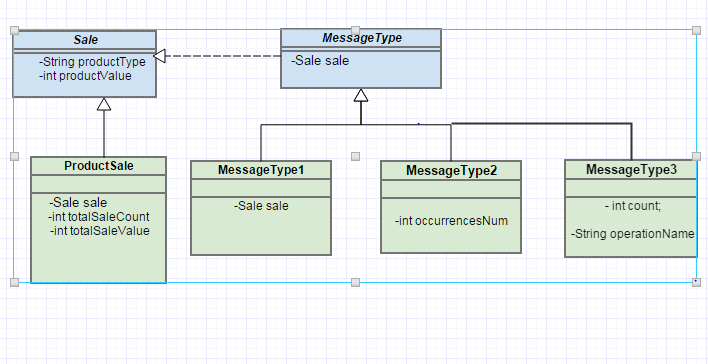
*Message Processing Problem Solution*

In this submission, the Message Processing Application problem was solved in Java language.

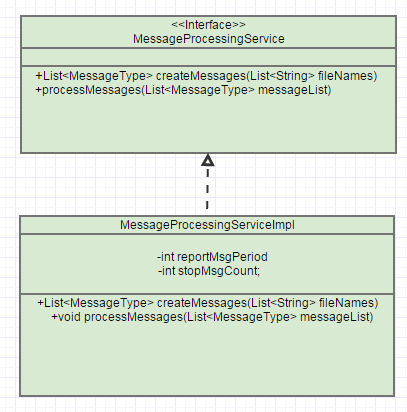
This document describes the details of design, project configuration, implementation and guides for

how to run Java implementations.

*Design*

MessageType is an abstract class composed of Sale object. Sale holds the product type and product value belongs to that sale. MessageType1, MessageType2 and MessageType3 are subclasses extended from MessageType abstract class. ProductSale holds the totalproduct count and total product value about whole sale. 

MessageProcessingService is an interface that responsible for reading messages from files and process the messages for sale. All business logic is implemented in MessageProcessingService Impl class.



Default file path, message period count (for reporting), message stop count are kept as parametric in application.properties file.

report.msg.period=2

stop.msg.count=6

file.path=D:\\messages.txt

Messages must be kept as below format at messages.txt file

**MessageType1:ProductName:ProductValue**

**MessageType2: ProductName: ProductValue: ProductCount**

**MessageType3: ProductName: ProductValue:ADJUSTMENT:Count**

Example message.txt file is attached here.



*Version Controlling with Git*

In this project git is selected as the version control system and Github cloud service is selected as

the git repository.

Source codes are committed to following links:

*Build Management with Maven*

Maven is used as the build and artifact dependency management tool for Java implementations.

In order to compile source code and build the implementation, following command should be

executed from command line in the project directories:

mvn clean package

This command will clean previous build directory, re-compile the source code and

package compiled classes.

*Running Projects*

Java project is implemented using Spring Boot for application start-up.

In order to run the project, just type the following command and provide message input file as specified in the problem definition.

$ mvn spring-boot:run -Drun.arguments="inputFileName"

The application will print the results on conlose as below:

Use default input file-> D:\messages.txt

+-----------------+--------------------------------------+

| Sale Product | Number of Sales | Total Value |

+-----------------+--------------------------------------+

| Apple | 1 | 10 |

| Orange | 10 | 20 |

+-----------------+--------------------------------------+

+-----------------+--------------------------------------+

| Sale Product | Number of Sales | Total Value |

+-----------------+--------------------------------------+

| Apple | 2 | 20 |

| Orange | 20 | 40 |

+-----------------+--------------------------------------+

+-----------------+--------------------------------------+

| Sale Product | Number of Sales | Total Value |

+-----------------+--------------------------------------+

| Apple | 2 | 60 |

| Orange | 20 | 40 |

+-----------------+--------------------------------------+

Application is pausing and will not accept new message...

Adjustment resports are below:

ADD 5 Apple

MULTIPLY 2 Apple