

CSci 364
Spring, 2023
Programming Assignment #3
Due: Wednesday, March 1, 2023

Design and implement a distributed computing application using Java Message Service (JMS) (aka Jakarta Messaging) and serialized objects. This will require implementing a two types of JMS clients—one that reads work data (the manager) and another JMS client (the employee) that performs the work tasks.

Apache ActiveMQ v5 will be the JMS provider. (See below for a download link and instructions for starting ActiveMQ.) The provider will host two message queues: a data queue for sending from the manager to the worker and a work queue for sending from the employee to the manager.

The manager

The manager JMS client reads a text input file specified on the command line. For each row of data, the manager sends the row to the data queue as a `javax.jms.TextMessage` object. After the employee processes the data, the manager receives the work result from the work queue as a `javax.jms.ObjectMessage` object. The manager then prints the original input data (the row) and the results to the console.

The employee

The employee JMS client will receive JMS `TextMessage` objects containing a `String` of data (a row) from the data queue in ActiveMQ. Your employee client shall process the data using one of three `api.Worker` (see the `api` package) subclass objects described below. Specifically, the client shall run the `Worker.doWork()` method. After the `doWork` method has completed, the client shall send a JMS `ObjectMessage` to the work queue in ActiveMQ. The `ObjectMessage` object will contain the instance of the `Worker` subclass with the computed result.

The input file

Items on each line in the input file are separated by spaces. The format of each line is as follows. The ids in the input file are unique.

```
id      value1 [value2 [value3 [...]]]
```

The line starts with an `id` field, an integer that represents the work task id. The line has one or more additional values. If the line has `value1`, the work task is to check if the value is prime. If the line has two values, `value1` and `value2` are numerator and denominator of a fraction, respectively. The task is to determine if the fraction can be reduced. Finally, if the line has three or more values, the client must sort the `String` values.

The work tasks

Each work task object extends `api.Worker` abstract class and must implement the `doWork()` and `getWorkResults()` methods.

PrimeChecker

See the provided code for an example.

FractionReducer

Given a numerator and denominator, calculate the greatest common factor. Divide the input fields by the greatest common factor. Your class should have two accessor methods to retrieve the reduced numerator and reduced denominator.

Given input numerator of 12 and input denominator of 9, the reduced numerator should be 4 and the reduced denominator should be 3.

<https://www.mathsisfun.com/greatest-common-factor.html>

ValueSorter

Sort the values into ascending order.

A partial working example of the manager and worker is available on Blackboard (hw3.zip). DO NOT CHANGE the provided code.

Given following example input:

```
1 27
2 13
3 7 3
4 cherry apple banana
5 4 6
6 2 3 4 1
```

The manager output should be similar to the following.

Task: 1, Input: 27. Prime false

Task: 2, Input: 13. Prime true

Task: 3, Input: 7 3. Reduced fraction: 7/3

Task: 4, Input: cherry apple banana. Sorted: apple banana cherry

Task: 5, Input: 4 6. Reduced fraction: 2/3

Task: 6, Input: 2 3 4. Sorted: 1 2 3 4

Grading Rubric

In addition to the grading rubrics from HW1 and HW2, you must not utilize anything defined in the manager package from the employee package and vice versa.

Submit to Blackboard

Submit your cleaned project directory (see below) to Blackboard as a tarred or zipped archive (hw3-lastname-firstname.tar | hw3-lastname-firstname.zip). Your Ant script should have targets to clean and compile your source code and generate Javadoc comments.

```
hw3-yourname/
  build.xml
  lib/
    activemq-all-5.15.8.jar
    log4j-api-2.20.0.jar
    log4j-core-2.20.0.jar
  src/
    api/
      Worker.java
      PrimeChecker.java
      FractionReducer.java
      ValueSorter.java
  manager/
```

[Java source files related to the manager JMS client]

employee/

[Java source files related to the employee JMS client]

Obtaining and running Apache ActiveMQ

Download the latest version of Apache ActiveMQ (currently version 5.17.3).

<https://activemq.apache.org/components/classic/download/>

Untar or unzip to a location of your choice in your file system. Open a terminal window and change to the apache-activemq-5.17.3/ directory.

Start the ActiveMQ JMS provider

bin/activemq console

For more help...

<https://activemq.apache.org/version-5-getting-started>