CSE 488 (Section 1)
[Summer 2022]

Lab 02 Assignment Submission Report

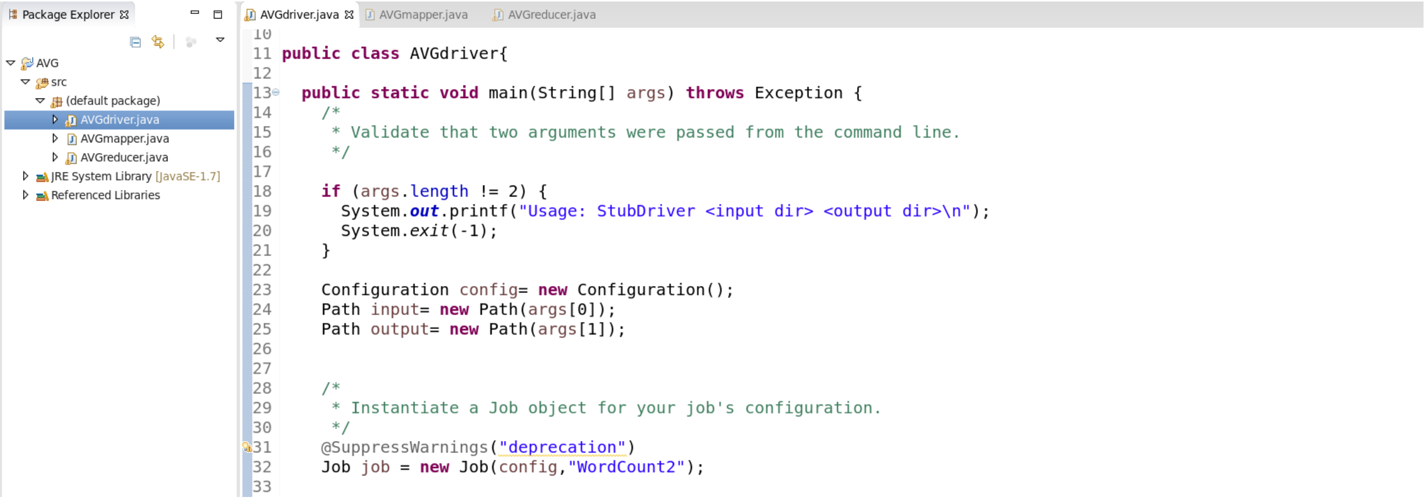
Assignment Title: MapReduce Programming

Submitted by:
Rizvee Hassan Prito
2019-3-60-041

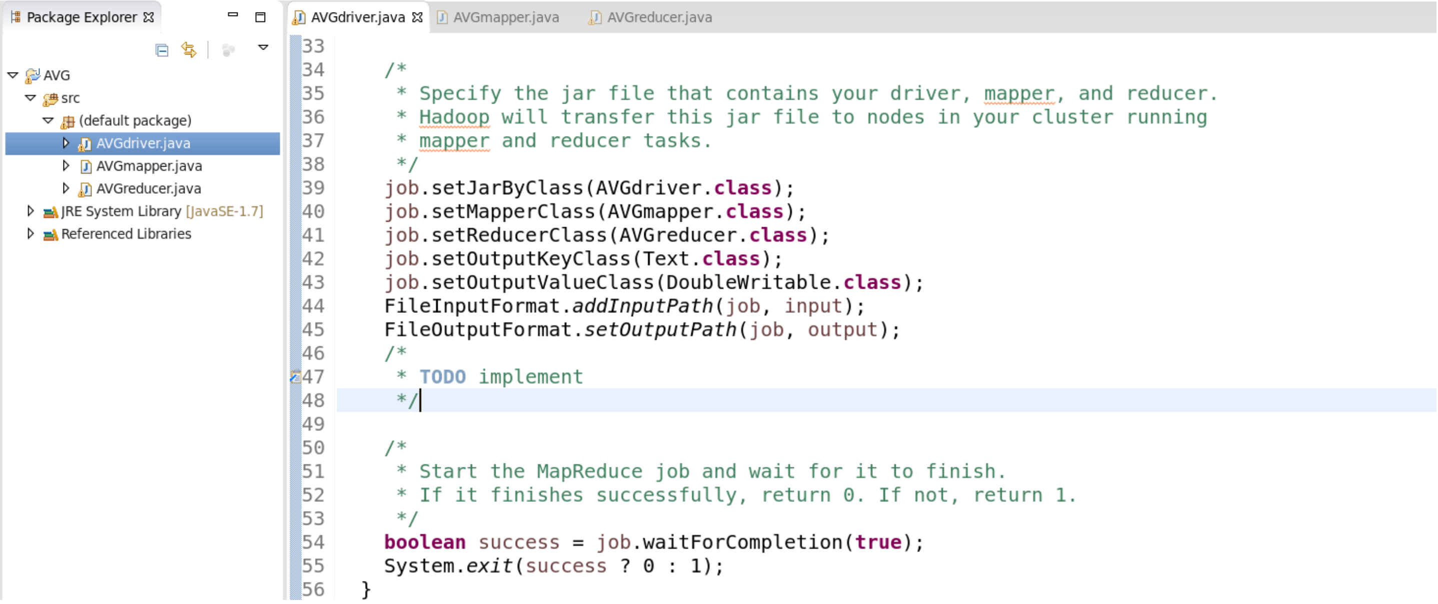
1. Screenshots

Problem 1:

Driver class:

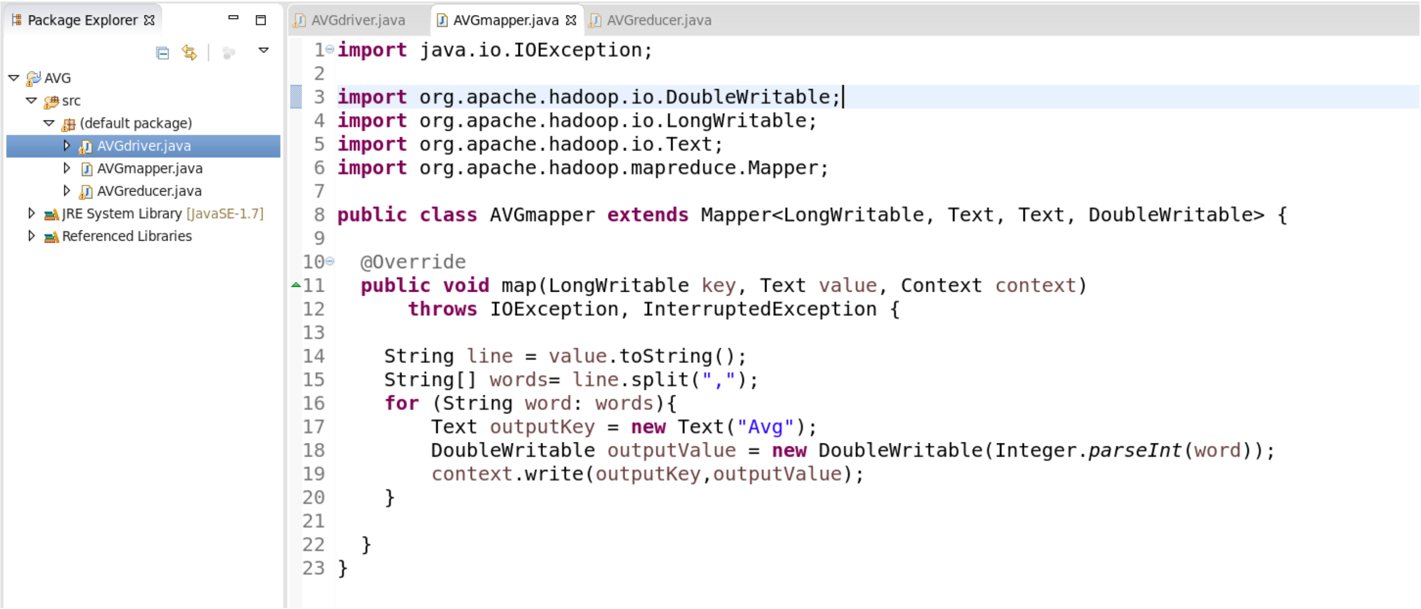


```
public class AVGdriver{

  public static void main(String[] args) throws Exception {
    /*
     * Validate that two arguments were passed from the command line.
     */

    if (args.length != 2) {
      System.out.printf("Usage: StubDriver <input dir> <output dir>\n");
      System.exit(-1);
    }

    Configuration config= new Configuration();
    Path input= new Path(args[0]);
    Path output= new Path(args[1]);


    /*
     * Instantiate a Job object for your job's configuration.
     */
    @SuppressWarnings("deprecation")
    Job job = new Job(config,"WordCount2");

```



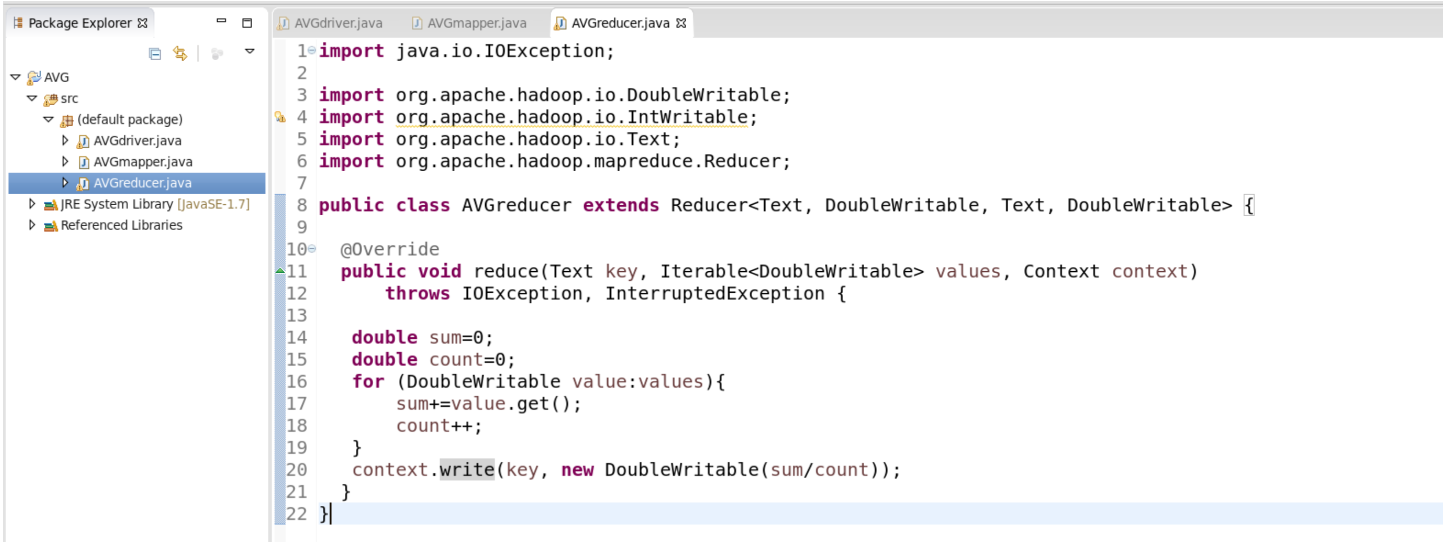
```

    /*
     * Specify the jar file that contains your driver, mapper, and reducer.
     * Hadoop will transfer this jar file to nodes in your cluster running
     * mapper and reducer tasks.
     */
    job.setJarByClass(AVGdriver.class);
    job.setMapperClass(AVGmapper.class);
    job.setReducerClass(AVGreducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(DoubleWritable.class);
    FileInputFormat.addInputPath(job, input);
    FileOutputFormat.setOutputPath(job, output);
    /*
     * TODO implement
     */

    /*
     * Start the MapReduce job and wait for it to finish.
     * If it finishes successfully, return 0. If not, return 1.
     */
    boolean success = job.waitForCompletion(true);
    System.exit(success ? 0 : 1);
  }
```

Mapper class:



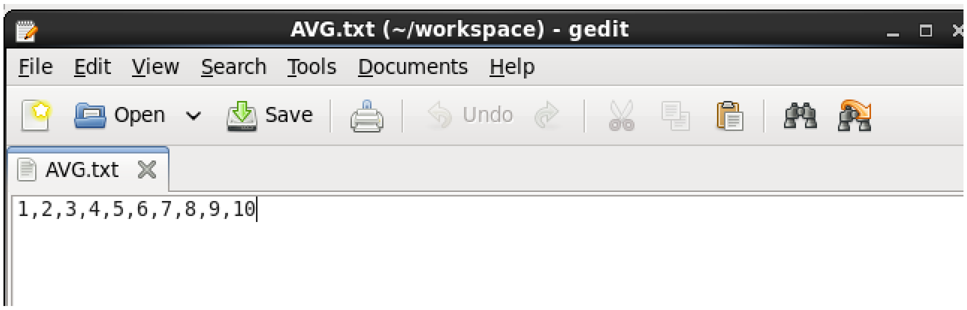
```java
import java.io.IOException;

import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class AVGmapper extends Mapper<LongWritable, Text, Text, DoubleWritable> {

  @Override
  public void map(LongWritable key, Text value, Context context)
      throws IOException, InterruptedException {

    String line = value.toString();
    String[] words= line.split(",");
    for (String word: words){
        Text outputKey = new Text("Avg");
        DoubleWritable outputValue = new DoubleWritable(Integer.parseInt(word));
        context.write(outputKey,outputValue);
    }

  }
}
```

Reducer class:



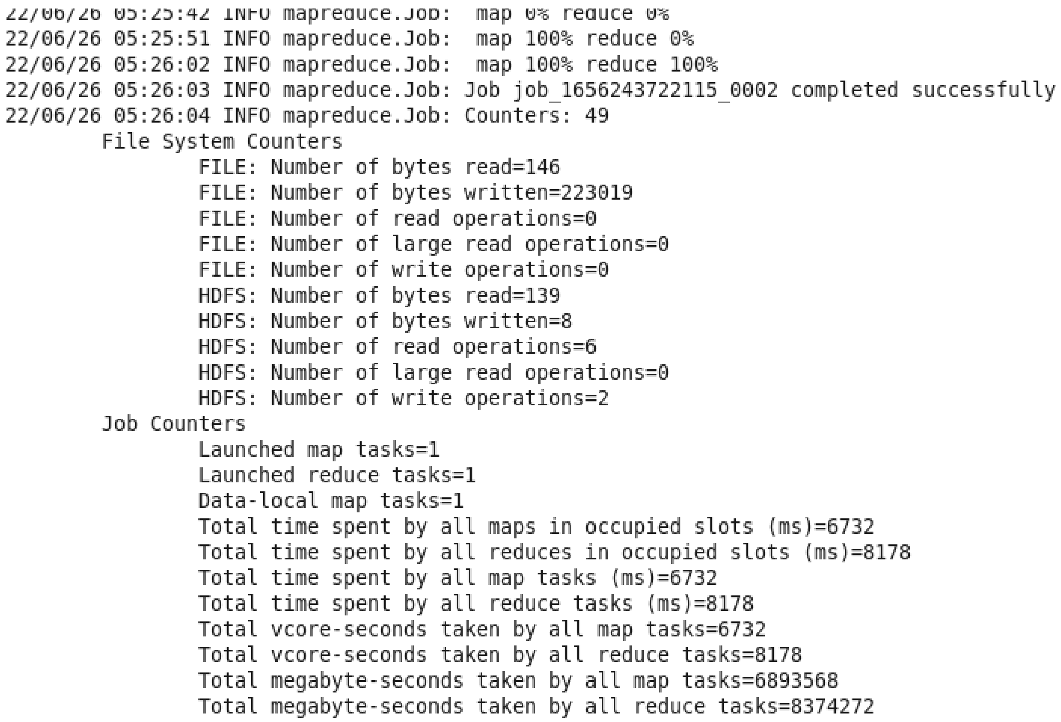
```java
import java.io.IOException;

import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class AVGreducer extends Reducer<Text, DoubleWritable, Text, DoubleWritable> {

  @Override
  public void reduce(Text key, Iterable<DoubleWritable> values, Context context)
      throws IOException, InterruptedException {

   double sum=0;
   double count=0;
   for (DoubleWritable value:values){
       sum+=value.get();
       count++;
   }
   context.write(key, new DoubleWritable(sum/count));
  }
}
```

Input File:

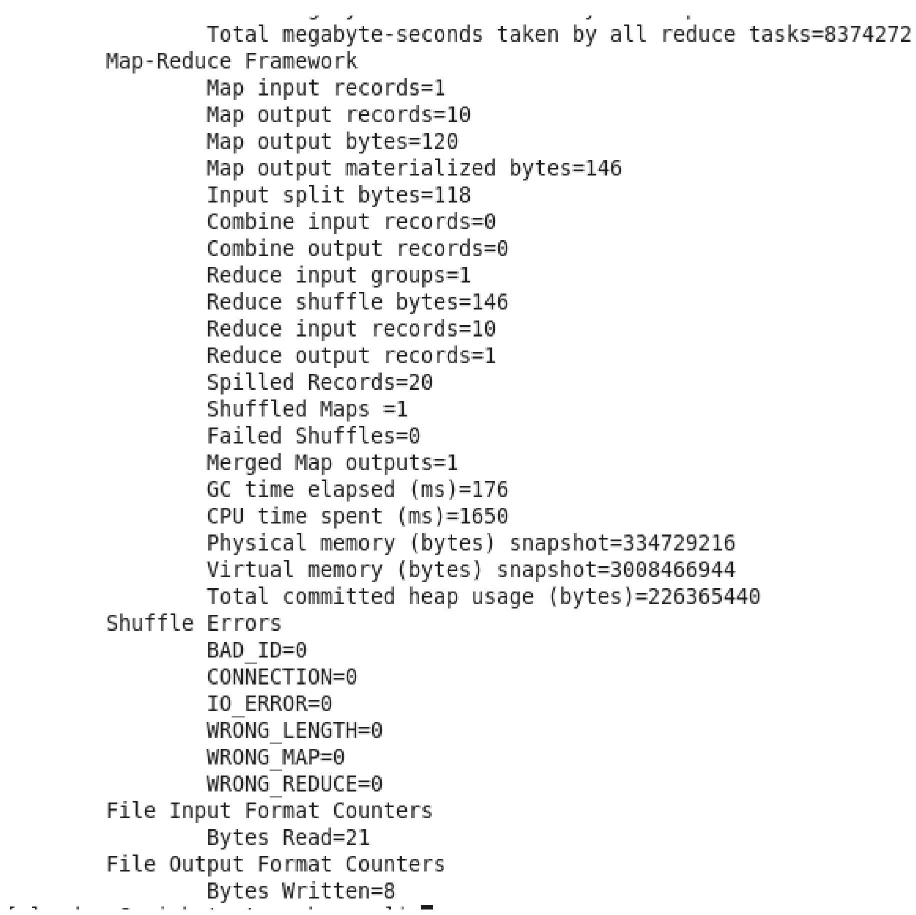

AVG.txt (~/workspace) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
AVG.txt

```
1,2,3,4,5,6,7,8,9,10
```

Terminal:

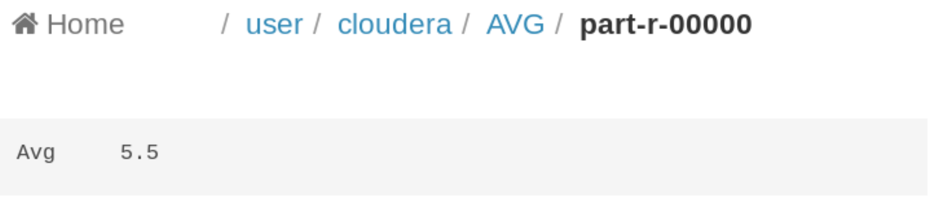


```
22/06/26 05:25:42 INFO mapreduce.Job:  map 0% reduce 0%
22/06/26 05:25:51 INFO mapreduce.Job:  map 100% reduce 0%
22/06/26 05:26:02 INFO mapreduce.Job:  map 100% reduce 100%
22/06/26 05:26:03 INFO mapreduce.Job: Job job_1656243722115_0002 completed successfully
22/06/26 05:26:04 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=146
                FILE: Number of bytes written=223019
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=139
                HDFS: Number of bytes written=8
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=6732
                Total time spent by all reduces in occupied slots (ms)=8178
                Total time spent by all map tasks (ms)=6732
                Total time spent by all reduce tasks (ms)=8178
                Total vcore-seconds taken by all map tasks=6732
                Total vcore-seconds taken by all reduce tasks=8178
                Total megabyte-seconds taken by all map tasks=6893568
                Total megabyte-seconds taken by all reduce tasks=8374272
```



```
                Total megabyte-seconds taken by all reduce tasks=8374272
        Map-Reduce Framework
                Map input records=1
                Map output records=10
                Map output bytes=120
                Map output materialized bytes=146
                Input split bytes=118
                Combine input records=0
                Combine output records=0
                Reduce input groups=1
                Reduce shuffle bytes=146
                Reduce input records=10
                Reduce output records=1
                Spilled Records=20
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=176
                CPU time spent (ms)=1650
                Physical memory (bytes) snapshot=334729216
                Virtual memory (bytes) snapshot=3008466944
                Total committed heap usage (bytes)=226365440
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=21
        File Output Format Counters
                Bytes Written=8
```

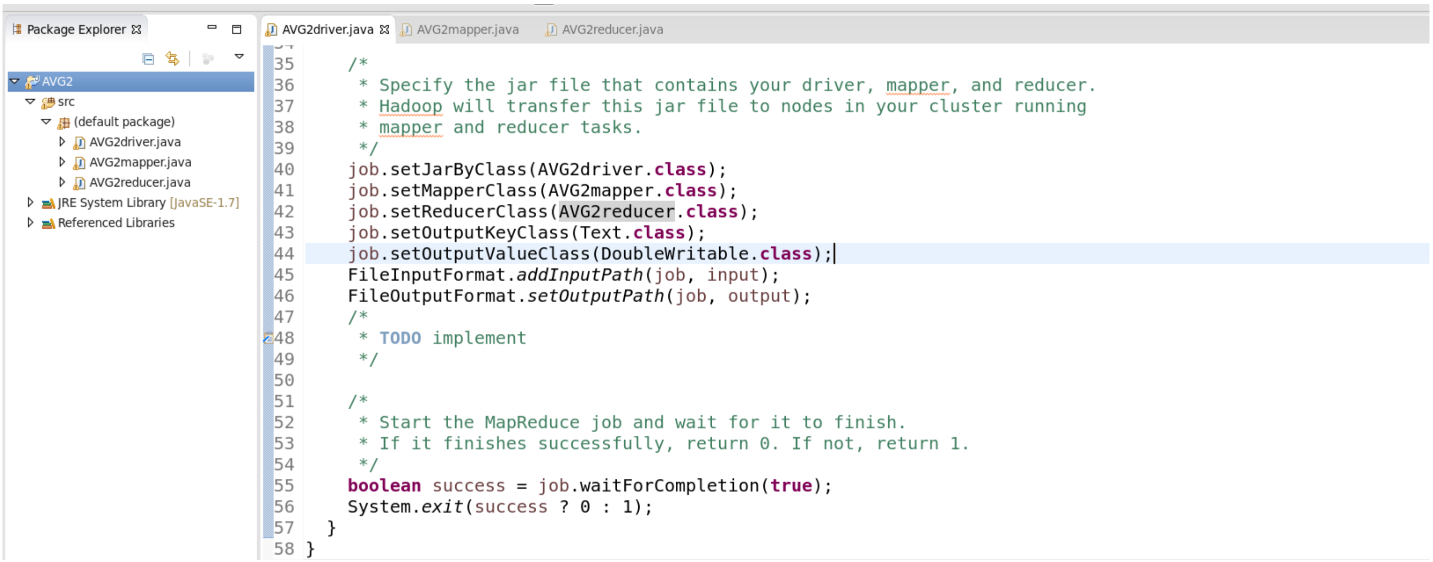
Output:



```
Avg	5.5
```

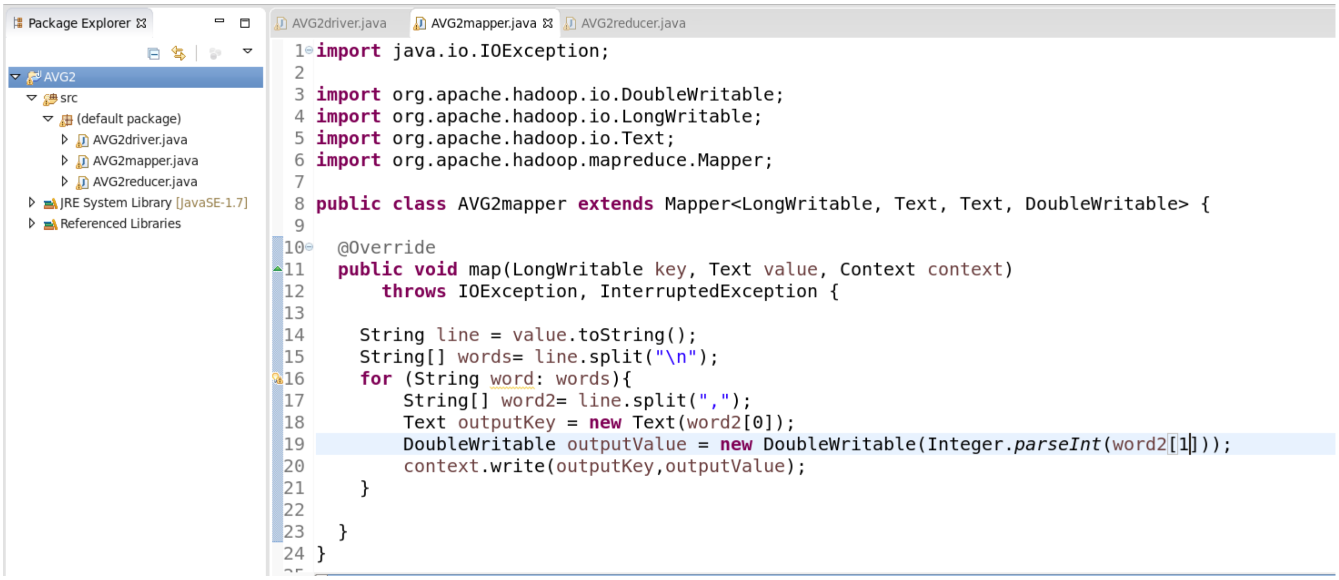
Problem 2:

Driver class:



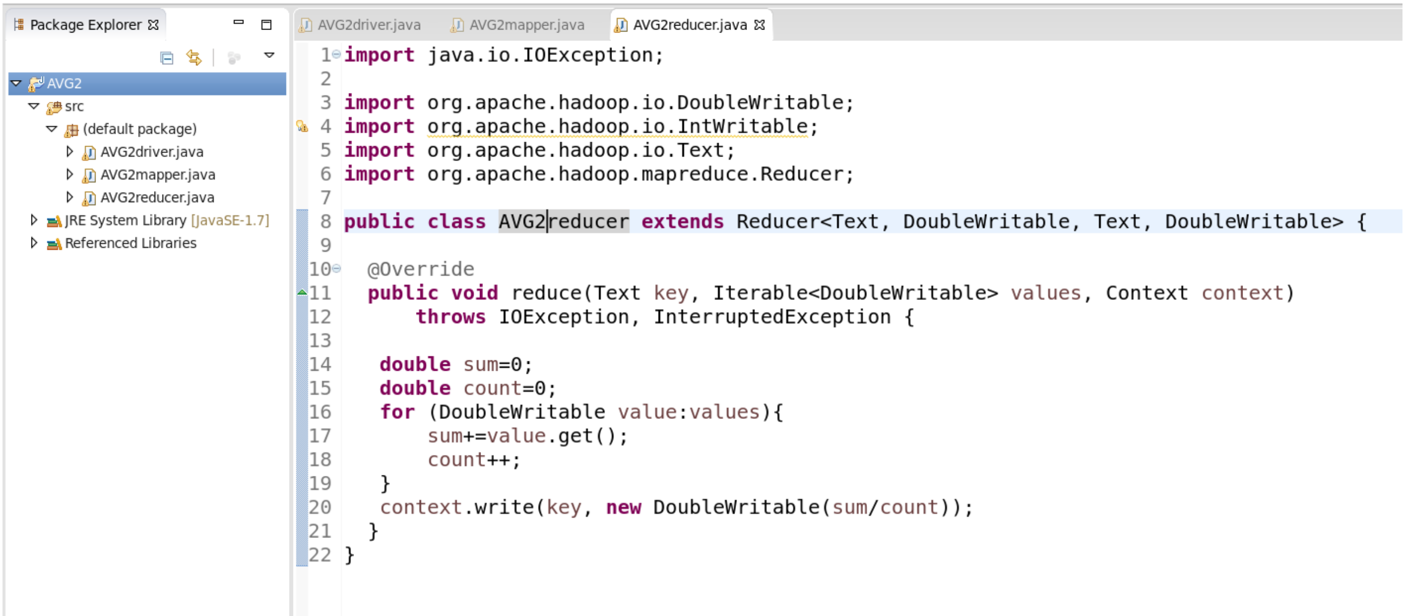
```
    /*
     * Specify the jar file that contains your driver, mapper, and reducer.
     * Hadoop will transfer this jar file to nodes in your cluster running
     * mapper and reducer tasks.
     */
    job.setJarByClass(AVG2driver.class);
    job.setMapperClass(AVG2mapper.class);
    job.setReducerClass(AVG2reducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(DoubleWritable.class);
    FileInputFormat.addInputPath(job, input);
    FileOutputFormat.setOutputPath(job, output);
    /*
     * TODO implement
     */

    /*
     * Start the MapReduce job and wait for it to finish.
     * If it finishes successfully, return 0. If not, return 1.
     */
    boolean success = job.waitForCompletion(true);
    System.exit(success ? 0 : 1);
  }
}
```

Mapper class:



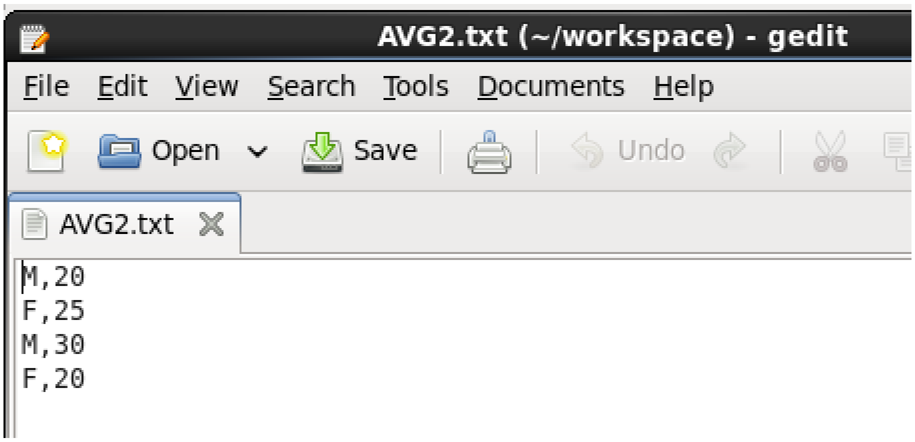
```java
import java.io.IOException;

import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class AVG2mapper extends Mapper<LongWritable, Text, Text, DoubleWritable> {

  @Override
  public void map(LongWritable key, Text value, Context context)
      throws IOException, InterruptedException {

    String line = value.toString();
    String[] words= line.split("\n");
    for (String word: words){
        String[] word2= line.split(",");
        Text outputKey = new Text(word2[0]);
        DoubleWritable outputValue = new DoubleWritable(Integer.parseInt(word2[1]));
        context.write(outputKey,outputValue);
    }

  }
}
```

Reducer class:



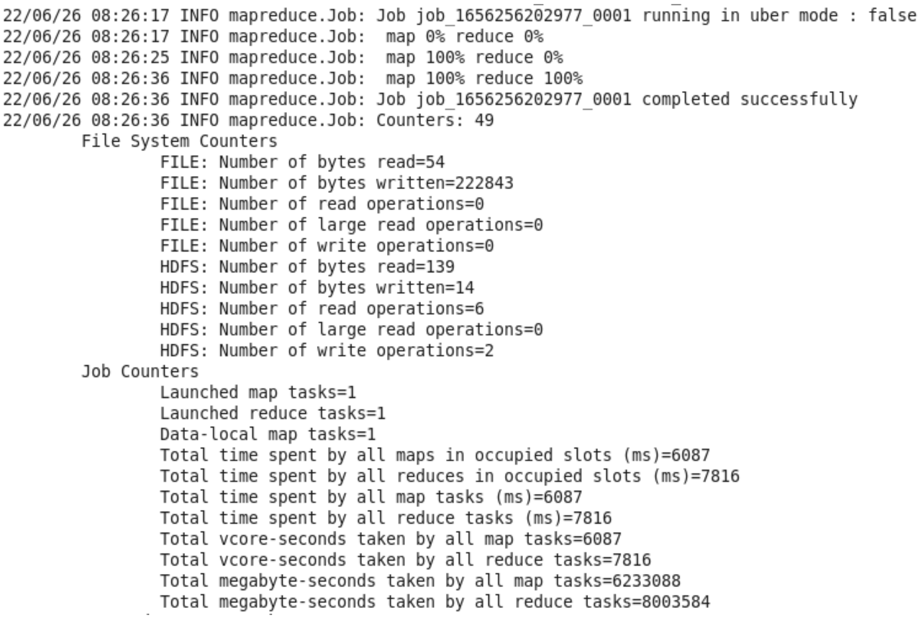
```java
import java.io.IOException;

import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class AVG2reducer extends Reducer<Text, DoubleWritable, Text, DoubleWritable> {

  @Override
  public void reduce(Text key, Iterable<DoubleWritable> values, Context context)
      throws IOException, InterruptedException {

   double sum=0;
   double count=0;
   for (DoubleWritable value:values){
       sum+=value.get();
       count++;
   }
   context.write(key, new DoubleWritable(sum/count));
  }
}
```

Input File:

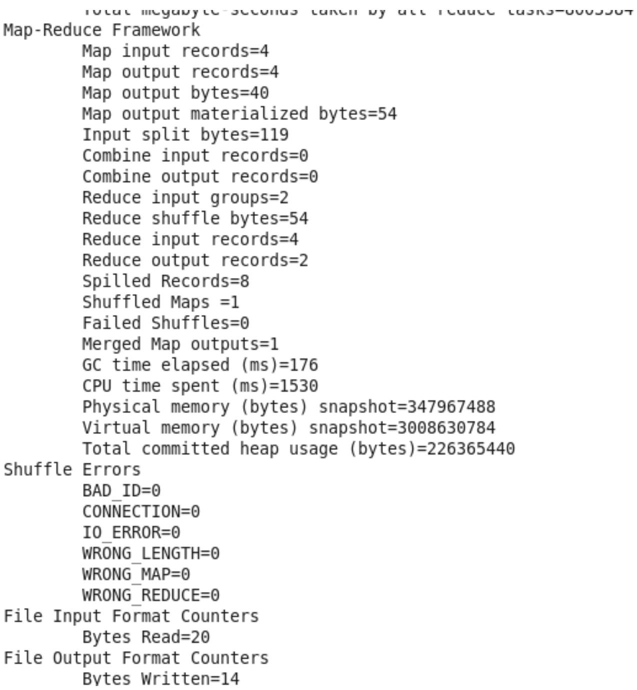

AVG2.txt (~/workspace) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
AVG2.txt

```
M,20
F,25
M,30
F,20
```

Terminal:

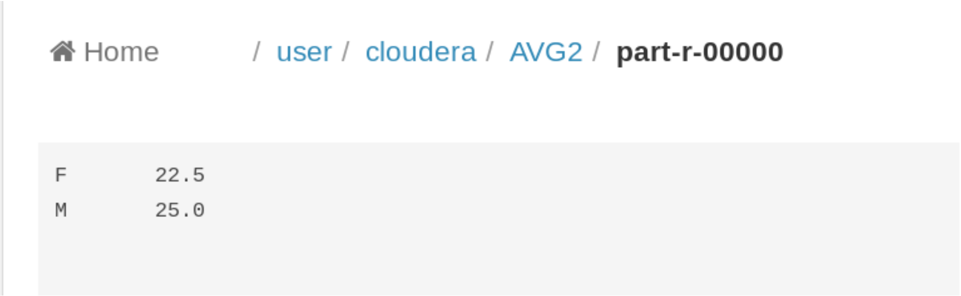


```
22/06/26 08:26:17 INFO mapreduce.Job: Job job_1656256202977_0001 running in uber mode : false
22/06/26 08:26:17 INFO mapreduce.Job:  map 0% reduce 0%
22/06/26 08:26:25 INFO mapreduce.Job:  map 100% reduce 0%
22/06/26 08:26:36 INFO mapreduce.Job:  map 100% reduce 100%
22/06/26 08:26:36 INFO mapreduce.Job: Job job_1656256202977_0001 completed successfully
22/06/26 08:26:36 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=54
                FILE: Number of bytes written=222843
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=139
                HDFS: Number of bytes written=14
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=6087
                Total time spent by all reduces in occupied slots (ms)=7816
                Total time spent by all map tasks (ms)=6087
                Total time spent by all reduce tasks (ms)=7816
                Total vcore-seconds taken by all map tasks=6087
                Total vcore-seconds taken by all reduce tasks=7816
                Total megabyte-seconds taken by all map tasks=6233088
                Total megabyte-seconds taken by all reduce tasks=8003584
```



```
                Total megabyte-seconds taken by all reduce tasks=[illegible]
        Map-Reduce Framework
                Map input records=4
                Map output records=4
                Map output bytes=40
                Map output materialized bytes=54
                Input split bytes=119
                Combine input records=0
                Combine output records=0
                Reduce input groups=2
                Reduce shuffle bytes=54
                Reduce input records=4
                Reduce output records=2
                Spilled Records=8
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=176
                CPU time spent (ms)=1530
                Physical memory (bytes) snapshot=347967488
                Virtual memory (bytes) snapshot=3008630784
                Total committed heap usage (bytes)=226365440
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=20
        File Output Format Counters
                Bytes Written=14
```

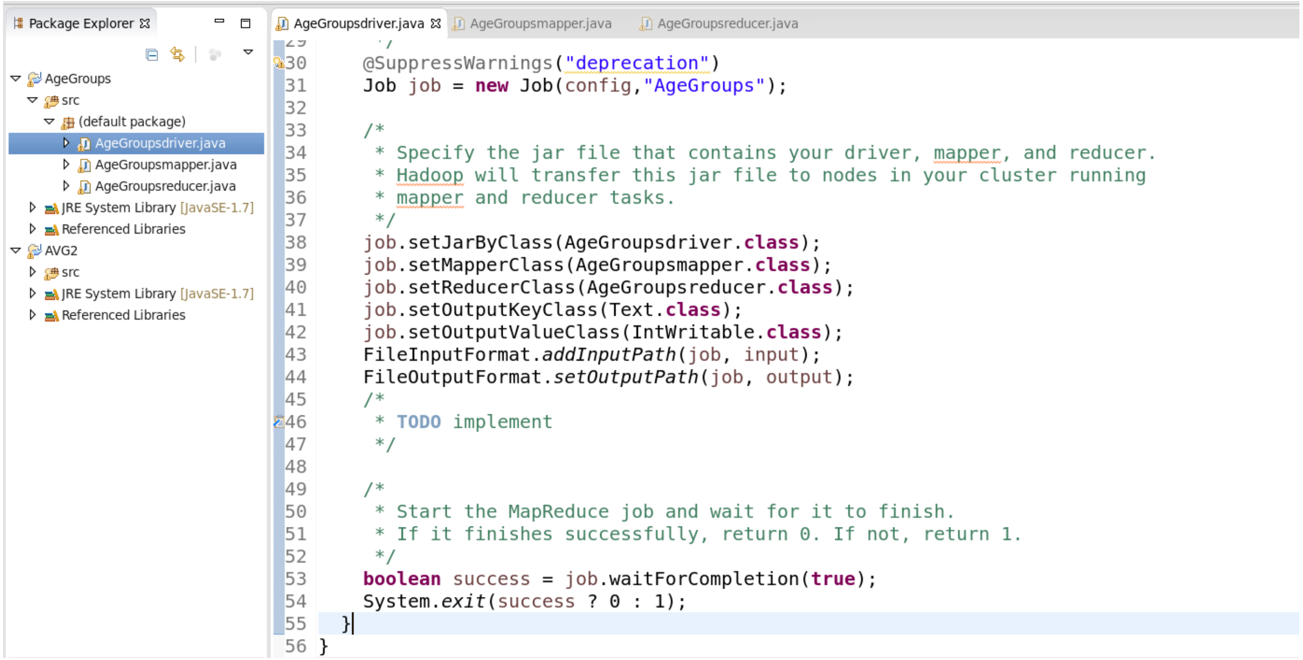
Output:



```
F	22.5
M	25.0
```

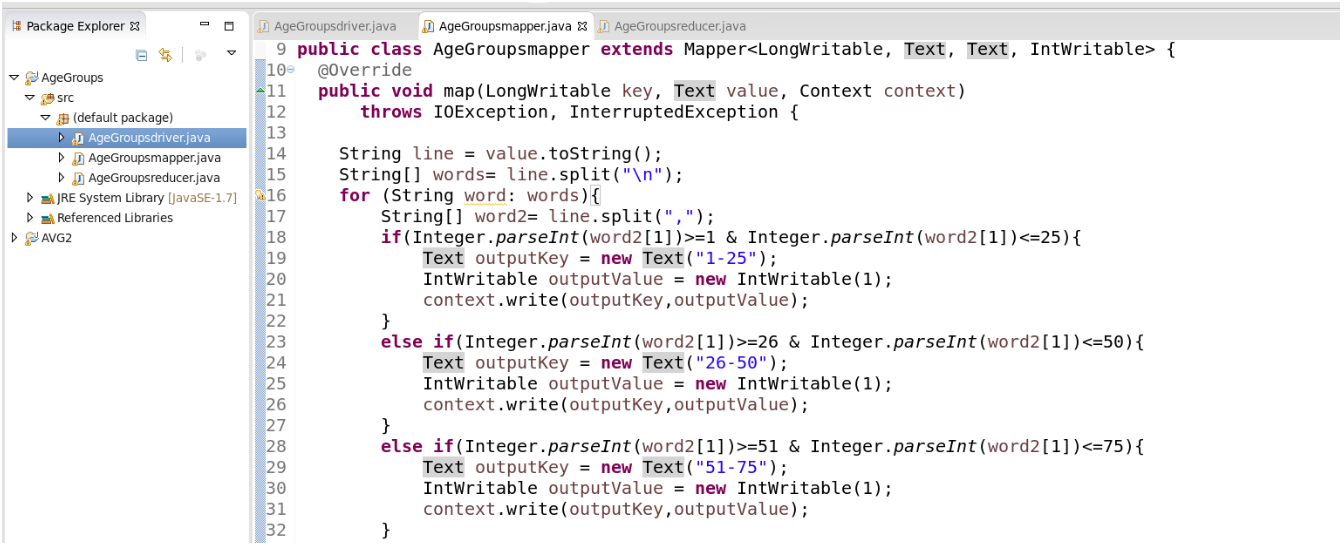
Problem 3:

Driver class:

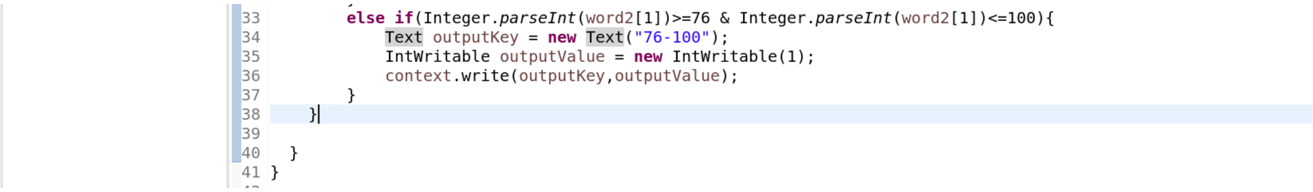


```
     */
    @SuppressWarnings("deprecation")
    Job job = new Job(config,"AgeGroups");

    /*
     * Specify the jar file that contains your driver, mapper, and reducer.
     * Hadoop will transfer this jar file to nodes in your cluster running
     * mapper and reducer tasks.
     */
    job.setJarByClass(AgeGroupsdriver.class);
    job.setMapperClass(AgeGroupsmapper.class);
    job.setReducerClass(AgeGroupsreducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, input);
    FileOutputFormat.setOutputPath(job, output);
    /*
     * TODO implement
     */

    /*
     * Start the MapReduce job and wait for it to finish.
     * If it finishes successfully, return 0. If not, return 1.
     */
    boolean success = job.waitForCompletion(true);
    System.exit(success ? 0 : 1);
  }
}
```

Mapper class:

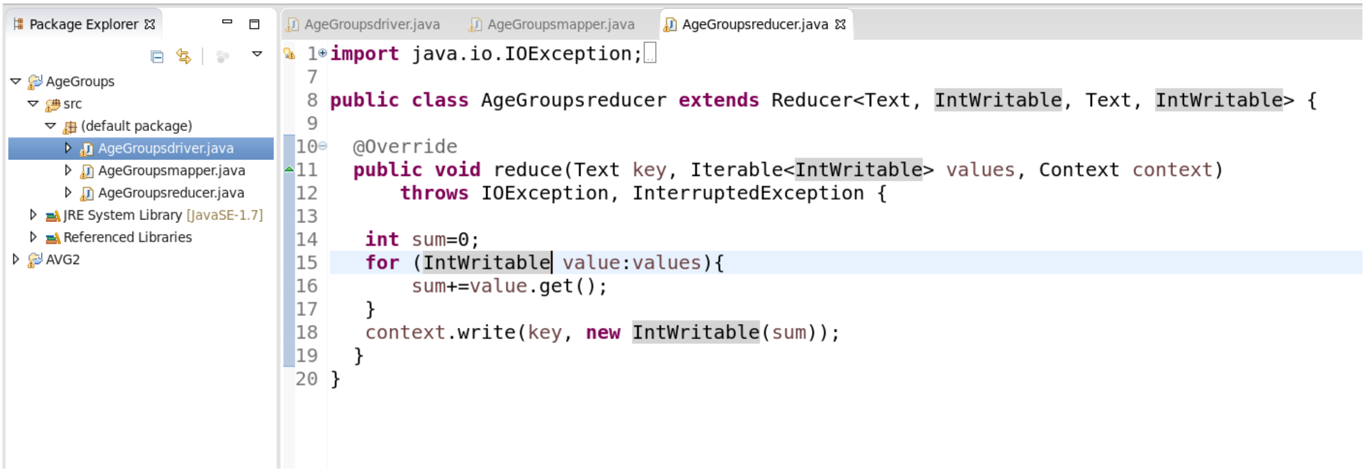


```
public class AgeGroupsmapper extends Mapper<LongWritable, Text, Text, IntWritable> {
  @Override
  public void map(LongWritable key, Text value, Context context)
      throws IOException, InterruptedException {

    String line = value.toString();
    String[] words= line.split("\n");
    for (String word: words){
        String[] word2= line.split(",");
        if(Integer.parseInt(word2[1])>=1 & Integer.parseInt(word2[1])<=25){
            Text outputKey = new Text("1-25");
            IntWritable outputValue = new IntWritable(1);
            context.write(outputKey,outputValue);
        }
        else if(Integer.parseInt(word2[1])>=26 & Integer.parseInt(word2[1])<=50){
            Text outputKey = new Text("26-50");
            IntWritable outputValue = new IntWritable(1);
            context.write(outputKey,outputValue);
        }
        else if(Integer.parseInt(word2[1])>=51 & Integer.parseInt(word2[1])<=75){
            Text outputKey = new Text("51-75");
            IntWritable outputValue = new IntWritable(1);
            context.write(outputKey,outputValue);
        }
```



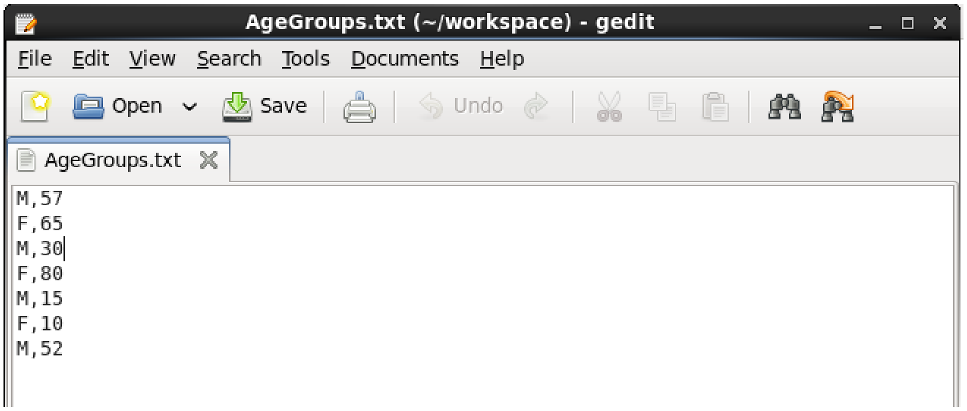
```
        else if(Integer.parseInt(word2[1])>=76 & Integer.parseInt(word2[1])<=100){
            Text outputKey = new Text("76-100");
            IntWritable outputValue = new IntWritable(1);
            context.write(outputKey,outputValue);
        }
    }


  }
}
```

Reducer class:



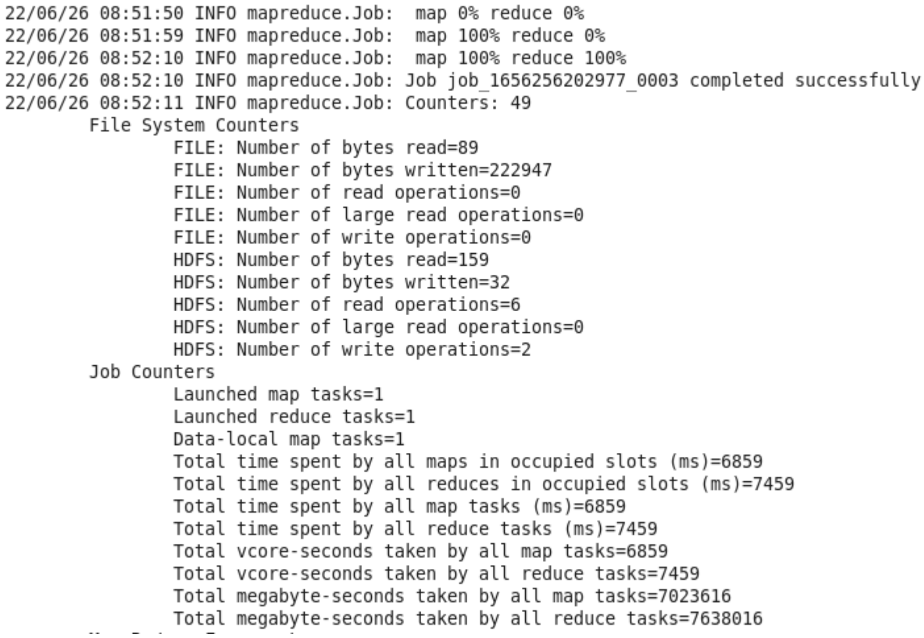
```java
import java.io.IOException;

public class AgeGroupsreducer extends Reducer<Text, IntWritable, Text, IntWritable> {

  @Override
  public void reduce(Text key, Iterable<IntWritable> values, Context context)
      throws IOException, InterruptedException {

   int sum=0;
   for (IntWritable value:values){
       sum+=value.get();
   }
   context.write(key, new IntWritable(sum));
  }
}
```

Input File:

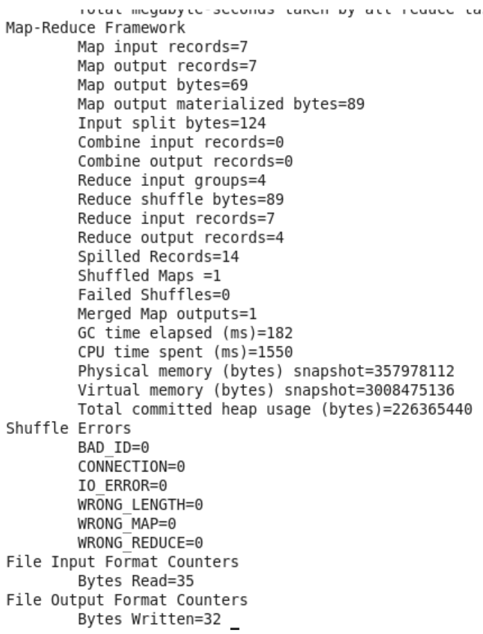


```
M,57
F,65
M,30
F,80
M,15
F,10
M,52
```

Terminal:

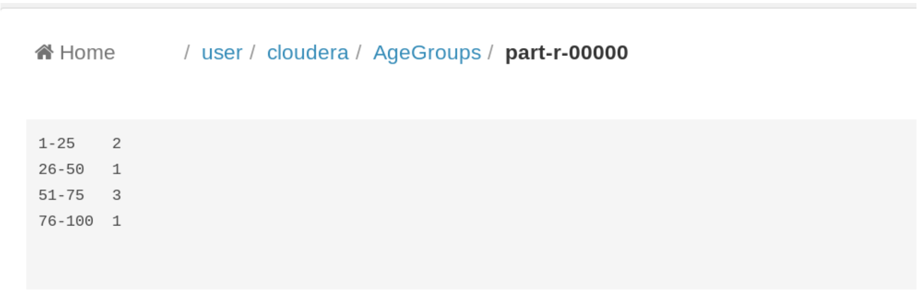


```
22/06/26 08:51:50 INFO mapreduce.Job:  map 0% reduce 0%
22/06/26 08:51:59 INFO mapreduce.Job:  map 100% reduce 0%
22/06/26 08:52:10 INFO mapreduce.Job:  map 100% reduce 100%
22/06/26 08:52:10 INFO mapreduce.Job: Job job_1656256202977_0003 completed successfully
22/06/26 08:52:11 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=89
                FILE: Number of bytes written=222947
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=159
                HDFS: Number of bytes written=32
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=6859
                Total time spent by all reduces in occupied slots (ms)=7459
                Total time spent by all map tasks (ms)=6859
                Total time spent by all reduce tasks (ms)=7459
                Total vcore-seconds taken by all map tasks=6859
                Total vcore-seconds taken by all reduce tasks=7459
                Total megabyte-seconds taken by all map tasks=7023616
                Total megabyte-seconds taken by all reduce tasks=7638016
```



```
        Map-Reduce Framework
                Map input records=7
                Map output records=7
                Map output bytes=69
                Map output materialized bytes=89
                Input split bytes=124
                Combine input records=0
                Combine output records=0
                Reduce input groups=4
                Reduce shuffle bytes=89
                Reduce input records=7
                Reduce output records=4
                Spilled Records=14
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=182
                CPU time spent (ms)=1550
                Physical memory (bytes) snapshot=357978112
                Virtual memory (bytes) snapshot=3008475136
                Total committed heap usage (bytes)=226365440
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=35
        File Output Format Counters
                Bytes Written=32 _
```

Output:



```
1-25    2
26-50   1
51-75   3
76-100  1
```

Learning Outcome:

From this lab we have learned how to solve problems by using MapReduce programming style and execute the program in Hadoop Framework.