

```
1 import java.io.IOException;
2 import java.util.Scanner;
3 import org.apache.hadoop.fs.Path;
4 import org.apache.hadoop.io.*;
5 import org.apache.hadoop.mapreduce.*;
6 import org.apache.hadoop.mapreduce.lib.input.*;
7 import org.apache.hadoop.mapreduce.lib.output.*;
8
9 public class Simple {
10     public static class MyMapper extends Mapper<
11         Object,Text,IntWritable,DoubleWritable> {
12         @Override
13         public void map ( Object key, Text value,
14             Context context )
15             throws IOException,
16             InterruptedException {
17             Scanner s = new Scanner(value.toString
18                 ()).useDelimiter(",");
19             int x = s.nextInt();
20             double y = s.nextDouble();
21             context.write(new IntWritable(x),new
22                 DoubleWritable(y));
23             s.close();
24         }
25     }
26
27     public static class MyReducer extends Reducer<
28         IntWritable,DoubleWritable,IntWritable,
29         DoubleWritable> {
30         @Override
31         public void reduce ( IntWritable key,
32             Iterable<DoubleWritable> values, Context context )
33             throws IOException,
34             InterruptedException {
35             double sum = 0.0;
36             long count = 0;
37             for (DoubleWritable v: values) {
38                 sum += v.get();
39                 count++;
40             };
41             context.write(key,new DoubleWritable(
```

```
32 sum/count));
33     }
34 }
35
36     public static void main ( String[] args )
37     throws Exception {
38         Job job = Job.getInstance();
39         job.setJobName("MyJob");
40         job.setJarByClass(Simple.class);
41         job.setOutputKeyClass(IntWritable.class);
42         job.setOutputValueClass(DoubleWritable.
43     class);
44         job.setMapOutputKeyClass(IntWritable.class
45     );
46         job.setMapOutputValueClass(DoubleWritable.
47     class);
48         job.setMapperClass(MyMapper.class);
49         job.setReducerClass(MyReducer.class);
50         job.setInputFormatClass(TextInputFormat.
51     class);
52         job.setOutputFormatClass(TextOutputFormat.
53     class);
54         FileInputFormat.setInputPaths(job,new Path(
55     args[0]));
56         FileOutputFormat.setOutputPath(job,new Path
57     (args[1]));
58         job.waitForCompletion(true);
59     }
60 }
```