**package** org.myorg;

**import java.io.IOException**;

**import java.util.\***;

**import org.apache.hadoop.fs.Path**;

**import org.apache.hadoop.conf.\***;

**import org.apache.hadoop.io.\***;

**import org.apache.hadoop.mapreduce.\***;

**import org.apache.hadoop.mapreduce.lib.input.FileInputFormat**;

**import org.apache.hadoop.mapreduce.lib.input.TextInputFormat**;

**import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat**;

**import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat**;

**public class WordCount** {

**public static class Map extends** Mapper<LongWritable, Text, Text, IntWritable> {

**private final static** IntWritable one = **new** IntWritable(1);

**private** Text word = **new** Text();

**public void** map(LongWritable key, Text value, Context context)

**throws** IOException, InterruptedException {

String line = value.toString();

StringTokenizer tokenizer = **new** StringTokenizer(line);

**while** (tokenizer.hasMoreTokens()) {

word.set(tokenizer.nextToken());

context.write(word, one);

}

}

}

**public static class Reduce extends** Reducer<Text, IntWritable, Text, IntWritable> {

**public void** reduce(Text key, Iterable<IntWritable> values, Context context)

**throws** IOException, InterruptedException {

**int** sum = 0;

**for** (IntWritable val : values) {

sum += val.get();

}

context.write(key, **new** IntWritable(sum));

}

**public static void** main(String[] args) **throws** Exception {

Configuration conf = **new** Configuration();

Job job = **new** Job(conf, "wordcount");

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

job.setMapperClass(Map.class);

job.setReducerClass(Reduce.class);

job.setInputFormatClass(TextInputFormat.class);

job.setOutputFormatClass(TextOutputFormat.class);

FileInputFormat.addInputPath(job, **new** Path(args[0]));

FileOutputFormat.setOutputPath(job, **new** Path(args[1]));

job.waitForCompletion(**true**);

}

}