WordCount.java

```java
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

  public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context
                ) throws IOException, InterruptedException {
      StringTokenizer itr = new StringTokenizer(value.toString());
      while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
      }
    }
  }


    public static class IntSumReducer
       extends Reducer<Text,IntWritable,Text,IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values,
                Context context
                ) throws IOException, InterruptedException {
      int sum = 0;
      for (IntWritable val : values) {
        sum += val.get();
      }
      result.set(sum);
      context.write(key, result);
    }
  }

  public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
```

```
      job.setCombinerClass(IntSumReducer.class);
      job.setReducerClass(IntSumReducer.class);
      job.setOutputKeyClass(Text.class);
      job.setOutputValueClass(IntWritable.class);
      FileInputFormat.addInputPath(job, new Path(args[0]));
      FileOutputFormat.setOutputPath(job, new Path(args[1]));
      System.exit(job.waitForCompletion(true) ? 0: 1);
   }
}
```

output: output file (part-r-00000)

```
11 For       1
12 Gibson  1
13 Magazine.    1
14 Map       1
15 Project     1
16 Reduce  1
17 Shadow  2
18 Shadow,      1
19 Smith   1
20 Smith's      1
21 Story   1
22 Street  2
23 The     4
24 Walter  1
25 WordCount   1
26 a       4
27 about   1
28 an      1
29 and     4
30 be      1
31 broadcasts   1
32 by      2
33 can     1
34 contained    1
35 contains     1
36 count   1
37 counts  1
38 crime-fighting 1
39 data    1
40 dated   1
41 different    1
42 each    1
43 evil    1
44 example      1
45 figure  1
46 files   2
47 files,  1
```

Open | ▼ | +   part-r-00000   Save | ≡ | — | □ | ✕
~/Desktop/Wordcountexp

Plain Text ▼   Tab Width: 8 ▼   Ln 22, Col 10   ▼   INS