WordCount.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().replaceAll("[^a-zA-Z0-9]",
"").toLowerCase());
                context.write(word, one);
            }
        }
    }
```

```java
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);
    }
}
```

Input.txt

Hadoop is an open-source framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models.

Hadoop is designed to scale up from single servers to thousands of machines, each offering local computation and storage.

Hello world! Welcome to the world of Big Data with Hadoop and MapReduce.

Output:

```
across    1
allows    1
and  3
an   1
big  1
clusters  1
computation    1
computers 1
data 2
designed  1
distributed    1
each 1
framework 1
for  1
from 1
```

```
hadoop    3
hello     1
is   2
large     1
local     1
machines  1
mapreduce 1
models    1
of   4
offering  1
open 1
processing     1
programming    1
scale     1
servers   1
sets 1
simple    1
single    1
source    1
storage   1
that 1
the  3
thousands 1
to   3
up   1
using     1
welcome   1
with 1
world     2
```