

```

package it.polito.bigdata.hadoop.exercise20;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.Job;
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.MultipleOutputs;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

/**
 * Driver class.
 */
public class DriverBigData extends Configured implements Tool {

    @Override
    public int run(String[] args) throws Exception {

        Path inputPath;
        Path outputDir;
        int exitCode;

        // Parse the parameters
        inputPath = new Path(args[0]);
        outputDir = new Path(args[1]);

        Configuration conf = this.getConf();

        // Define a new job
        Job job = Job.getInstance(conf);

        // Assign a name to the job
        job.setJobName("Exercise 20");

        // Set path of the input file/folder (if it is a folder, the job reads all the
        // files in the specified folder) for this job
        FileInputFormat.addInputPath(job, inputPath);

        // Set path of the output folder for this job
        FileOutputFormat.setOutputPath(job, outputDir);

        // Set multiple outputs. The name of the output files will have
        // the prefix specified in the mapper
        // "hightemp" and "normaltemp" are the "names" of the two prefixes/outputs
        // Set also map output key and value classes
        MultipleOutputs.addNamedOutput(job, "hightemp", TextOutputFormat.class,
        Text.class, NullWritable.class);
        MultipleOutputs.addNamedOutput(job, "normaltemp", TextOutputFormat.class,
        Text.class, NullWritable.class);

        // Specify the class of the Driver for this job
        job.setJarByClass(DriverBigData.class);
    }
}

```

```

// Set input format
job.setInputFormatClass(TextInputFormat.class);

// Set map class
job.setMapperClass(MapperBigData.class);

// Set number of reducers
job.setNumReduceTasks(0);

// Execute the job and wait for completion
if (job.waitForCompletion(true)==true)
    exitCode=0;
else
    exitCode=1;

return exitCode;
}

/** Main of the driver
 */

public static void main(String args[]) throws Exception {
    // Exploit the ToolRunner class to "configure" and run the Hadoop application
    int res = ToolRunner.run(new Configuration(), new DriverBigData(), args);

    System.exit(res);
}
}

```