Assignment No. 12

Code:

```
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.FloatWritable;
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.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class WeatherDataAverage {

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

    private Text category = new Text();
    private FloatWritable temperature = new FloatWritable();
    private FloatWritable windSpeed = new FloatWritable();
    private FloatWritable dewPoint = new FloatWritable();

    public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
      String[] cols = value.toString().split(" ");
      float temp = Float.parseFloat(cols[0]);
      float wind = Float.parseFloat(cols[1]);
      float dew = Float.parseFloat(cols[2]);

      category.set("Temperature");
      temperature.set(temp);
      context.write(category, temperature);

      category.set("WindSpeed");
      windSpeed.set(wind);
      context.write(category, windSpeed);

      category.set("DewPoint");
      dewPoint.set(dew);
      context.write(category, dewPoint);
    }
  }

  public static class FloatAverageReducer extends Reducer<Text, FloatWritable, Text, FloatWritable> {
    private FloatWritable result = new FloatWritable();
```

```
        public void reduce(Text key, Iterable<FloatWritable> values, Context context)
                throws IOException, InterruptedException {
            float sum = 0;
            int count = 0;
            for (FloatWritable a : values){sum += a.get();count++;}
            float avg = sum / count;
            result.set(avg);
            context.write(key, result);
        }
    }

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "weather data average");
        job.setJarByClass(WeatherDataAverage.class);
        job.setMapperClass(TokenizerMapper.class);
        job.setReducerClass(FloatAverageReducer.class);
        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(FloatWritable.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(FloatWritable.class);
        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);
        TextInputFormat.addInputPath(job, new Path(args[0]));
        TextOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

Output:

```
$ hadoop jar WeatherDataAverage.jar WeatherDataAverage weather.txt weather-output

2025-04-20 18:45:38,417 INFO mapreduce.Job: Job job_1713626600243_0002 completed successfully
...
$ cat weather-output/part-r-00000
DewPoint       19.833334
Temperature    27.833334
WindSpeed      12.500000
```