

nov. 25, 18 15:15

Anagrammes.java

Page 1/2

```

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;
import java.util.HashSet;
import java.util.Set;
import java.util.Map;

/*
 *      YOUSSEF NIDABRAHIM
 *      ZZ3 - F2
 *
 *      MAP/REDUCE PATTERN FOR COUNTING ANAGRAMMES
 */

public class Anagrammes {

    public static class AnagrammeMapper extends Mapper<Object, Text, Text, Text>{

        public void map(Object key, Text value, Context context) throws IOException,
        InterruptedException {

            char[] letters = value.toString().toLowerCase().toCharArray();
            Arrays.sort(letters);

            context.write(new Text(new String(letters)), value);

        }
    }

    public static class AnagrammeReducer extends Reducer<Text,Text,Text,Text> {

        public void reduce(Text key, Iterable<Text> values, Context context ) throws
        IOException, InterruptedException {

            Iterator<Text> i = values.iterator();
            String result = "";
            Boolean first = true;
            while(i.hasNext()){
                if(first){
                    result = i.next().toString();
                    first = false;
                }else
                    result = result+"|"+i.next().toString();
            }
            context.write(key, new Text(result));
        }
    }

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

        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "anagrammes counter");

```

nov. 25, 18 15:15

Anagrammes.java

Page 2/2

```

        job.setJarByClass(Anagrammes.class);
        job.setMapperClass(AnagrammeMapper.class);
        job.setReducerClass(AnagrammeReducer.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);
    }
}

```