

```

package it.polito.bigdata.spark.wordcount;

import java.util.Arrays;
import org.apache.spark.api.java.*;
import org.apache.spark.api.java.function.*;
import org.apache.spark.SparkConf;
import scala.Tuple2;

public class SparkWordCount {
    @SuppressWarnings("serial")
    public static void main(String[] args) {

        String inputFile;
        String outputPath;

        inputFile=args[0];
        outputPath=args[1];

        // Create a configuration object and set the name of the application
        SparkConf conf=new SparkConf().setAppName("Spark Line Count");

        // Create a Spark Context object
        JavaSparkContext sc = new JavaSparkContext(conf);

        // Build an RDD of Strings from the input textual file
        // Each element of the RDD is a line of the input file
        JavaRDD<String> lines=sc.textFile(inputFile);

        // Split/transform the content of the lines RDD in a
        // list words an store in the words RDD
        //
        JavaRDD<String> words = lines.flatMap(
            new FlatMapFunction<String, String>() {
                @Override
                public Iterable<String> call(String s) {
                    return Arrays.asList(s.split("\\s+"));
                }
            });

        // Map/transform each word in the words RDD to a pair (word,1)
        // an store in the words_one RDD
        JavaPairRDD<String, Integer> words_one = words.mapToPair(
            new PairFunction<String, String, Integer>() {
                @Override
                public Tuple2<String, Integer> call(String s) {
                    return new Tuple2<String,
Integer>(s.toLowerCase(), 1);
                }
            });

        // Count the occurrence of each word.
        // Reduce by key the pairs of the words_one RDD and store
        // the result (the list of pairs (word, num. of occurrences)
        // in the counts RDD
        JavaPairRDD<String, Integer> counts = words_one.reduceByKey(
            new Function2<Integer, Integer, Integer>() {
                @Override
                public Integer call(Integer i1, Integer i2) {

```

```
        return i1 + i2;
    });
}

// Store the result in the output folder
counts.saveAsTextFile(outputPath);

// Close the Spark Context object
sc.close();
}
}
```