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 bowler {

public static class TokenizerMapper1

extends Mapper<Object, Text, Text, IntWritable>{

private Text text = new Text();

public void map(Object key, Text value, Context context

) throws IOException, InterruptedException {

String line = value.toString();

String[] row= line.split(",");

String tuple=row[2]+","+row[4];

int run=Integer.parseInt(row[5]);

IntWritable word = new IntWritable(run);

word.set(tuple);

context.write(text,word);

}

}

public static class IntSumReducer

extends Reducer<Text,IntWritable,Text,IntWritable> {

private IntWritable res = new IntWritable();

public void reduce(Text key, Iterable<IntWritable> values,

Context context

) throws IOException, InterruptedException {

int result= 0;

for (IntWritable val : values) {

result += val.get();

}

result.set(result);

context.write(key, res);

}

}

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

Configuration config = new Configuration();

Job job1 = Job.getInstance(config, "word count");

job.setJarByClass(bowler.class);

job.setMapperClass(TokenizerMapper1.class);

// job.setCombinerClass(IntSumReducer.class);

job.setReducerClass(IntSumReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job1, new Path(args[0]));

FileOutputFormat.setOutputPath(job1, new Path(args[1]));

System.exit(job1.waitForCompletion(true) ? 0 : 1);

}

}