Write a custom key to evaluate the number of people who died and the number of people who survived, along with their genders.

import org.apache.hadoop.fs.Path;  
import org.apache.hadoop.conf.\*;  
import org.apache.hadoop.mapreduce.Job;  
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;  
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;  
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;  
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;  
import org.apache.hadoop.io.IntWritable;  
import org.apache.hadoop.io.NullWritable;  
import org.apache.hadoop.io.Text;  
public class TitanicDriver{  
    @SuppressWarnings("deprecation")  
    public static void main(String[] args)throws Exception  
    {  
        Configuration conf=new Configuration();  
        Job job=new Job(conf,"Tv sales");  
        job.setJarByClass(TitanicDriver.class);  
        FileInputFormat.addInputPath(job, new Path(args[0]));  
        Path outputPath =new Path(args[1]);  
        FileOutputFormat.setOutputPath(job, outputPath);  
        outputPath.getFileSystem(conf).delete(outputPath, true);  
        job.setMapperClass(TitanicMapper.class);  
        job.setReducerClass(TitanicReducer.class);  
        job.setInputFormatClass(TextInputFormat.class);  
        job.setOutputFormatClass(TextOutputFormat.class);  
        job.setMapOutputKeyClass(TCW.class);  
        job.setMapOutputValueClass(IntWritable.class);  
        job.setOutputKeyClass(TCW.class);  
        job.setOutputValueClass(IntWritable.class);  
        job.waitForCompletion(true);  
  
    }  
}import java.io.IOException;  
  
  
import org.apache.hadoop.io.IntWritable;  
import org.apache.hadoop.io.LongWritable;  
import org.apache.hadoop.io.Text;  
import org.apache.hadoop.mapreduce.\*;  
  
public class TitanicMapper extends Mapper<LongWritable, Text,TCW,IntWritable>{  
    TCW outkey=new TCW();  
    IntWritable outvalue=new IntWritable();  
    public void map(LongWritable key, Text value, Context context)  
    throws IOException, InterruptedException{  
        String[] lineArray = value.toString().split(",");  
        if(lineArray.length>1)  
        {  
        outkey.set(lineArray[4],Integer.parseInt(lineArray[1]));  
            outvalue.set(1);  
        }  
            context.write(outkey,outvalue);  
  
  
    }  
  
}import java.io.IOException;  
import java.util.StringTokenizer;  
  
import org.apache.hadoop.io.IntWritable;  
import org.apache.hadoop.io.LongWritable;  
import org.apache.hadoop.io.NullWritable;  
import org.apache.hadoop.io.Text;  
import org.apache.hadoop.mapreduce.\*;  
import org.apache.hadoop.mapreduce.Mapper.Context;  
public class TitanicReducer extends Reducer<TCW,IntWritable,TCW,IntWritable>{  
    public void reduce(TCW key, Iterable<IntWritable>values, Context context)  
            throws IOException, InterruptedException{  
        int sum=0;  
        for(IntWritable value: values)  
        {  
            sum=sum+value.get();  
        }  
        context.write(key, new IntWritable(sum));  
    }  
  
}import java.io.DataInput;  
import java.io.DataOutput;  
import java.io.IOException;  
  
import org.apache.hadoop.io.WritableComparable;  
  
public class TCW implements WritableComparable<TCW>{  
    private String company;  
    private int size;  
  
    public String getCompany() {  
        return company;  
    }  
    public int getSize() {  
        return size;  
    }  
  
  
  
    @Override  
    public void readFields(DataInput input) throws IOException {  
        // TODO Auto-generated method stub  
        company=input.readUTF();  
        size=input.readInt();  
  
    }  
  
    @Override  
    public void write(DataOutput output) throws IOException {  
        // TODO Auto-generated method stub  
        output.writeUTF(company);  
        output.writeInt(size);  
    }  
  
    public String toString()  
    {  
        return company+ "\t" +size;  
    }  
  
    @Override  
    public int compareTo(TCW TvCW) {  
        // TODO Auto-generated method stub  
        int cmp=company.compareTo(TvCW.company);  
        if(cmp!=0)  
        {  
            return cmp;  
        }  
        return (-1)\*(size - TvCW.getSize());  
    }  
    public int hashcode(){  
        return company.hashCode();  
    }  
    public boolean equals(Object o)  
    {  
        if(o instanceof TCW)  
        {  
            TCW TvCW =(TCW) o;  
            return company.equalsIgnoreCase(TvCW.company);  
        }  
        return false;  
        // TODO Auto-generated method stub  
  
    }  
    public void set(String company, int size) {  
        // TODO Auto-generated method stub  
        this.company=company;  
        this.size=size;  
    }  
  
  
}



