CLASS WordCounCREATEt.java

```java
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 WordCount {

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

       private final static IntWritable one = new IntWritable(1);
       private Text word = new Text();

       public void map(Object key, Text value, Context context
               ) throws IOException, InterruptedException {
       StringTokenizer itr = new StringTokenizer(value.toString());
       while (itr.hasMoreTokens()) {
       word.set(itr.nextToken());
       context.write(word, one);
       }
       }
  }

  public static class IntSumReducer
       extends Reducer<Text,IntWritable,Text,IntWritable> {
       private IntWritable result = new IntWritable();

       public void reduce(Text key, Iterable<IntWritable> values,
               Context context
               ) throws IOException, InterruptedException {
       int sum = 0;
       for (IntWritable val : values) {
       sum += val.get();
       }
       result.set(sum);
```

```java
            context.write(key, result);
        }
    }

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "word count");
        job.setJarByClass(WordCount.class);
        job.setMapperClass(TokenizerMapper.class);
        job.setCombinerClass(IntSumReducer.class);
        job.setReducerClass(IntSumReducer.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);
    }
}
```

TERMINAL

```
[cloudera@quickstart ~]$ ls
cloudera-manager  eclipse                    Music          Templates
cm_api.py         enterprise-deployment.json parcels        Videos
Desktop           express-deployment.json    Pictures       WordCount.jar
Documents         kerberos                   Processfile.txt WordCountjar.jar
Downloads         lib                        Public         workspace
[cloudera@quickstart ~]$ pwd
/home/cloudera
[cloudera@quickstart ~]$ cat > /home/cloudera/devansh.txt
devansh
jay shree ram
bharat mata ke jay
hp boss
om papadiwal
rohit patil
^Z
[1]+  Stopped          cat > /home/cloudera/devansh.txt
[cloudera@quickstart ~]$ cat /home/cloudera/devansh.txt
devansh
jay shree ram
bharat mata ke jay
hp boss
om papadiwal
rohit patil
[cloudera@quickstart ~]$ hdfs dfs -ls
[cloudera@quickstart ~]$ hdfs dfs -ls /
```

```
Found 7 items
drwxrwxrwx   - hdfs      supergroup          0 2017-10-23 09:15 /benchmarks
drwxr-xr-x   - hbase     supergroup          0 2025-04-22 01:22 /hbase
drwxr-xr-x   - cloudera supergroup           0 2025-03-27 23:38 /inputfolder1
drwxr-xr-x   - solr      solr         0 2017-10-23 09:18 /solr
drwxrwxrwt   - hdfs      supergroup          0 2025-03-26 22:24 /tmp
drwxr-xr-x   - hdfs      supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs      supergroup          0 2017-10-23 09:17 /var
[cloudera@quickstart ~]$ hdfs dfs -mkdir /input
[cloudera@quickstart ~]$ hdfs dfs -put /home/cloudera/devansh.txt /input/
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 8 items
drwxrwxrwx   - hdfs      supergroup          0 2017-10-23 09:15 /benchmarks
drwxr-xr-x   - hbase     supergroup          0 2025-04-22 01:22 /hbase
drwxr-xr-x   - cloudera supergroup           0 2025-04-22 01:33 /input
drwxr-xr-x   - cloudera supergroup           0 2025-03-27 23:38 /inputfolder1
drwxr-xr-x   - solr      solr         0 2017-10-23 09:18 /solr
drwxrwxrwt   - hdfs      supergroup          0 2025-03-26 22:24 /tmp
drwxr-xr-x   - hdfs      supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs      supergroup          0 2017-10-23 09:17 /var
[cloudera@quickstart ~]$ hdfs dfs -cat /input/devansh.txt
devansh
jay shree ram
bharat mata ke jay
hp boss
om papadiwal
rohit patil
[cloudera@quickstart ~]$ hadoop jar /home/cloudera/WordCount.jar WordCount
/input/devansh.txt /out1
25/04/22 01:36:50 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
25/04/22 01:36:51 WARN mapreduce.JobResourceUploader: Hadoop command-line option
parsing not performed. Implement the Tool interface and execute your application with
ToolRunner to remedy this.
25/04/22 01:36:52 INFO input.FileInputFormat: Total input paths to process : 1
25/04/22 01:36:52 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
    at java.lang.Object.wait(Native Method)
    at java.lang.Thread.join(Thread.java:1281)
    at java.lang.Thread.join(Thread.java:1355)
    at
org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStre
am.java:967)
    at
org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.jav
a:705)
    at
org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)
25/04/22 01:36:52 WARN hdfs.DFSClient: Caught exception
```

java.lang.InterruptedException
    at java.lang.Object.wait(Native Method)
    at java.lang.Thread.join(Thread.java:1281)
    at java.lang.Thread.join(Thread.java:1355)
    at
org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStre
am.java:967)
    at
org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.jav
a:705)
    at
org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)
25/04/22 01:36:52 INFO mapreduce.JobSubmitter: number of splits:1
25/04/22 01:36:53 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1745309395132_0001
25/04/22 01:36:55 INFO impl.YarnClientImpl: Submitted application
application_1745309395132_0001
25/04/22 01:36:55 INFO mapreduce.Job: The url to track the job:
http://quickstart.cloudera:8088/proxy/application_1745309395132_0001/
25/04/22 01:36:55 INFO mapreduce.Job: Running job: job_1745309395132_0001
25/04/22 01:37:16 INFO mapreduce.Job: Job job_1745309395132_0001 running in uber
mode : false
25/04/22 01:37:16 INFO mapreduce.Job: map 0% reduce 0%
25/04/22 01:37:47 INFO mapreduce.Job: map 100% reduce 0%
25/04/22 01:38:05 INFO mapreduce.Job: map 100% reduce 100%
25/04/22 01:38:06 INFO mapreduce.Job: Job job_1745309395132_0001 completed
successfully
25/04/22 01:38:07 INFO mapreduce.Job: Counters: 49
    File System Counters
        FILE: Number of bytes read=154
        FILE: Number of bytes written=286987
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=188
        HDFS: Number of bytes written=96
        HDFS: Number of read operations=6
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=2
    Job Counters
        Launched map tasks=1
        Launched reduce tasks=1
        Data-local map tasks=1
        Total time spent by all maps in occupied slots (ms)=27952
        Total time spent by all reduces in occupied slots (ms)=14497
        Total time spent by all map tasks (ms)=27952
        Total time spent by all reduce tasks (ms)=14497
        Total vcore-milliseconds taken by all map tasks=27952

```
            Total vcore-milliseconds taken by all reduce tasks=14497
            Total megabyte-milliseconds taken by all map tasks=28622848
            Total megabyte-milliseconds taken by all reduce tasks=14844928
    Map-Reduce Framework
            Map input records=6
            Map output records=14
            Map output bytes=130
            Map output materialized bytes=154
            Input split bytes=114
            Combine input records=14
            Combine output records=13
            Reduce input groups=13
            Reduce shuffle bytes=154
            Reduce input records=13
            Reduce output records=13
            Spilled Records=26
            Shuffled Maps =1
            Failed Shuffles=0
            Merged Map outputs=1
            GC time elapsed (ms)=401
            CPU time spent (ms)=2470
            Physical memory (bytes) snapshot=342106112
            Virtual memory (bytes) snapshot=3015086080
            Total committed heap usage (bytes)=226365440
    Shuffle Errors
            BAD_ID=0
            CONNECTION=0
            IO_ERROR=0
            WRONG_LENGTH=0
            WRONG_MAP=0
            WRONG_REDUCE=0
    File Input Format Counters
            Bytes Read=74
    File Output Format Counters
            Bytes Written=96
[cloudera@quickstart ~]$ hdfs dfs -ls /out1
Found 2 items
-rw-r--r--   1 cloudera supergroup        0 2025-04-22 01:38 /out1/_SUCCESS
-rw-r--r--   1 cloudera supergroup       96 2025-04-22 01:38 /out1/part-r-00000
[cloudera@quickstart ~]$ hdfs dfs -ls /out1/part-r-00000
-rw-r--r--   1 cloudera supergroup       96 2025-04-22 01:38 /out1/part-r-00000
[cloudera@quickstart ~]$ hdfs dfs -cat /out1/part-r-00000
bharat    1
boss    1
devansh    1
hp    1
jay    2
ke    1
```

```
mata    1
om      1
papadiwal    1
patil   1
ram     1
rohit   1
shree   1
```

ASSIGNMENT 2

(MAPREDUCER)

REFERENCES :-
https://mapreduceprogram.blogspot.com/2024/02/mapreduce-program-in-hadoop-by-using.html

```java
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 MapReduce {

  public static class IPMapper
        extends Mapper<Object, Text, Text, IntWritable>{

        private final static IntWritable one = new IntWritable(1);
        private Text ip = new Text();

        public void map(Object key, Text value, Context context
                ) throws IOException, InterruptedException {
        // Assuming the IP address is the first token in each line
        StringTokenizer itr = new StringTokenizer(value.toString());
        if (itr.hasMoreTokens()) {
        ip.set(itr.nextToken());
        context.write(ip, one);
```

```java
        }
        }
    }

    public static class IntSumReducer
            extends Reducer<Text,IntWritable,Text,IntWritable> {
        private IntWritable result = new IntWritable();

        public void reduce(Text key, Iterable<IntWritable> values,
                Context context
                ) throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
        sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
        }
    }

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "IP address count");
        job.setJarByClass(MapReduce.class);
        job.setMapperClass(IPMapper.class);
        job.setCombinerClass(IntSumReducer.class);
        job.setReducerClass(IntSumReducer.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);
    }
}
```