



EXPERIMENT NO. 4

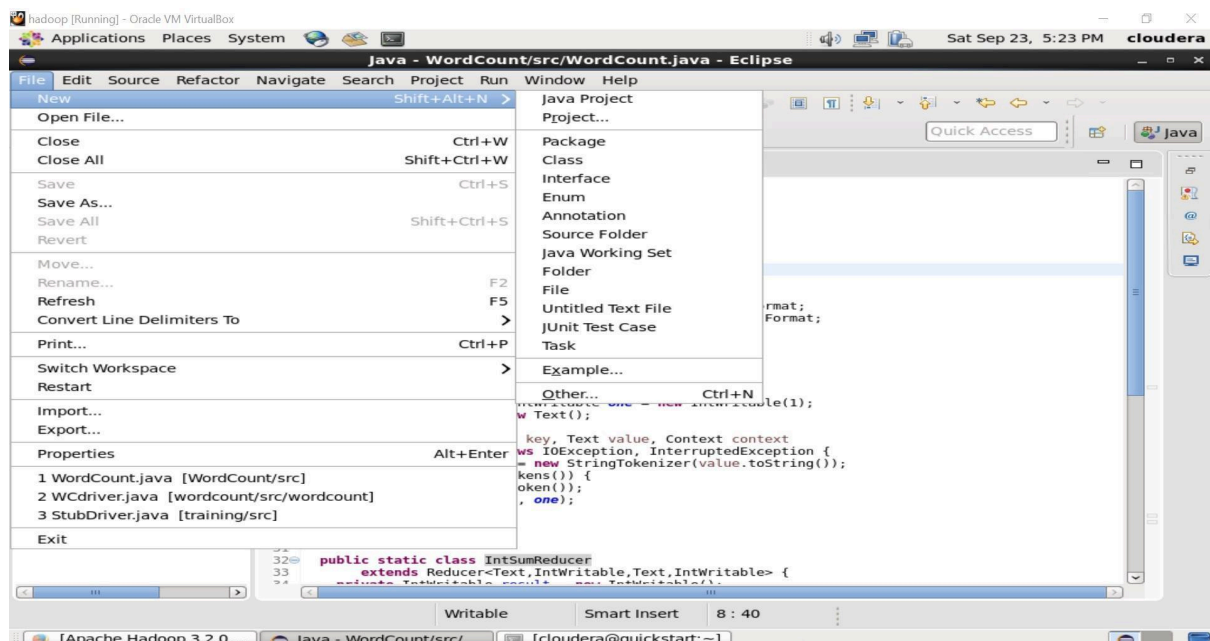
Aim: Execution of MapReduce program for sorting of numbers and counting word occurrences in a text file.

Theory:

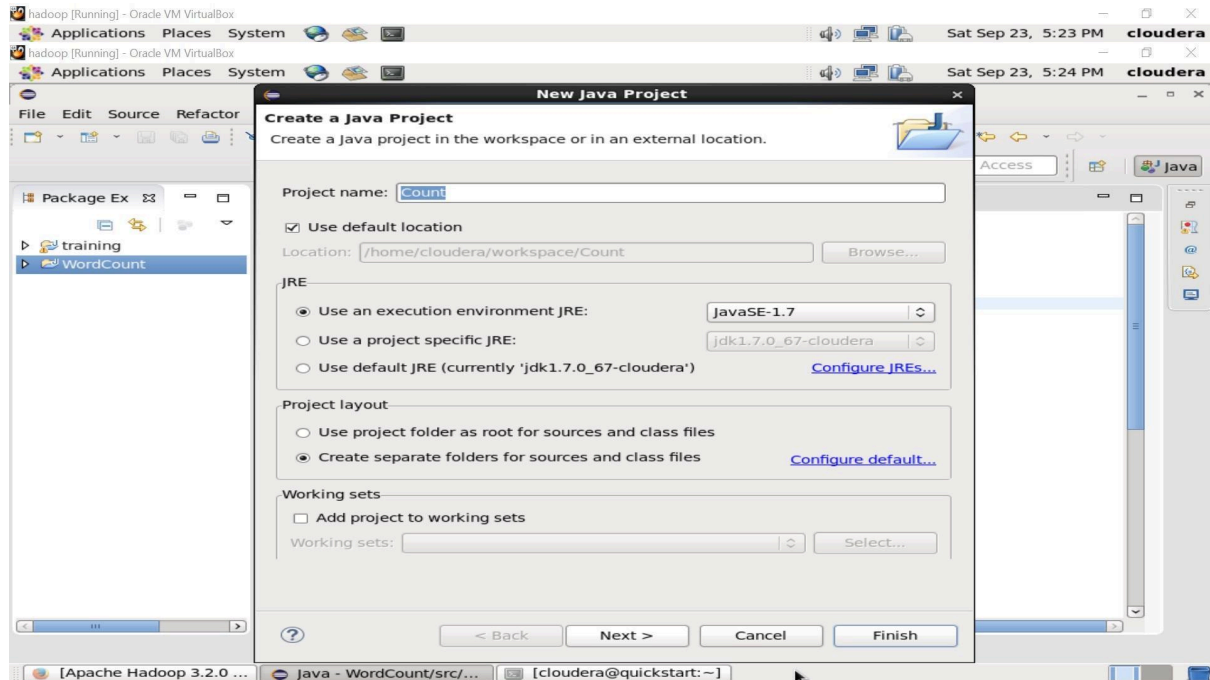
1. Explain the Use of Every Hadoop Ecosystem Component.

Practical

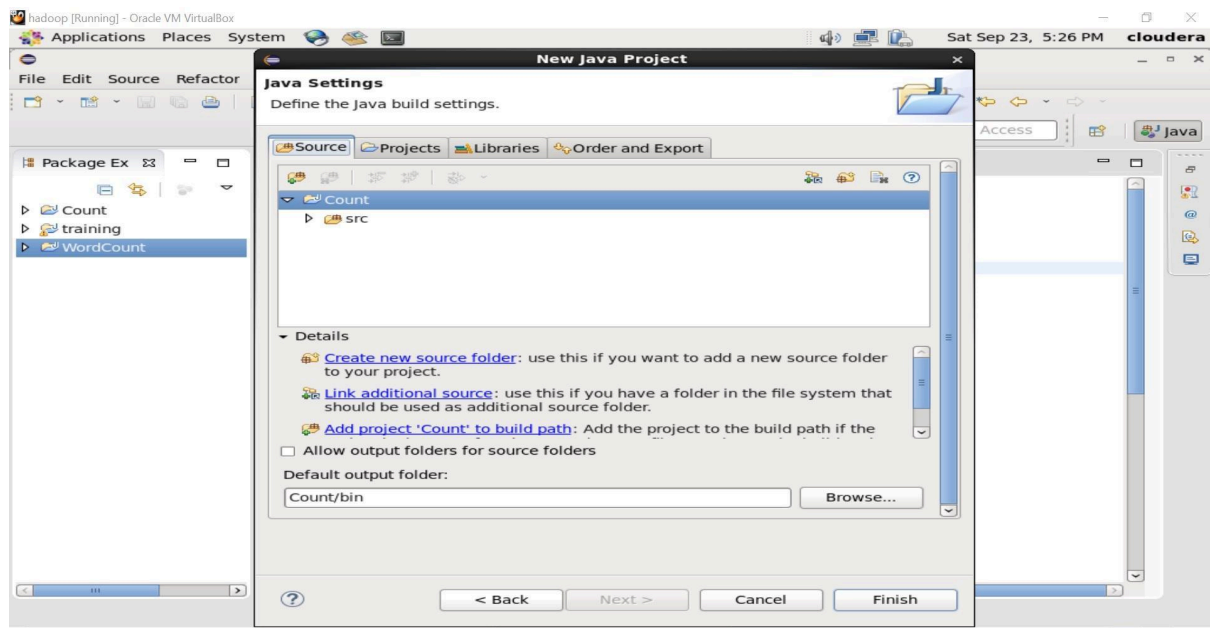
1. In cloudera → open eclipse → File → java Project



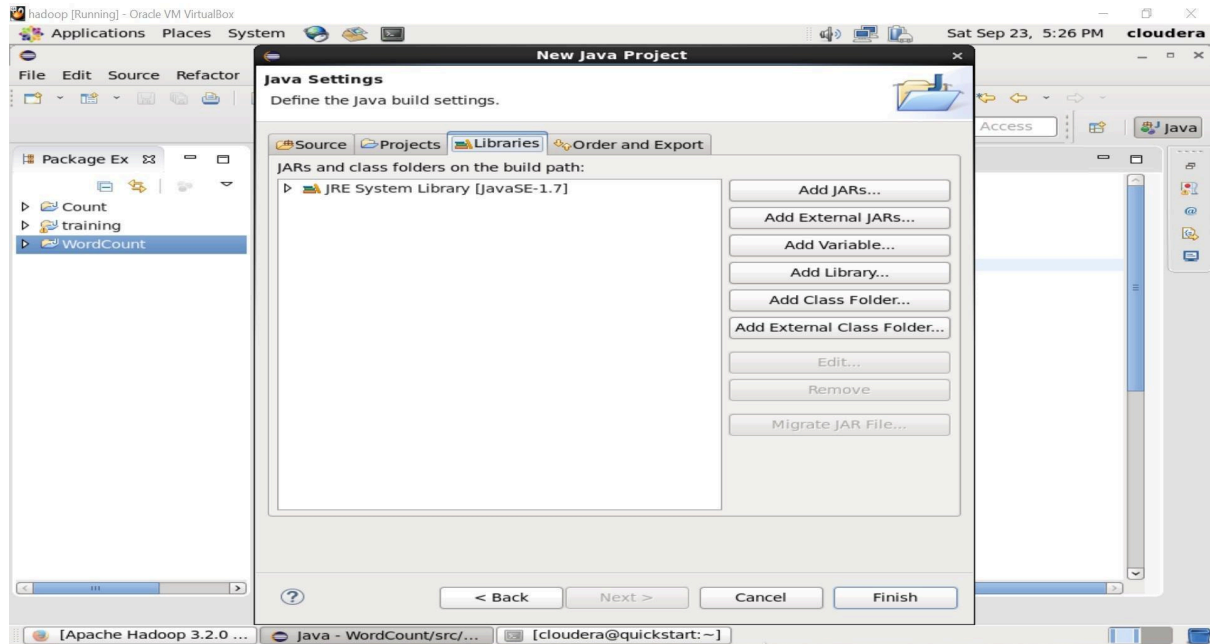
2. Save as file name "Count". Don't click on the finish button but continue with the next button.



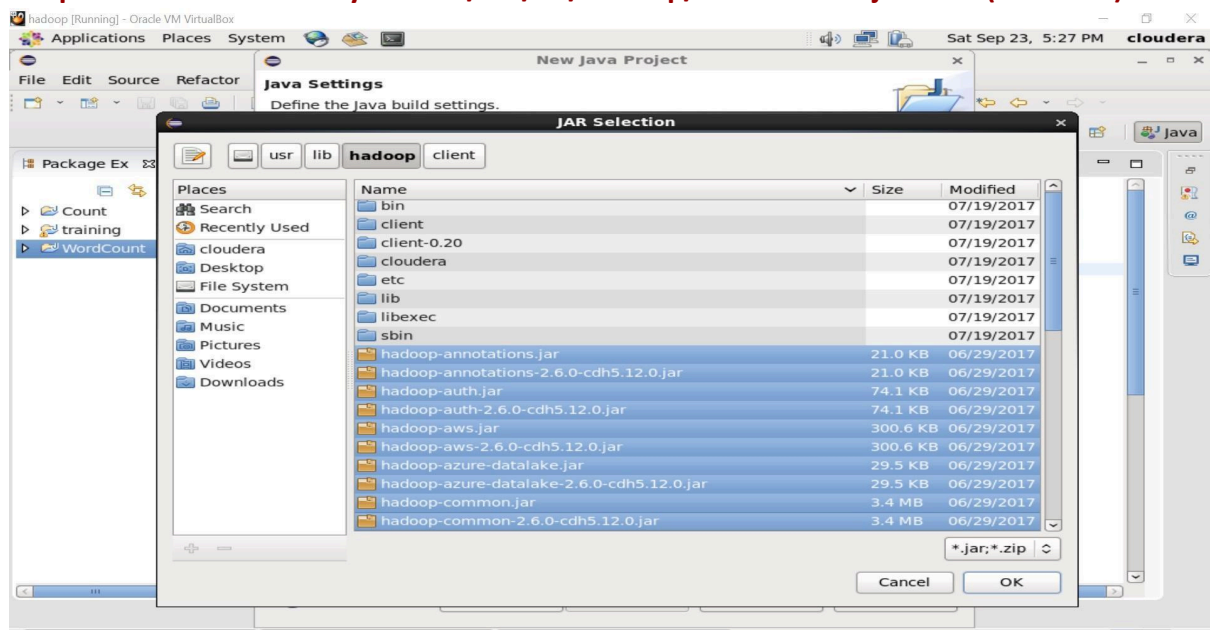
3. click on the Libraries tab.



4. Click on Add External JARS files.



5. Open the file from Filesystem → /usr/lib/hadoop/select the all jar files. (shfit+ctrl)

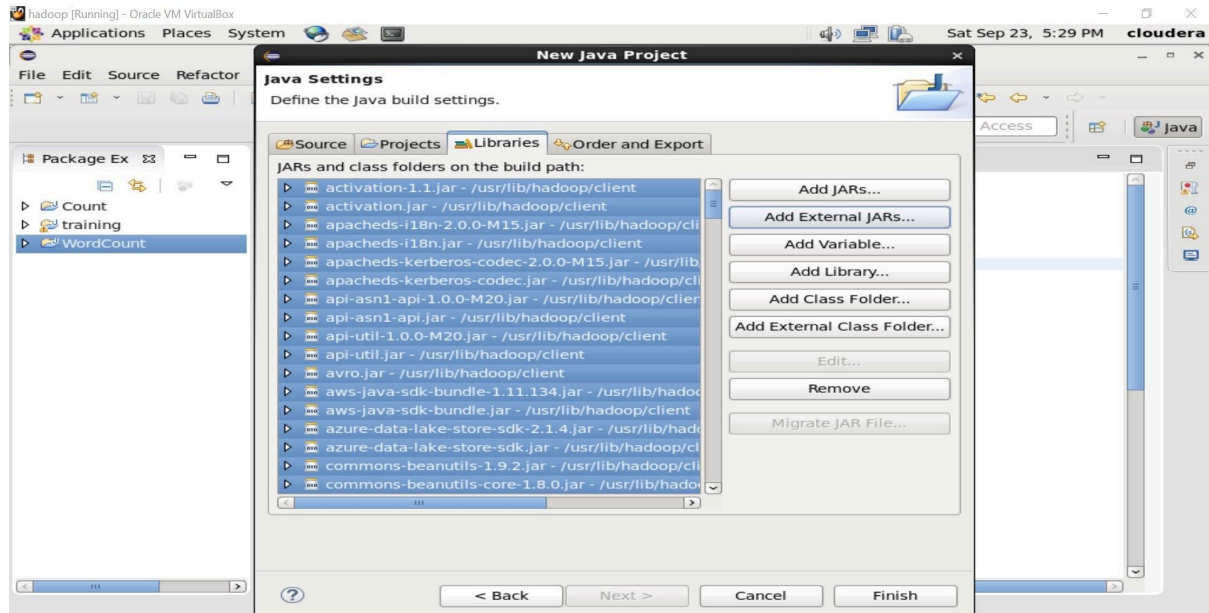




Option 5 is not for Cloudera_1.5 so directly jump on option 6.

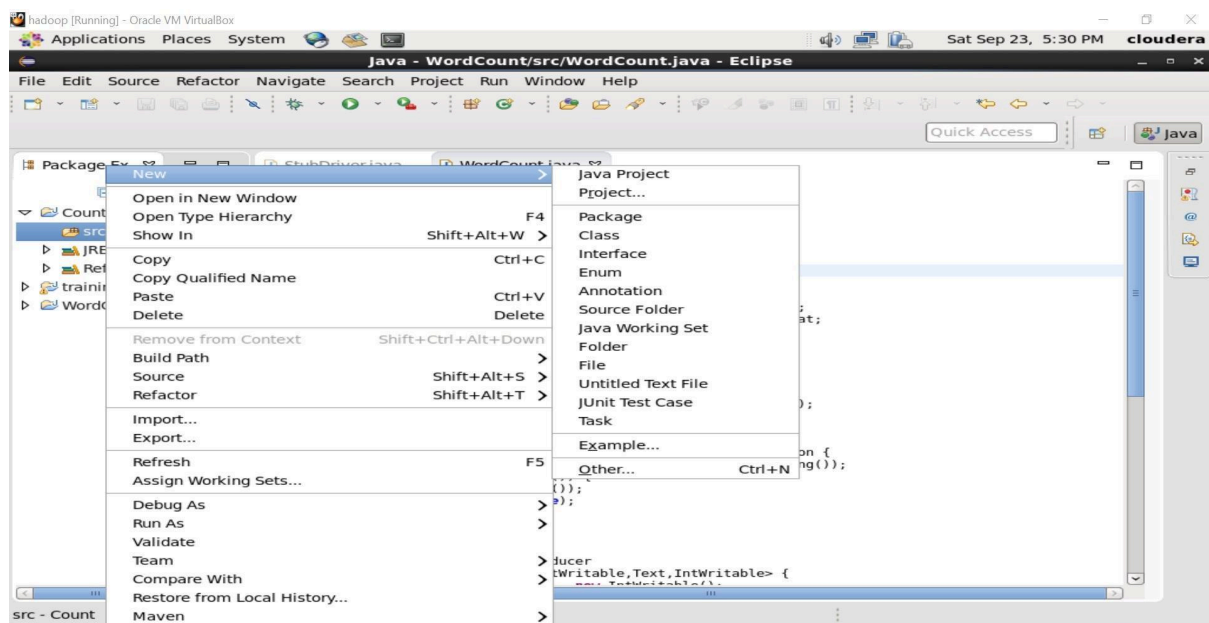
5. Again click on Add external JARS tab.

Open the file from Filesystem→ /usr/lib/hadoop/client/select all jar files.



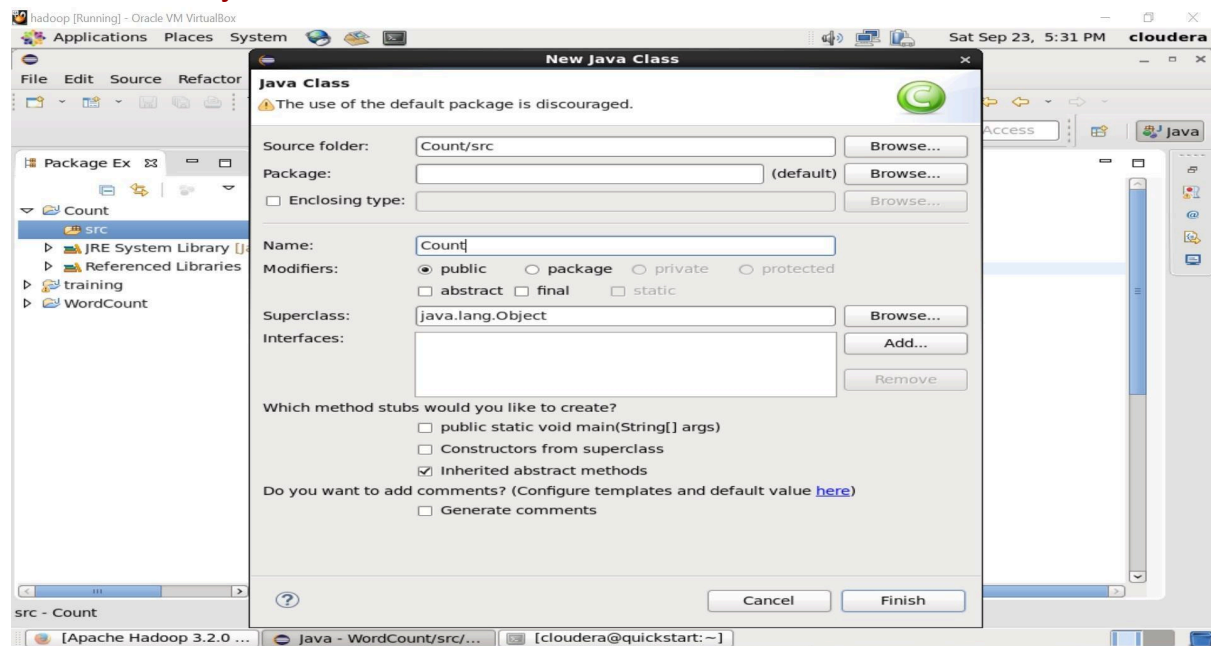
Click the finish button.

6. Right-click on the Src file→NEW→ Class.





7. Save as Count java class name



It will open the count.java file

8. Write MapReduce Program for word count in the java program.

```
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 shortcount2025 {

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

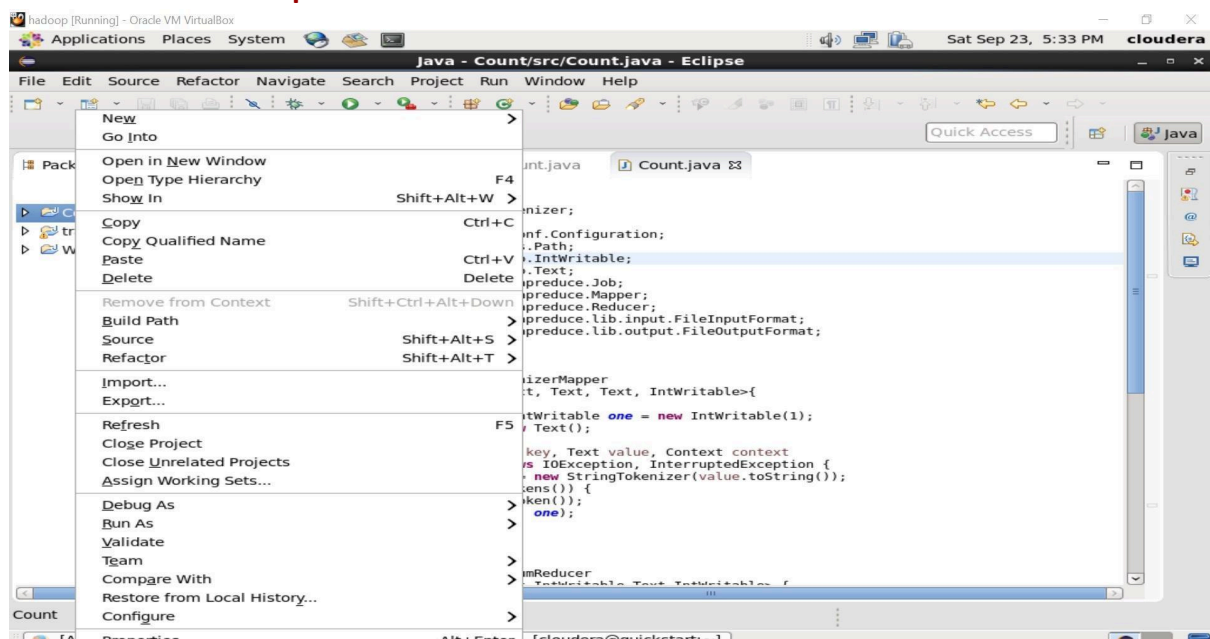
public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = new Job(conf, "short count 2025");
    //Job job = Job.getInstance(conf, "shortcount2025");
    job.setJarByClass(shortcount2025.class);
    job.setMapperClass(TokenizerMapper.class);
    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);
}
}
```

9.Save the file. (ctrl+S)

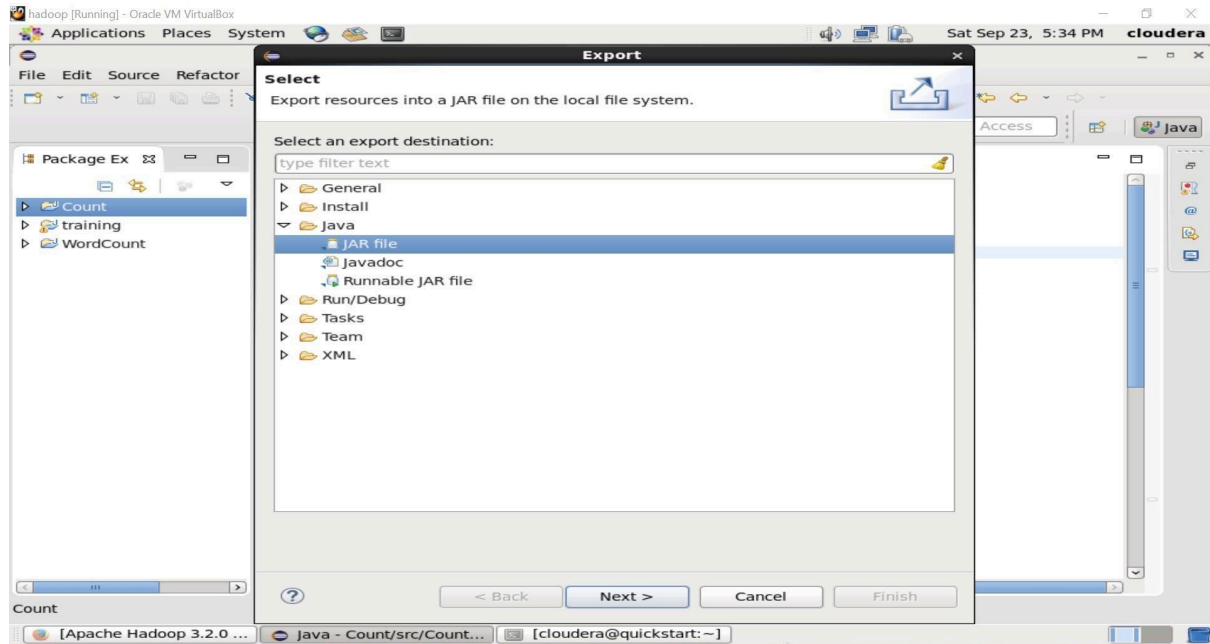


```
1 import java.io.IOException;
2 import java.util.StringTokenizer;
3
4 import org.apache.hadoop.conf.Configuration;
5 import org.apache.hadoop.fs.Path;
6 import org.apache.hadoop.io.IntWritable;
7 import org.apache.hadoop.io.Text;
8 import org.apache.hadoop.mapreduce.Job;
9 import org.apache.hadoop.mapreduce.Mapper;
10 import org.apache.hadoop.mapreduce.Reducer;
11 import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
12 import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
13
14 public class Count {
15
16     public static class TokenizerMapper
17         extends Mapper<Object, Text, Text, IntWritable>{
18
19         private final static IntWritable one = new IntWritable(1);
20         private Text word = new Text();
21
22     public void map(Object key, Text value, Context context
23         ) throws IOException, InterruptedException {
24         StringTokenizer itr = new StringTokenizer(value.toString());
25         while (itr.hasMoreTokens()) {
26             word.set(itr.nextToken());
27             context.write(word, one);
28         }
29     }
30 }
31
32 public static class IntSumReducer
33     extends Reducer<Text, IntWritable, Text, IntWritable> {
34
35     private IntWritable sum = new IntWritable(0);
36
37     public void reduce(Text key, Iterable<IntWritable> values, Context context
38         ) throws IOException, InterruptedException {
39         for (IntWritable val : values) {
40             sum.add(val.get());
41         }
42         context.write(key, sum);
43     }
44 }
```

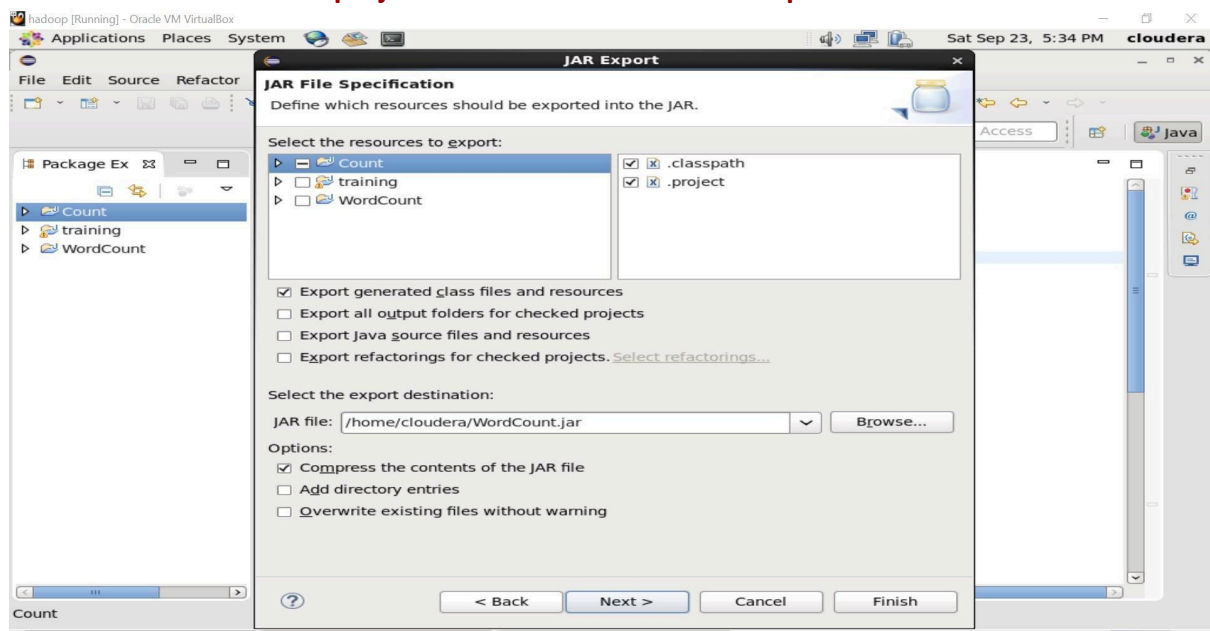
10. Click on File → Export.



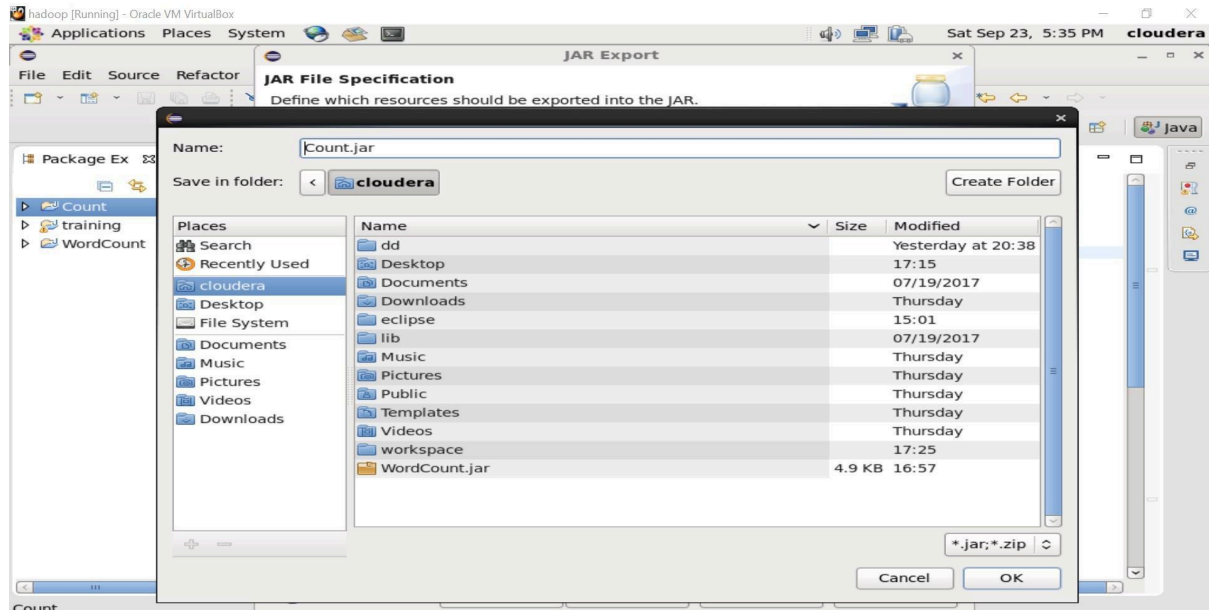
11. Export resources into Java→ a JAR file on the local file system.



12. Click on NEXT. Select project as “Count”. Next Select Export Destination.



13. Save as Count.jar file in cloudera directory.(home/user/cloudera)



14. check the Cloudera Directory Count.jar available or not.

14. Go to Terminal type command below

pwd

15.Create one text file for mapreduce count program

nano cn.txt

Enter some word like

Sakec

Sakec

Vesp

Vesp

Vesit

Djsce

Djsce

Ctrl+x →type "yes" → Enter(file save)

16. shift cn.txt on Hadoop. Before that create one directory aids

hadoop dfs -mkdir /aids

17.shift the cn.txt on this directory on hadoop.

hadoop dfs -put /home/usr/cloudera/cn.txt /aids

18.hadoop jar /home/cloudera/Count.jar Count /aids/cn.txt /out2(initially take 1 min.)



```

hadoop [Running] - Oracle VM VirtualBox
Applications Places System Sat Sep 23, 5:38 PM cloudera
cloudera@quickstart:~
File Edit View Search Terminal Help
cloudera-manager d.txt Pictures
cm_api.py eclipse Public
cn.txt enterprise-deployment.json Templates
Count.jar express-deployment.json Videos
dd kerberos WordCount.jar
Desktop lib workspace
Documents Music
[cloudera@quickstart ~]$ hadoop jar /home/cloudera/Count.jar Count /aids/cn.t
xt /out2
23/09/23 17:38:27 INFO client.RMPProxy: Connecting to ResourceManager at /0.0.
0.0:8032
23/09/23 17:38:28 WARN mapreduce.JobResourceUploader: Hadoop command-line opt
ion parsing not performed. Implement the Tool interface and execute your appl
ication with ToolRunner to remedy this.
23/09/23 17:38:29 INFO input.FileInputFormat: Total input paths to process :
1
23/09/23 17:38:29 INFO mapreduce.JobSubmitter: number of splits:1
23/09/23 17:38:30 INFO mapreduce.JobSubmitter: Submitting tokens for job: job
_1695505871221_0002
23/09/23 17:38:30 INFO impl.YarnClientImpl: Submitted application application
_1695505871221_0002
23/09/23 17:38:30 INFO mapreduce.Job: The url to track the job: http://quicks
tart.cloudera:8088/proxy/application_1695505871221_0002/
23/09/23 17:38:30 INFO mapreduce.Job: Running job: job_1695505871221_0002

```

19.Checkout out2 file being created.

hadoop dfs -ls /

19.hadoop dfs -ls /aids/out2

```

hadoop [Running] - Oracle VM VirtualBox
Applications Places System Sat Sep 23, 5:46 PM cloudera
cloudera@quickstart:~
File Edit View Search Terminal Help
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=61
File Output Format Counters
Bytes Written=49
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 10 items
drwxr-xr-x - cloudera supergroup 0 2023-09-23 17:06 /aids
drwxrwxrwx - hdfs supergroup 0 2017-07-19 05:34 /benchmarks
drwxr-xr-x - hbase supergroup 0 2023-09-23 15:01 /hbase
drwxr-xr-x - cloudera supergroup 0 2023-09-22 09:11 /ml
drwxr-xr-x - cloudera supergroup 0 2023-09-23 17:11 /out1
drwxr-xr-x - cloudera supergroup 0 2023-09-23 17:43 /out2
drwxr-xr-x - solr solr 0 2017-07-19 05:37 /solr
drwxrwxrwt - hdfs supergroup 0 2023-09-21 20:40 /tmp
drwxr-xr-x - hdfs supergroup 0 2017-07-19 05:36 /user
drwxr-xr-x - hdfs supergroup 0 2017-07-19 05:36 /var
[cloudera@quickstart ~]$ hdfs dfs -ls /out2
Found 2 items
-rw-r--r-- 1 cloudera supergroup 0 2023-09-23 17:43 /out2/_SUCCESS
-rw-r--r-- 1 cloudera supergroup 49 2023-09-23 17:43 /out2/part-r-0
0000
[cloudera@quickstart ~]$ hdfs dfs -cat /out2/part-r-00000

```

20. Status of Out2 file

hadoop dfs -ls /out2

21. Finally run command

hadoop dfs -cat /out2/part-r-00000



```
hadoop [Running] - Oracle VM VirtualBox
Applications Places System Sat Sep 23, 5:46 PM cloudera

cloudera@quickstart:~$ hdfs dfs -ls /
Found 10 items
drwxr-xr-x - cloudera supergroup 0 2023-09-23 17:06 /aids
drwxrwxrwx - hdfs supergroup 0 2017-07-19 05:34 /benchmarks
drwxr-xr-x - hbase supergroup 0 2023-09-23 15:01 /hbase
drwxr-xr-x - cloudera supergroup 0 2023-09-22 09:11 /ml
drwxr-xr-x - cloudera supergroup 0 2023-09-23 17:11 /out1
drwxr-xr-x - cloudera supergroup 0 2023-09-23 17:43 /out2
drwxr-xr-x - solr solr 0 2017-07-19 05:37 /solr
drwxrwxrwt - hdfs supergroup 0 2023-09-21 20:40 /tmp
drwxr-xr-x - hdfs supergroup 0 2017-07-19 05:36 /user
drwxr-xr-x - hdfs supergroup 0 2017-07-19 05:36 /var
cloudera@quickstart ~]$ hdfs dfs -ls /out2
Found 2 items
-rw-r--r-- 1 cloudera supergroup 0 2023-09-23 17:43 /out2/_SUCCESS
-rw-r--r-- 1 cloudera supergroup 49 2023-09-23 17:43 /out2/part-r-00000
cloudera@quickstart ~]$ hdfs dfs -cat /out2/part-r-00000
dashrath 1
djsce 2
kale 1
sakec 2
vesit 2
vesp 2
cloudera@quickstart ~]$
```

23. Want see on Browser put url : localhost:50070

NameNode 'localhost.localdomain:8020'

Started:	Wed Sep 27 13:33:24 PDT 2023
Version:	0 202-cdh3u2_95a824e4005b2a94fe1c1f1ef9db4c672ba43cb
Compiled:	Thu Oct 13 21:51:41 PDT 2011 by root from Unknown
Upgrades:	There are no upgrades in progress.

Cluster Summary

128 files and directories, 402 blocks = 1230 total. Heap Size is 44.44 MB / 177.81 MB (24%)

Configured Capacity	17.33 GB
DFS Used	93.34 MB
Non DFS Used	5.73 GB
DFS Remaining	11.52 GB
DFS Used%	0.53 %
DFS Remaining%	66.43 %
Live Nodes	1
Dead Nodes	0
Decommissioning Nodes	0
Number of Under-Replicated Blocks	114

NameNode Storage:

Storage Directory	Type	State
/var/lib/hadoop-0.20/cache/hadoop/dfs/name	IMAGE_AND_EDITS	Active

3.0.0.0's Distribution Including Apache Hadoop, 2023.

Conclusion: hence we study how to run word count programs using MapReduce module on Hadoop.

References:

- <https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html>
- <https://youtu.be/Wb5p8S5jZCc?si=5J8rmuyMzWMowzbX>
- https://www.youtube.com/watch?v=uH5y6nTo_04&t=2s