PROGRAM:

1. WordCount.java –

```
package WC;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.Job;
public class WordCount{
        public static void main(String[] args) throws Exception{
                if (args.length != 2){
                        System.out.printf("Usage: WordCount <input dir> <output dir>\n");
                        System.exit(-1); }
                Job job = new Job();
                job.setJarByClass(WordCount.class);
                job.setJobName("Word Count");
                FileInputFormat.setInputPaths(job, new Path(args[0]));
                FileOutputFormat.setOutputPath(job, new Path(args[1]));
                job.setMapperClass(WordCountMapper.class);
                job.setReducerClass(WordCountReducer.class);
                job.setMapOutputKeyClass(Text.class);
                job.setMapOutputValueClass(IntWritable.class);
                job.setOutputKeyClass(Text.class);
                job.setOutputValueClass(IntWritable.class);
                boolean success = job.waitForCompletion(true);
                System.exit(success ? 0 : 1);       }}
```

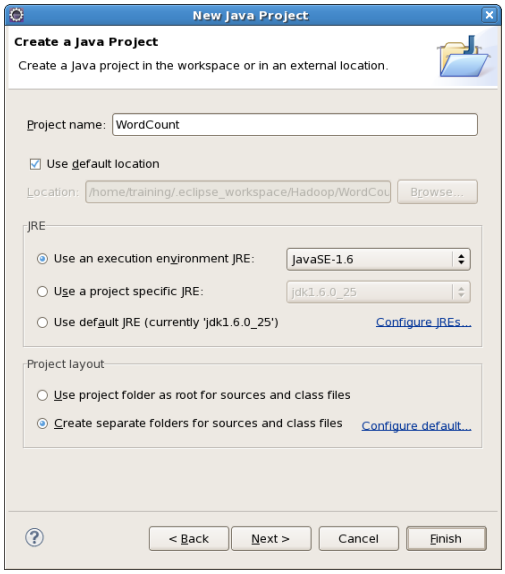
2. WordCountMapper.java –

```
Package WC;
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.Mapper;
public class WordCountMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException{
                String line = value.toString();
                for (String word : line.split("\\W+")){
                        if (word.length() > 0)
                                context.write(new Text(word), new IntWritable(1));  }} }
```

3. WordCountReducer.java –

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class SumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
public void reduce(Text key, Iterable<IntWritable> values, Context context)  throws IOException,
InterruptedException{
                int wordCount = 0;
                for (IntWritable value : values)
                        wordCount += value.get();
                context.write(key, new IntWritable(wordCount));          }}
```

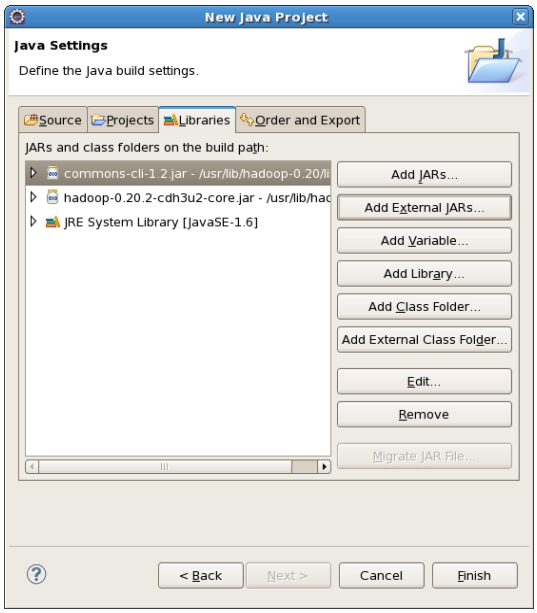
OUTPUT: Step 1: In Cloudera_training_VM_1.6, open Eclipse. Files > New Project > Give project name as “WordCount”.



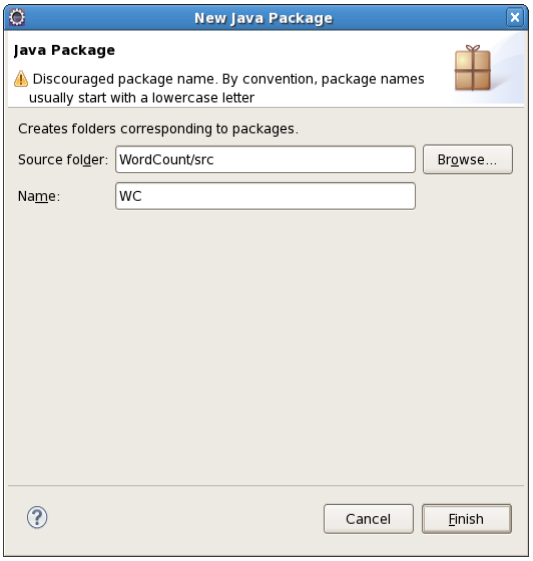
Click Next > Libraries >

1. hadoop folder > add hadoop-0.02.2-cdh3u-core.jar
2. lib folder > add commons-cli-1.2.jar

Click OK.

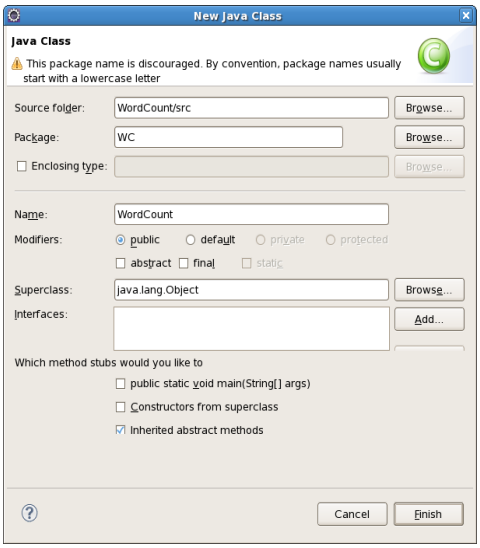


Step 2: Files > New Package > Give package name as “WC”.

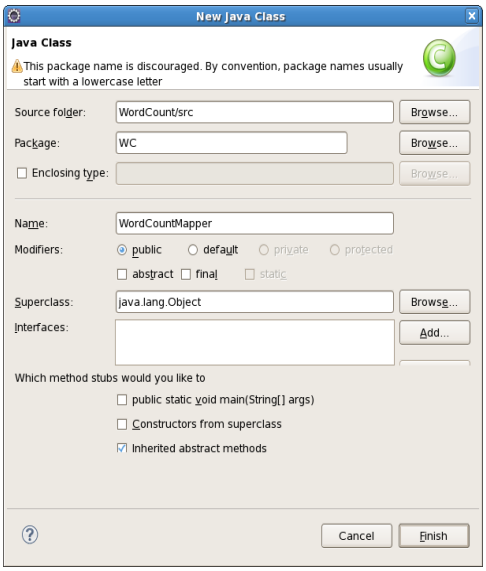


Step 3: Create three java class files with appropriate code –

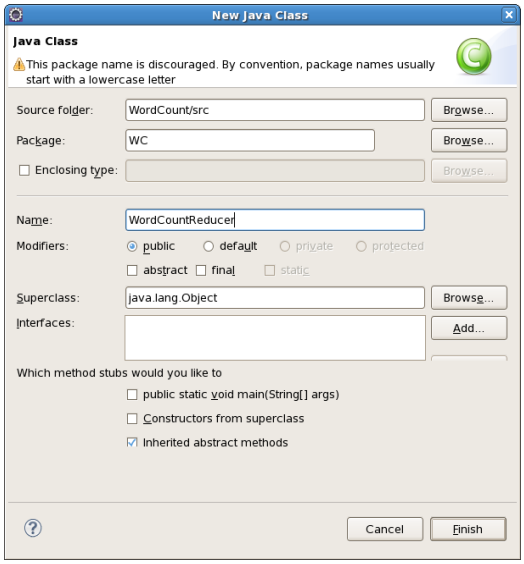
1. WordCount.java



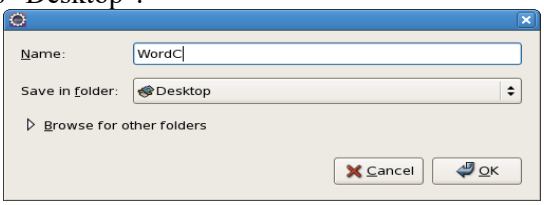
1. WordCountMapper.java



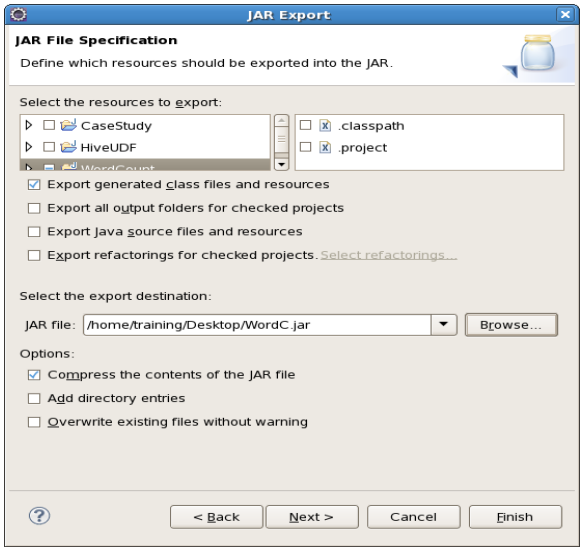
1. WordCountReducer.java



Step 4: Right click on WC package > Export. In Java folder, select “JAR file”. Give the name as “WordC” > save in folder as “Desktop”.



Click “OK”.



Click “Finish”.

[training@localhost Desktop]$ cd

[training@localhost ~]$ vi inputabc.txt

[training@localhost ~]$ hadoop fs -copyFromLocal /home/training/inputabc.txt /user/training

[training@localhost ~]$ cd Desktop/

[training@localhost Desktop]$ hadoop jar WordC.jar WC.WordCount /user/training/inputWC.txt

Usage: WordCount <input dir> <output dir>

[training@localhost Desktop]$ hadoop jar WordC.jar WC.WordCount /user/training/inputabc.txt

Usage: WordCount <input dir> <output dir>

[training@localhost Desktop]$ hadoop jar WordC.jar WC.WordCount /user/training/inputabc.txt outputabc.txt

19/09/08 23:19:00 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the same.

19/09/08 23:19:00 INFO input.FileInputFormat: Total input paths to process : 1

19/09/08 23:19:00 WARN snappy.LoadSnappy: Snappy native library is available

19/09/08 23:19:00 INFO util.NativeCodeLoader: Loaded the native-hadoop library

19/09/08 23:19:00 INFO snappy.LoadSnappy: Snappy native library loaded

19/09/08 23:19:00 INFO mapred.JobClient: Running job: job_201909082224_0003

19/09/08 23:19:01 INFO mapred.JobClient: map 0% reduce 0%

19/09/08 23:19:03 INFO mapred.JobClient: map 100% reduce 0%

19/09/08 23:19:11 INFO mapred.JobClient: map 100% reduce 100%

19/09/08 23:19:11 INFO mapred.JobClient: Job complete: job_201909082224_0003

19/09/08 23:19:11 INFO mapred.JobClient: Counters: 22

19/09/08 23:19:11 INFO mapred.JobClient: Job Counters

19/09/08 23:19:11 INFO mapred.JobClient: Launched reduce tasks=1

19/09/08 23:19:11 INFO mapred.JobClient: SLOTS_MILLIS_MAPS=1702

19/09/08 23:19:11 INFO mapred.JobClient: Total time spent by all reduces waiting after reserving slots (ms)=0

19/09/08 23:19:11 INFO mapred.JobClient: Total time spent by all maps waiting after reserving slots (ms)=0

19/09/08 23:19:11 INFO mapred.JobClient: Launched map tasks=1

[training@localhost Desktop]$ hadoop fs -ls /user/training/outputabc.txt

Found 3 items

-rw-r--r--   1 training supergroup   0 2019-09-08 23:19 /user/training/outputabc.txt/_SUCCESS

drwxr-xr-x   - training supergroup   0 2019-09-08 23:19 /user/training/outputabc.txt/_logs

-rw-r--r--   1 training supergroup   1324 2019-09-08 23:19 /user/training/outputabc.txt/part-r-00000

OUTPUT

[training@localhost Desktop]$ hadoop fs -cat /user/training/outputabc.txt/part-r-00000

1	1

10	1

11	1

12 1

13 1

14 1

15 1

16 1

17 1

18 1

19 1

2 11

20 1

21 1

22 1

23 1

24 1

25 1

26 1

27 1

28 1

29 1

3 1

30 1

4 1

5 1

6 1

7 1

8 1

9 1

BMP 10

Baseball 10

Block 10

BlockStyleTest 10

Boston 10

Bruins 9

Called 10

Defined 10

Detroit 10

Excel 9

First 30

Football 9

Fourth 10

Giants 9

Heading 10

Heading1 10

Here 29

Hockey 9

InlineStyle 10

Islanders 9

Jets 9

Lions 9

Mets 10

New 10

Normal 80

Patriots 9

Rangers 9

Red 19

Second 50

Sox 10

Style 10

Text 10

Third 20

This 199

Tigers 10

Wings 9

Yankees 10

```
York    10
a       90
aligned 10
an      9
and     20
block   10
bold    20
bulleted        50
called  10
centered        10
concludes       9
default 50
defined 10
embedded        9
final   10
in      10
inline  10
is      239
italic  20
item    130
left    10
list    110
main    10
more    20
nested  20
normal  30
[training@localhost Desktop]$
```