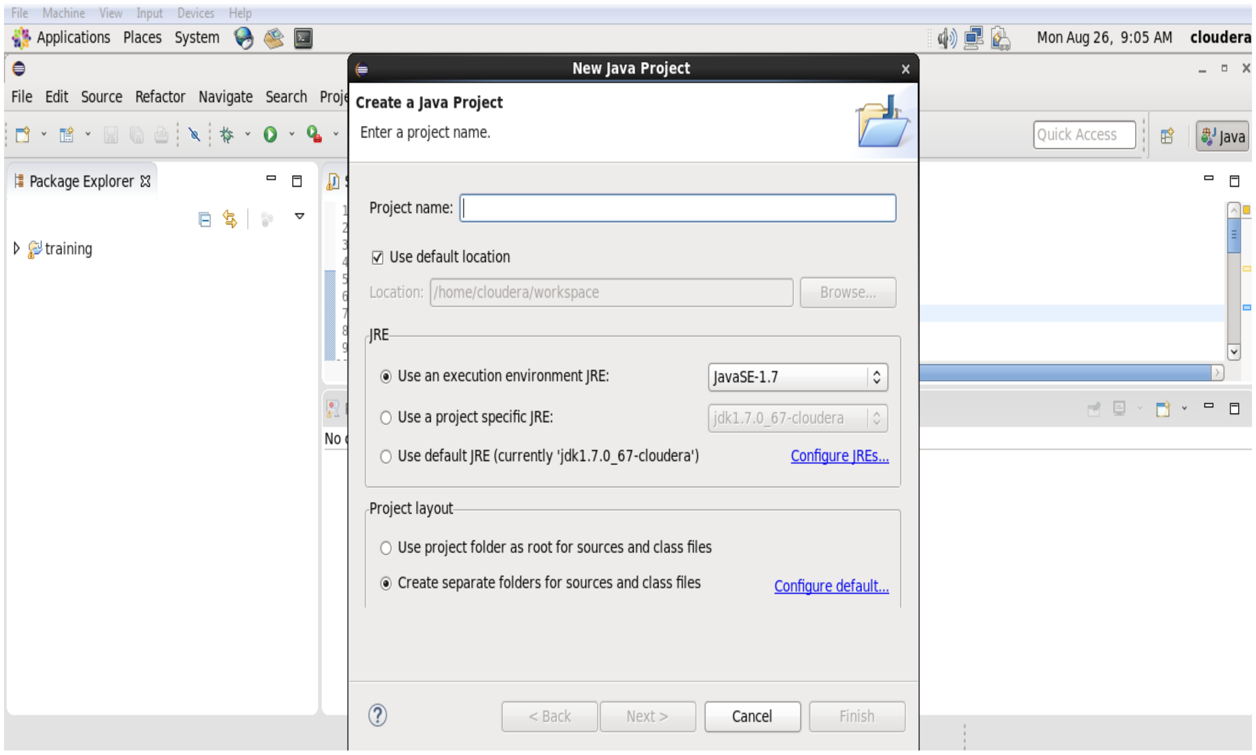
CSCE 5300 Introduction to Big Data and Data Science

ICE-2 Eclipse Project Guidelines

Eclipse WordCount Project:

Step by step instructions:

File > New > Java Project > Next.


File Machine View Input Devices Help
Applications Places System
Mon Aug 26, 9:05 AM cloudera
File Edit Source Refactor Navigate Search
Package Explorer
training
Quick Access
Java
< Back
Next >
Cancel
Finish

New Java Project

Create a Java Project

Enter a project name.

Project name:

[✓] Use default location

Location: /home/cloudera/workspace Browse...

JRE

[✓ Use an execution environment JRE: JavaSE-1.7]

Use a project specific JRE: jdk1.7.0_67-cloudera

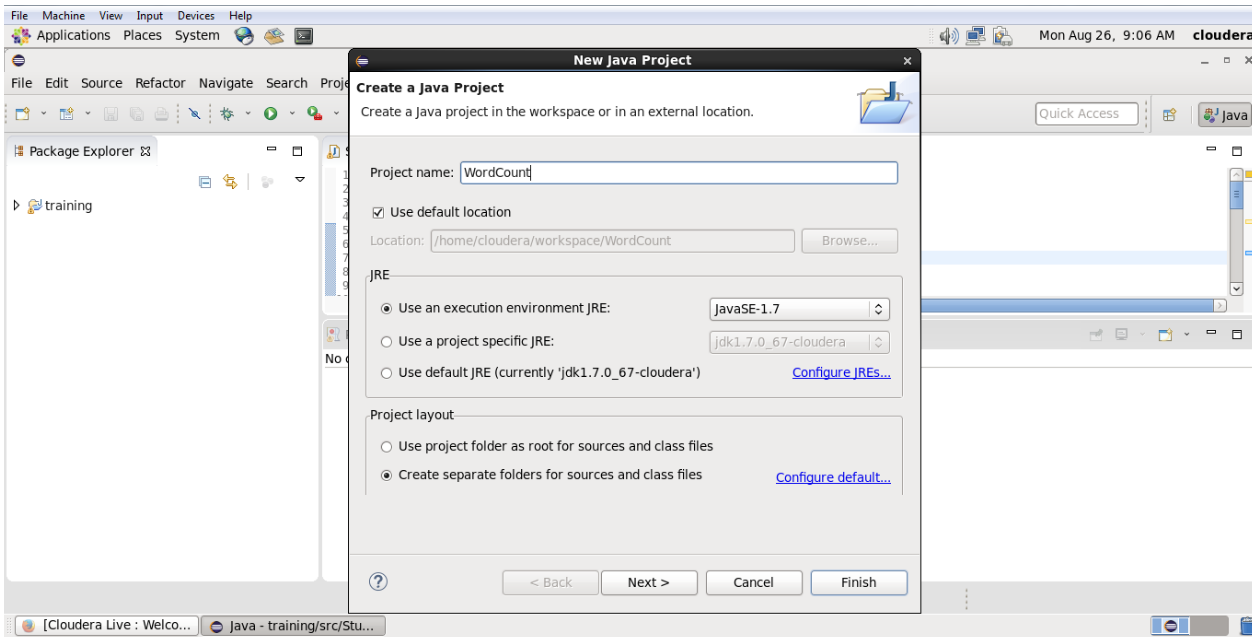
Use default JRE (currently 'jdk1.7.0_67-cloudera') Configure JREs...

Project layout

Use project folder as root for sources and class files

[✓ Create separate folders for sources and class files] Configure default...

"WordCount" as our project name and click "Finish":



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: WordCount

☑ Use default location

Location: /home/cloudera/workspace/WordCount

JRE

(●) Use an execution environment JRE: JavaSE-1.7

( ) Use a project specific JRE: jdk1.7.0_67-cloudera

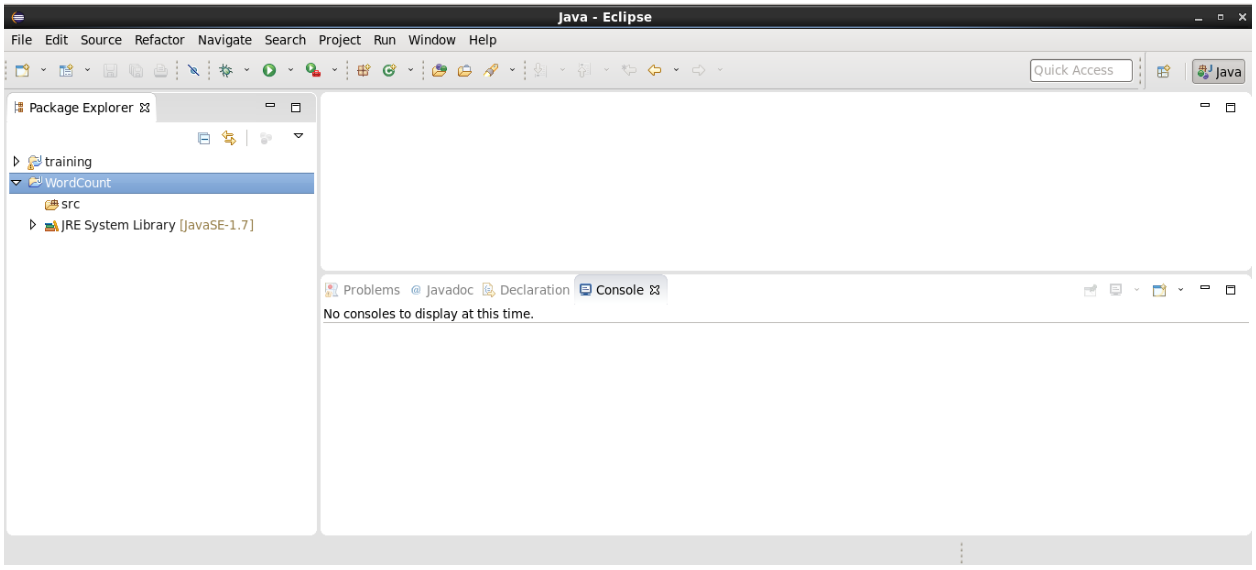
( ) Use default JRE (currently 'jdk1.7.0_67-cloudera') Configure JREs...

Project layout

( ) Use project folder as root for sources and class files

(●) Create separate folders for sources and class files Configure default...

< Back | Next > | Cancel | Finish



Java - Eclipse

Package Explorer

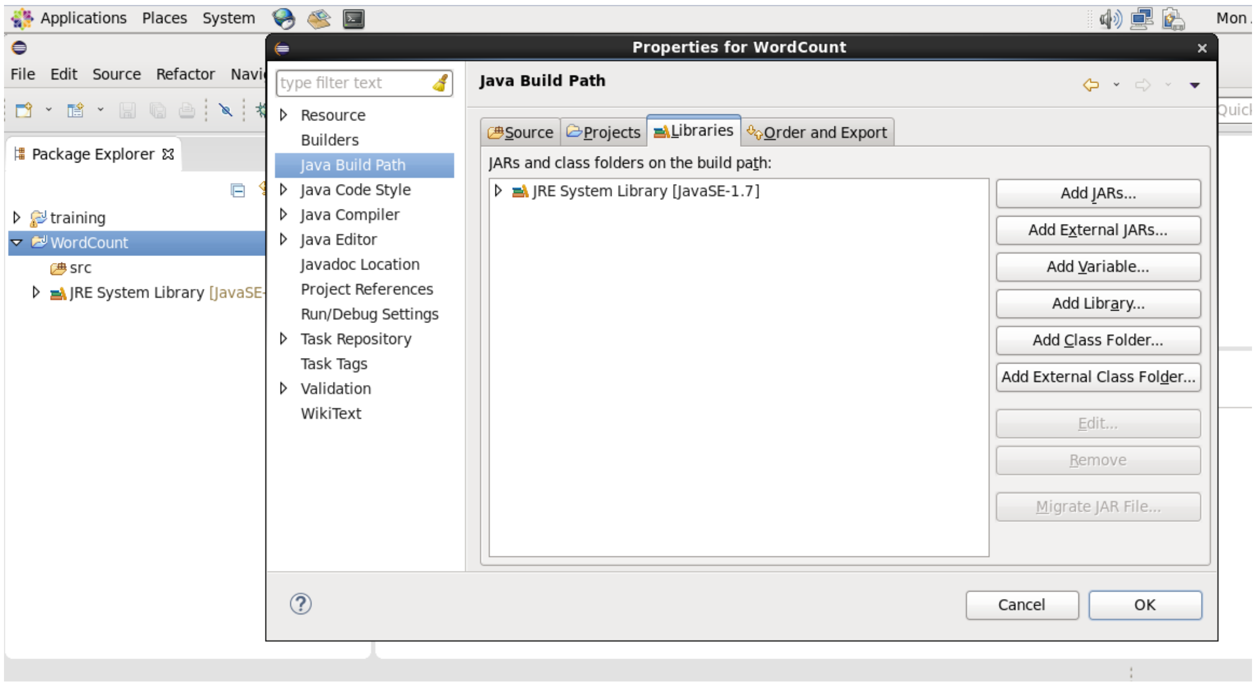
- training
- WordCount
  - src
  - JRE System Library [JavaSE-1.7]

Problems | Javadoc | Declaration | Console

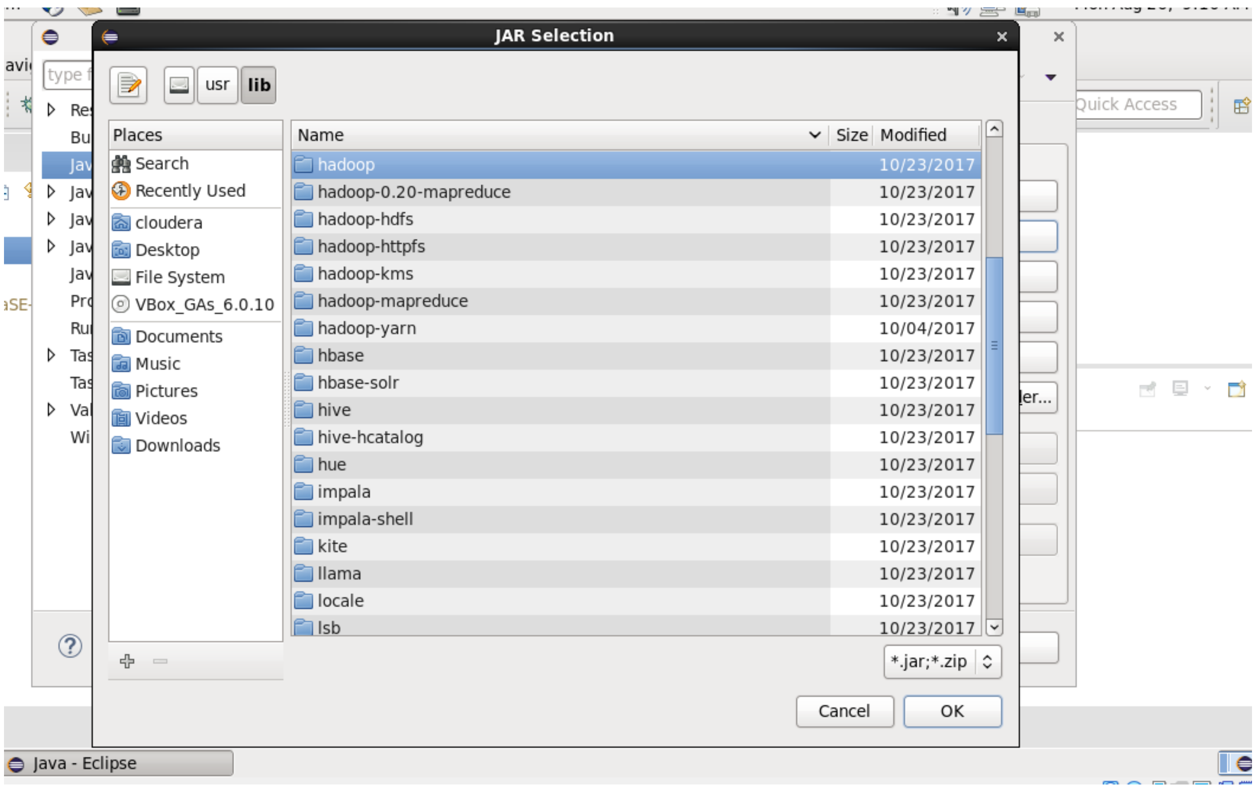
No consoles to display at this time.

Getting references to hadoop libraries

Right click on WordCount project and select "Properties":

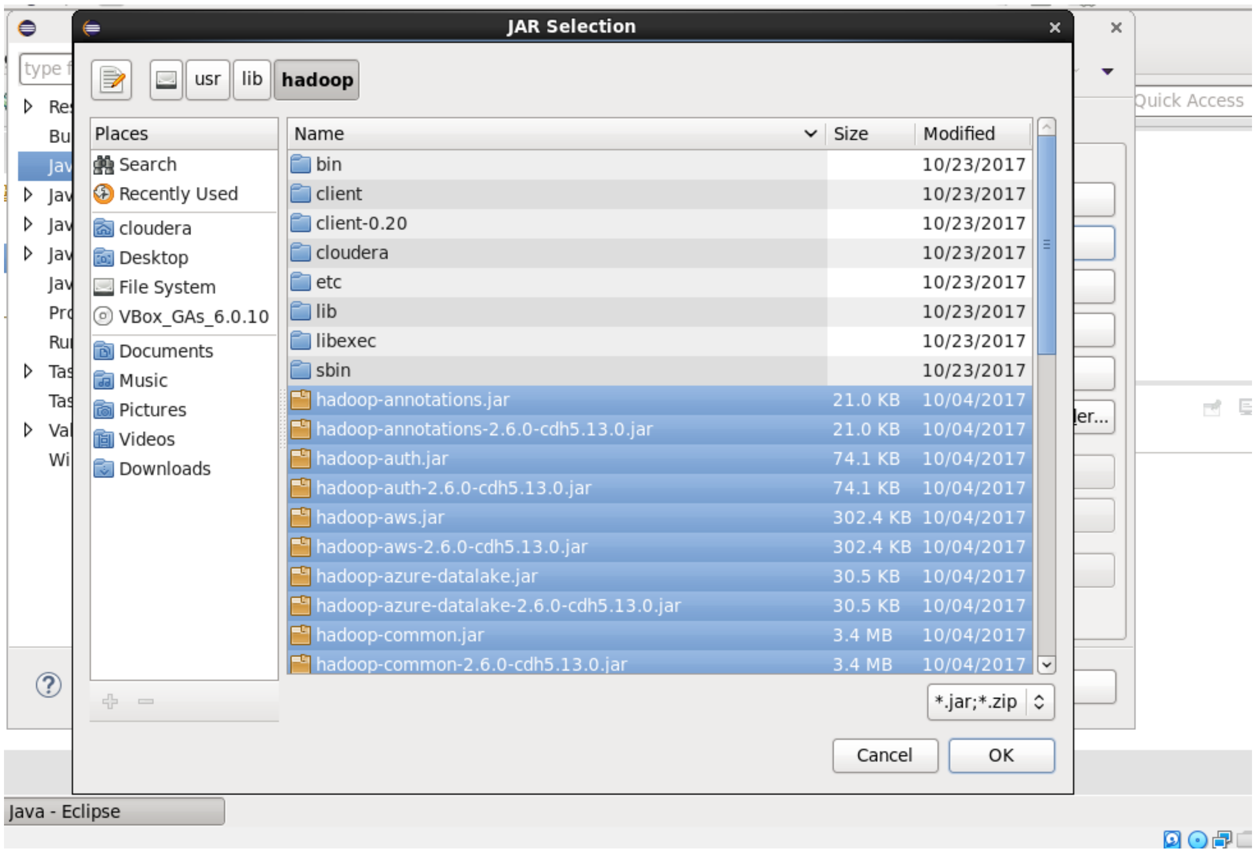


Hit "Add External JARs...", then, File System > usr > lib > hadoop :

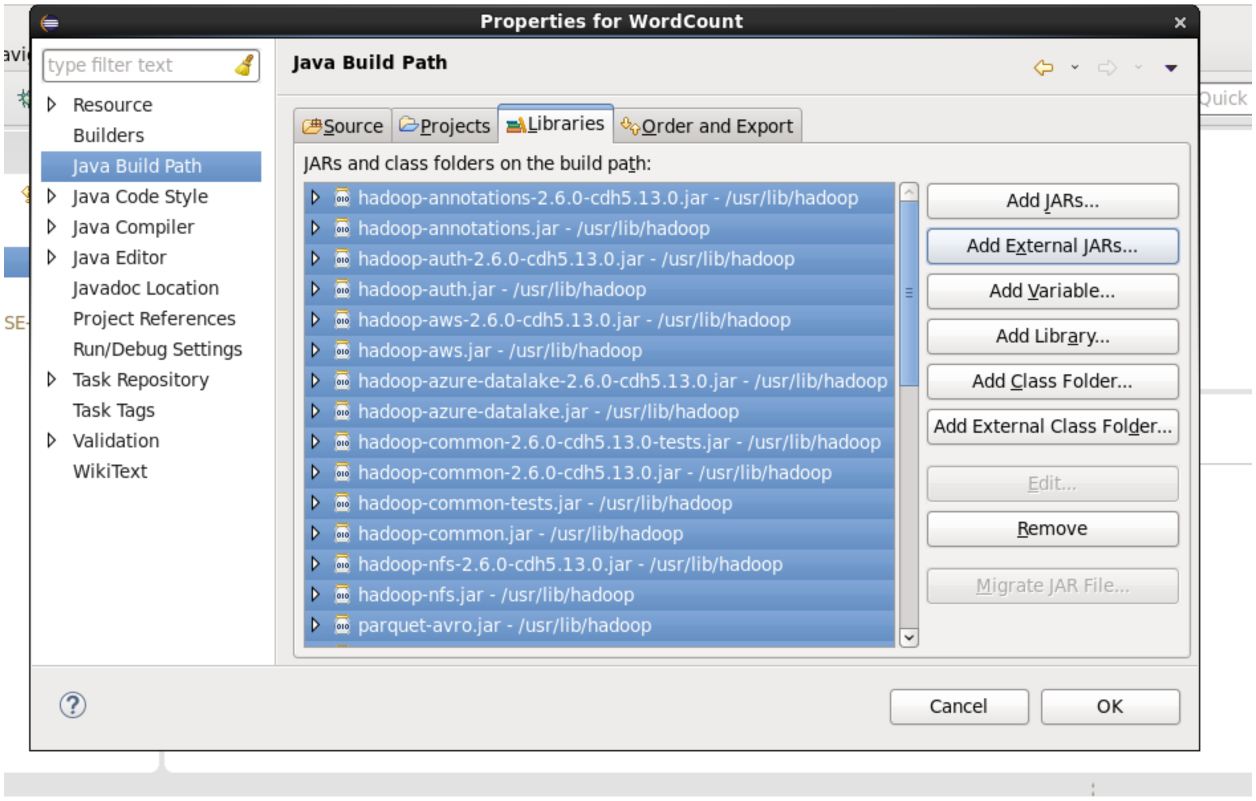


Name	Size	Modified
hadoop		10/23/2017
hadoop-0.20-mapreduce		10/23/2017
hadoop-hdfs		10/23/2017
hadoop-httpfs		10/23/2017
hadoop-kms		10/23/2017
hadoop-mapreduce		10/23/2017
hadoop-yarn		10/04/2017
hbase		10/23/2017
hbase-solr		10/23/2017
hive		10/23/2017
hive-hcatalog		10/23/2017
hue		10/23/2017
impala		10/23/2017
impala-shell		10/23/2017
kite		10/23/2017
llama		10/23/2017
locale		10/23/2017
lsb		10/23/2017

We may want to select all jars, and click OK:

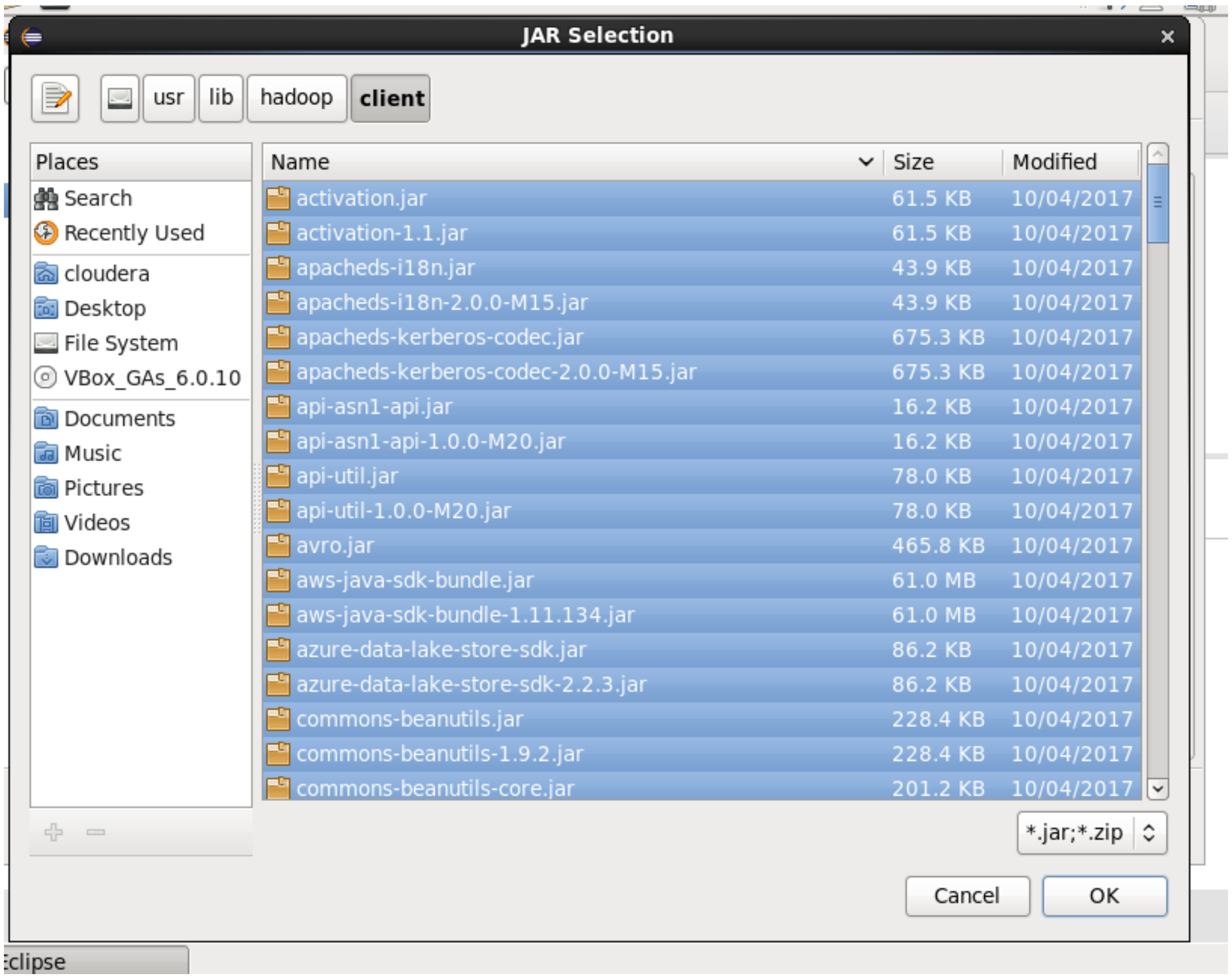


Name	Size	Modified
bin		10/23/2017
client		10/23/2017
client-0.20		10/23/2017
cloudera		10/23/2017
etc		10/23/2017
lib		10/23/2017
libexec		10/23/2017
sbin		10/23/2017
hadoop-annotations.jar	21.0 KB	10/04/2017
hadoop-annotations-2.6.0-cdh5.13.0.jar	21.0 KB	10/04/2017
hadoop-auth.jar	74.1 KB	10/04/2017
hadoop-auth-2.6.0-cdh5.13.0.jar	74.1 KB	10/04/2017
hadoop-aws.jar	302.4 KB	10/04/2017
hadoop-aws-2.6.0-cdh5.13.0.jar	302.4 KB	10/04/2017
hadoop-azure-datalake.jar	30.5 KB	10/04/2017
hadoop-azure-datalake-2.6.0-cdh5.13.0.jar	30.5 KB	10/04/2017
hadoop-common.jar	3.4 MB	10/04/2017
hadoop-common-2.6.0-cdh5.13.0.jar	3.4 MB	10/04/2017



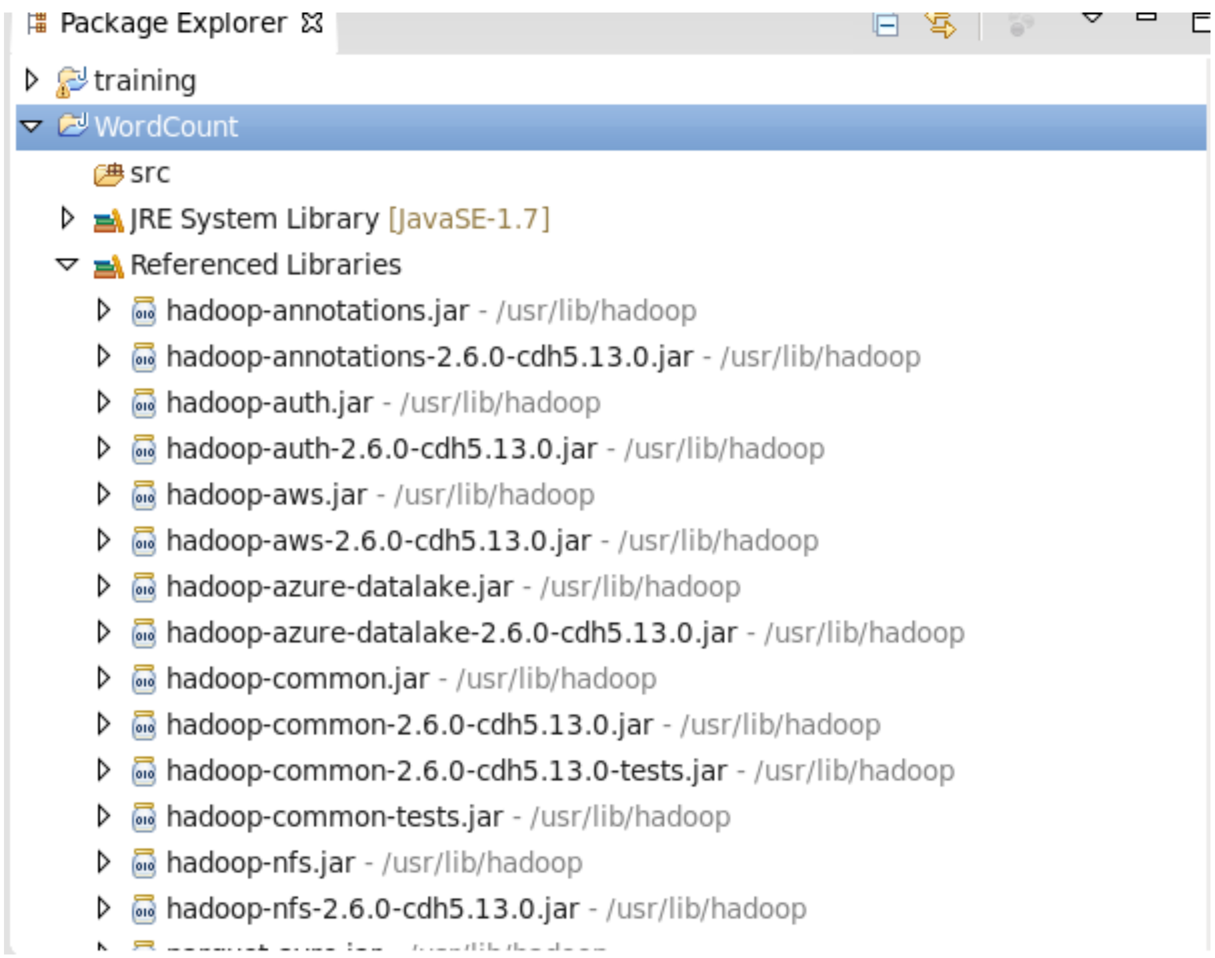
```
hadoop-annotations-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop
hadoop-annotations.jar - /usr/lib/hadoop
hadoop-auth-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop
hadoop-auth.jar - /usr/lib/hadoop
hadoop-aws-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop
hadoop-aws.jar - /usr/lib/hadoop
hadoop-azure-datalake-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop
hadoop-azure-datalake.jar - /usr/lib/hadoop
hadoop-common-2.6.0-cdh5.13.0-tests.jar - /usr/lib/hadoop
hadoop-common-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop
hadoop-common-tests.jar - /usr/lib/hadoop
hadoop-common.jar - /usr/lib/hadoop
hadoop-nfs-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop
hadoop-nfs.jar - /usr/lib/hadoop
parquet-avro.jar - /usr/lib/hadoop
```

We need to add more external libs. Go to "Add External JARs..." again, then grab all libs in "client": Then, hit "OK"


JAR Selection

usr lib hadoop **client**

Name	Size	Modified
activation.jar	61.5 KB	10/04/2017
activation-1.1.jar	61.5 KB	10/04/2017
apacheds-i18n.jar	43.9 KB	10/04/2017
apacheds-i18n-2.0.0-M15.jar	43.9 KB	10/04/2017
apacheds-kerberos-codec.jar	675.3 KB	10/04/2017
apacheds-kerberos-codec-2.0.0-M15.jar	675.3 KB	10/04/2017
api-asn1-api.jar	16.2 KB	10/04/2017
api-asn1-api-1.0.0-M20.jar	16.2 KB	10/04/2017
api-util.jar	78.0 KB	10/04/2017
api-util-1.0.0-M20.jar	78.0 KB	10/04/2017
avro.jar	465.8 KB	10/04/2017
aws-java-sdk-bundle.jar	61.0 MB	10/04/2017
aws-java-sdk-bundle-1.11.134.jar	61.0 MB	10/04/2017
azure-data-lake-store-sdk.jar	86.2 KB	10/04/2017
azure-data-lake-store-sdk-2.2.3.jar	86.2 KB	10/04/2017
commons-beanutils.jar	228.4 KB	10/04/2017
commons-beanutils-1.9.2.jar	228.4 KB	10/04/2017
commons-beanutils-core.jar	201.2 KB	10/04/2017


Package Explorer

training

WordCount

src

JRE System Library [JavaSE-1.7]

Referenced Libraries

hadoop-annotations.jar - /usr/lib/hadoop

hadoop-annotations-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop

hadoop-auth.jar - /usr/lib/hadoop

hadoop-auth-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop

hadoop-aws.jar - /usr/lib/hadoop

hadoop-aws-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop

hadoop-azure-datalake.jar - /usr/lib/hadoop

hadoop-azure-datalake-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop

hadoop-common.jar - /usr/lib/hadoop

hadoop-common-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop

hadoop-common-2.6.0-cdh5.13.0-tests.jar - /usr/lib/hadoop

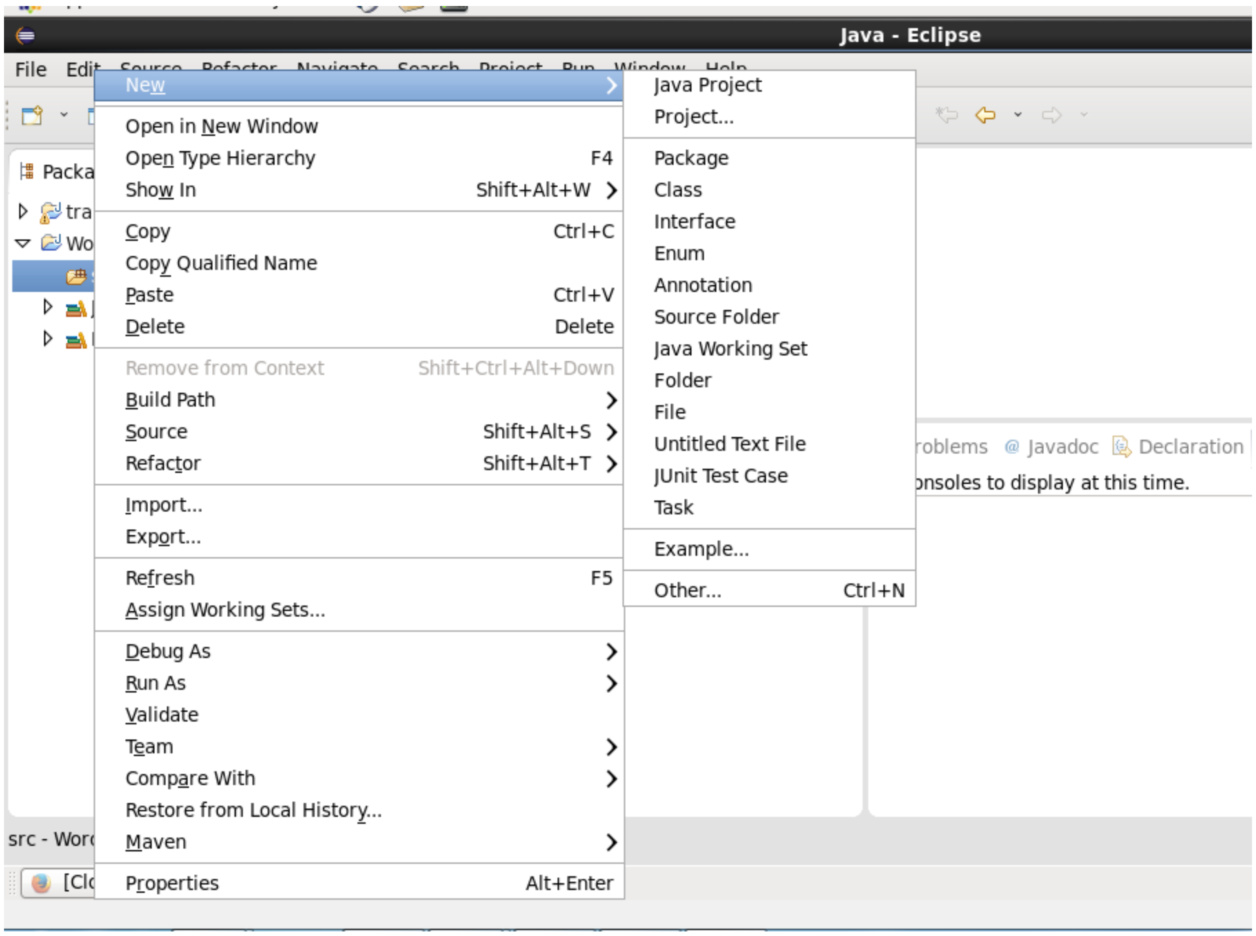
hadoop-common-tests.jar - /usr/lib/hadoop

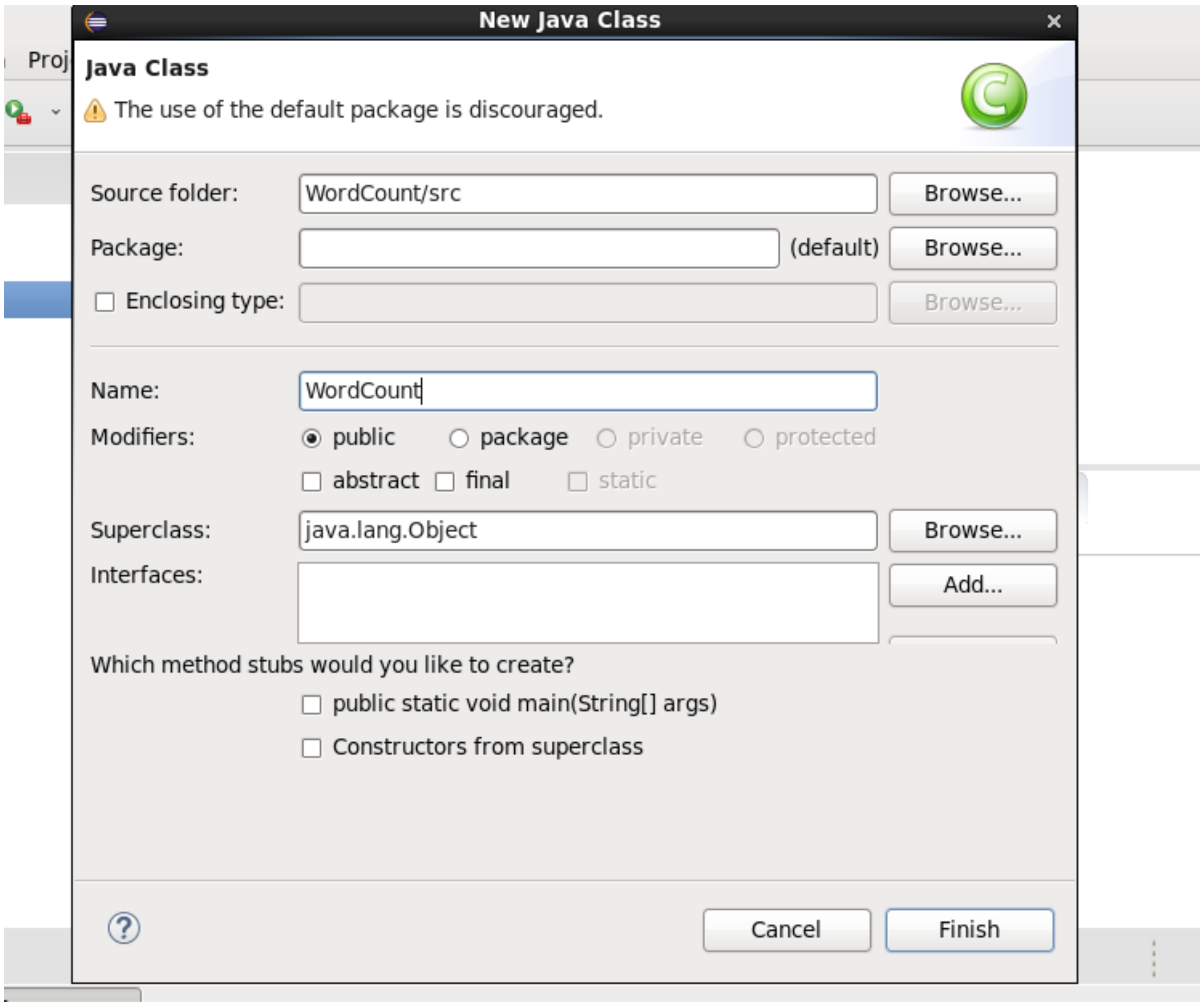
hadoop-nfs.jar - /usr/lib/hadoop

hadoop-nfs-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop

Creating class files

Right click on source, New > Class:





New Java Class

Java Class

The use of the default package is discouraged.

Source folder: WordCount/src

Package: (default)

☐ Enclosing type:

Name: WordCount

Modifiers: [✓ public] package private protected

☐ abstract ☐ final ☐ static

Superclass: java.lang.Object

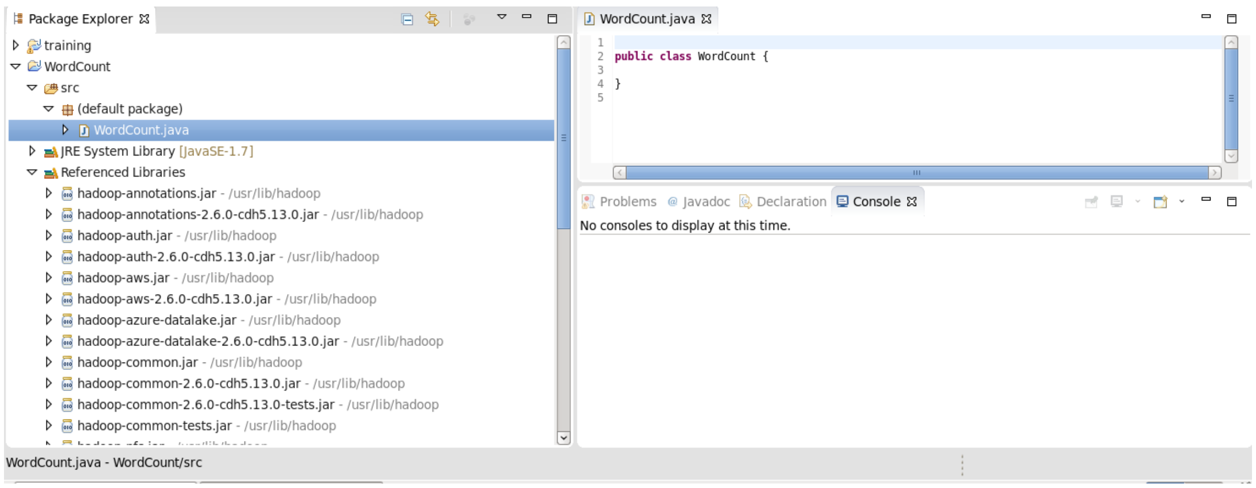
Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

Click "Finish".



Package Explorer

- training
- WordCount
  - src
    - (default package)
      - WordCount.java
  - JRE System Library [JavaSE-1.7]
  - Referenced Libraries
    - hadoop-annotations.jar - /usr/lib/hadoop
    - hadoop-annotations-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop
    - hadoop-auth.jar - /usr/lib/hadoop
    - hadoop-auth-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop
    - hadoop-aws.jar - /usr/lib/hadoop
    - hadoop-aws-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop
    - hadoop-azure-datalake.jar - /usr/lib/hadoop
    - hadoop-azure-datalake-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop
    - hadoop-common.jar - /usr/lib/hadoop
    - hadoop-common-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop
    - hadoop-common-2.6.0-cdh5.13.0-tests.jar - /usr/lib/hadoop
    - hadoop-common-tests.jar - /usr/lib/hadoop

WordCount.java

```

public class WordCount {

}

```

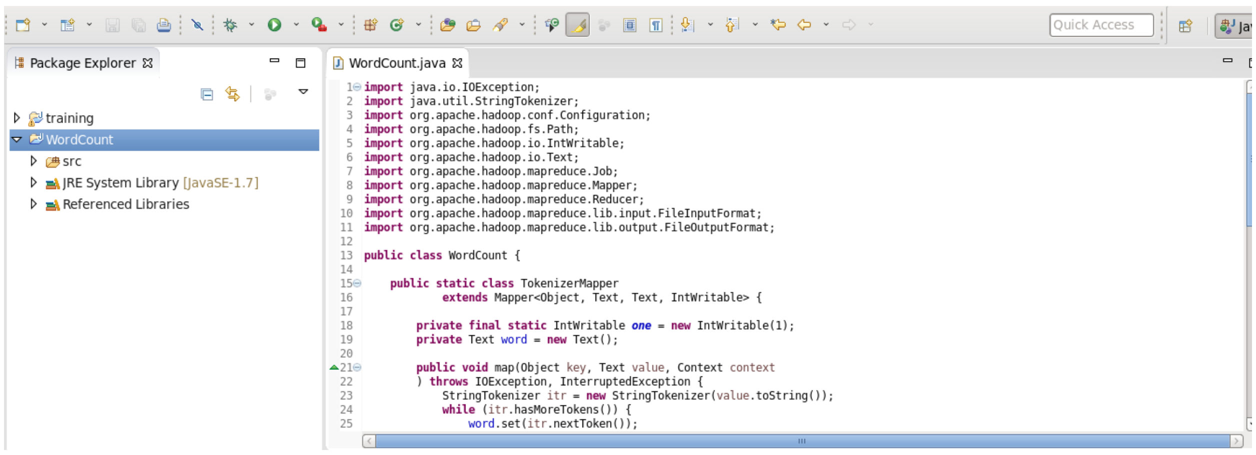
Problems Javadoc Declaration Console

No consoles to display at this time.

WordCount.java - WordCount/src

Copy and paste wordcount code:

```
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);
      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);
  }
}
```

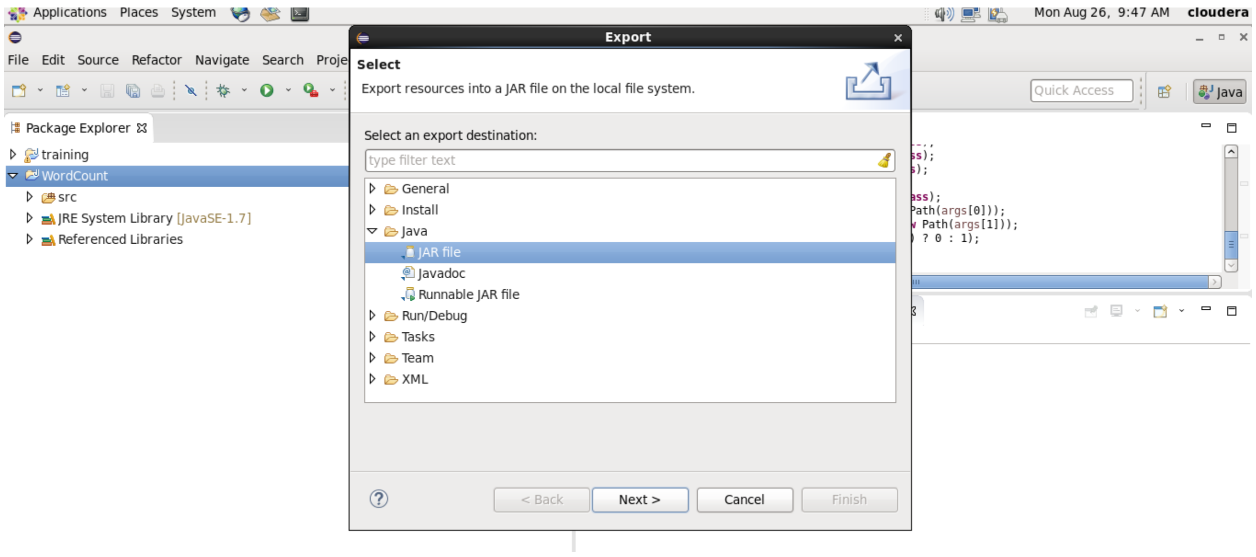


```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
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());
```

Exporting the Jar

Now we want to export it as a jar.

Right click on WordCount project and select "Export...":



Applications Places System Mon Aug 26, 9:47 AM cloudera

File Edit Source Refactor Navigate Search Proje

Package Explorer

- training
- WordCount
  - src
  - JRE System Library [JavaSE-1.7]
  - Referenced Libraries

Export

Select

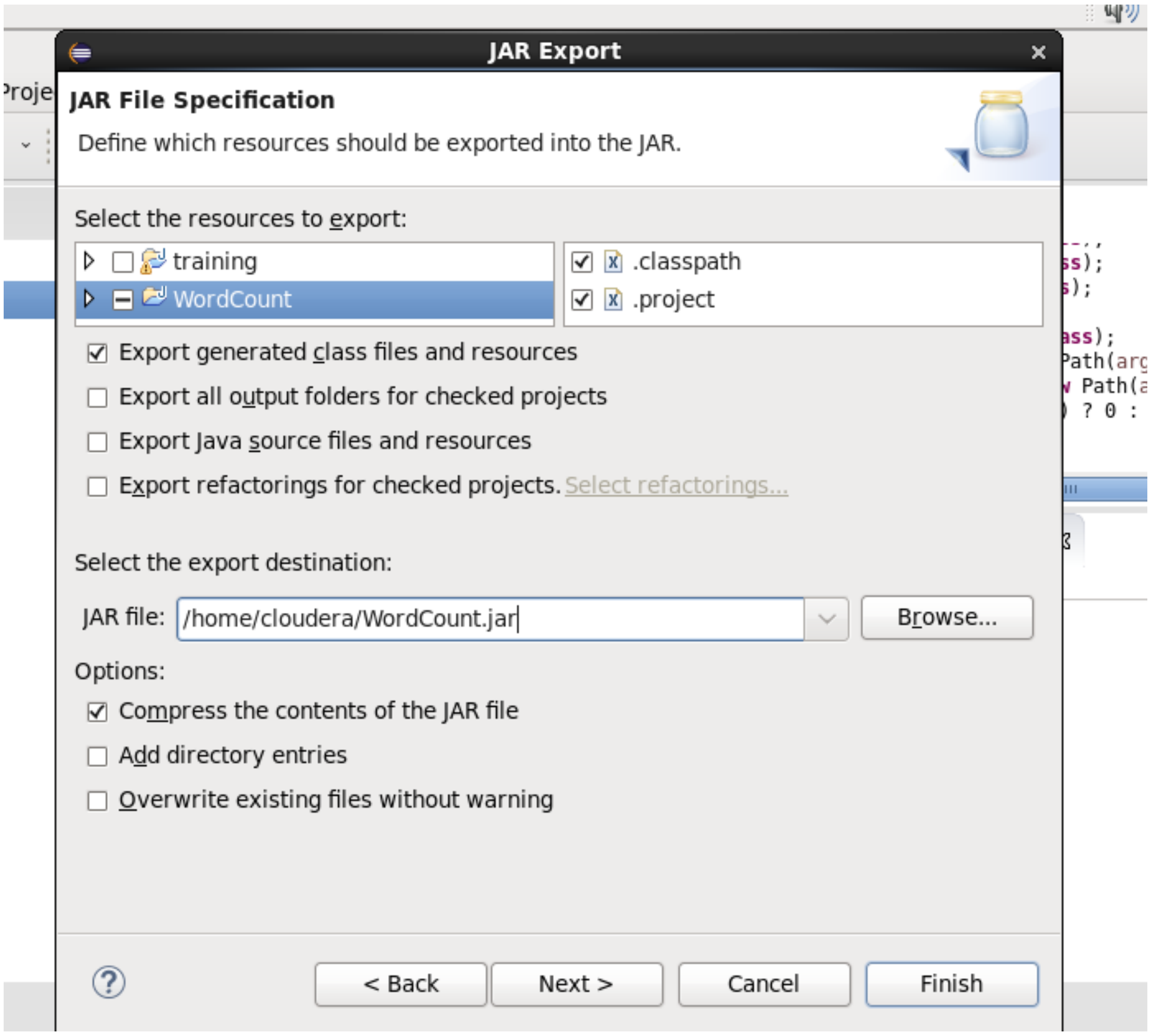
Export resources into a JAR file on the local file system.

Select an export destination:

type filter text

- General
- Install
- Java
  - JAR file
  - Javadoc
  - Runnable JAR file
- Run/Debug
- Tasks
- Team
- XML

< Back | Next > | Cancel | Finish



JAR Export

JAR File Specification

Define which resources should be exported into the JAR.

Select the resources to export:

- ☐ training
- ▬ WordCount

- ☑ .classpath
- ☑ .project

☑ Export generated class files and resources

☐ Export all output folders for checked projects

☐ Export Java source files and resources

☐ Export refactorings for checked projects. Select refactorings...

Select the export destination:

JAR file: /home/cloudera/WordCount.jar | Browse...

Options:

☑ Compress the contents of the JAR file

☐ Add directory entries

☐ Overwrite existing files without warning

< Back | Next > | Cancel | Finish

Input file

Here is our input:

```
[cloudera@quickstart ~]$ cat /home/cloudera/wordcount.txt
```

Copy paste input text here:

```
[cloudera@quickstart ~]$ hadoop fs -mkdir input

[cloudera@quickstart ~]$ hadoop fs -put /home/cloudera/wordcount.tx
t input/

[cloudera@quickstart ~]$ hadoop fs -ls input

Found 1 items

-rw-r--r--   1 cloudera cloudera     input/wordcount.txt

[cloudera@quickstart ~]$ hadoop fs -cat input/wordcount.txt

Your input text

 [cloudera@quickstart ~]$
```

run

Time to run MapReduce job:

```
$ hadoop jar /home/cloudera/WordCount.jar WordCount input/wordcount
.txt output
```

Output

Here is our output from the MR run:

```
[cloudera@quickstart ~]$ hadoop fs -ls output
 Found 2 items -rw-r--r-- 1 cloudera cloudera output/_SUCCESS -rw-r--r-- 1
cloudera cloudera output/part-00000
[cloudera@quickstart ~]$ hadoop fs -cat output/part-00000
```

Issues found:

Connection issues to connect to cloudera

Restart all services using cloudera manager