Phân tích dữ liệu lớn

# MapReduce - Lập trình chương trình WordCount

Apr 24, 2022

## I. Thử nghiệm với hàm mẫu WordCount của Hadoop

Trong thư mục **C:\hadoop-3.3.0\share\hadoop\mapreduce** Hadoop đã có sẵn chương trình MapReduce **hadoop-mapreduce-examples-3.3.0.jar**. Ta sẽ thử nghiệm bài toán đếm từ bằng cách tạo ra file text chứa dữ liệu và đầu ra mong muốn là các cặp **[từ: số lượng xuất hiện]**

### Bước 1: Tạo file data.txt

Nội dung của file data.txt là:

```
Bus Car bus
car train car
bus car train
bus TRAIN BUS
buS caR CAR
car BUS TRAIN
```

### Bước 2: Tạo thư mục input tại hdfs và lưu file data.txt

Tạo thư mục **input** trong hdfs với câu lệnh:

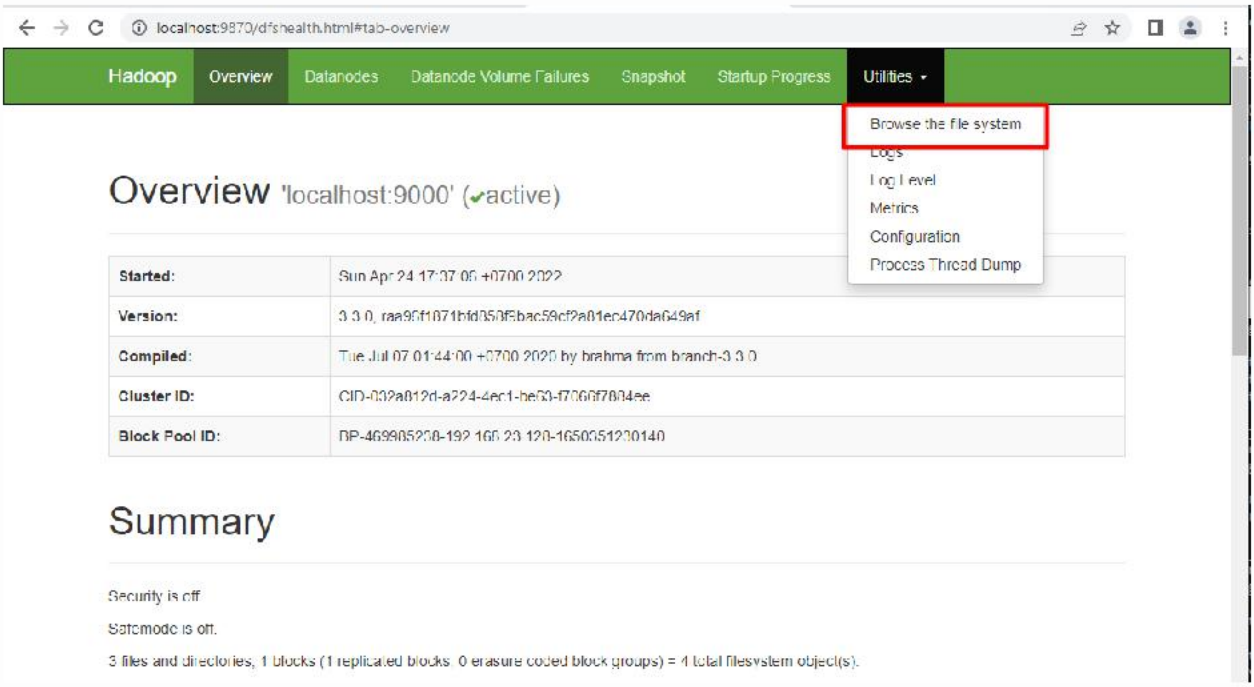
```
hdfs dfs -mkdir /input
```

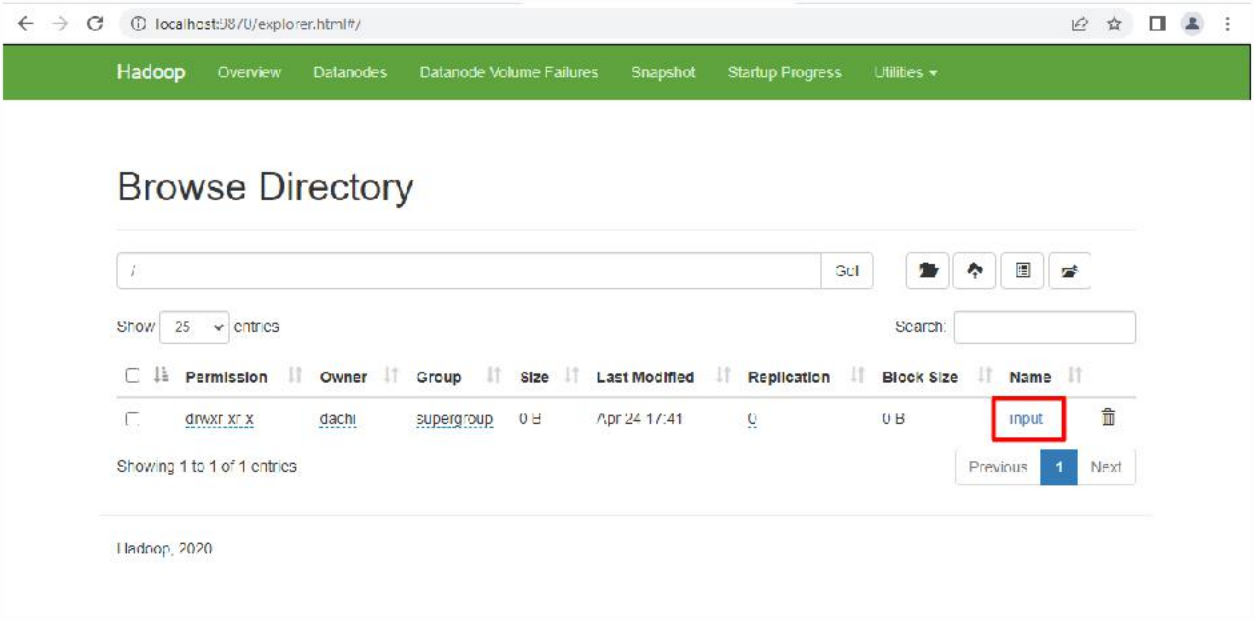
Đẩy file **data.txt** vào folder **input** vừa tạo:

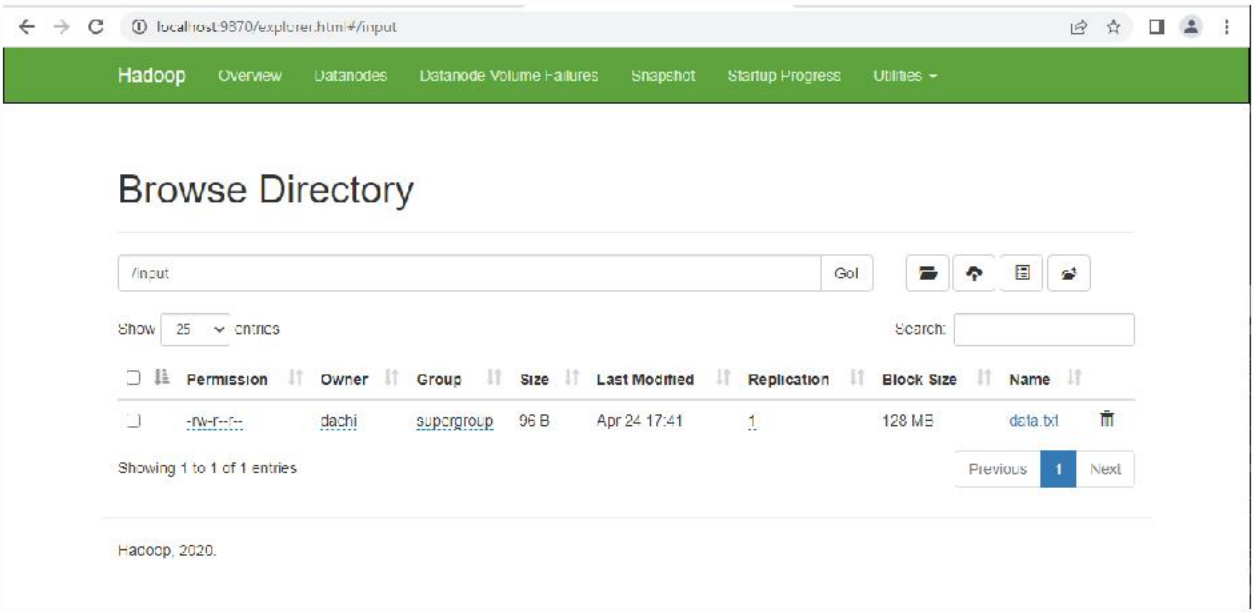
```
hdfs dfs -put "C:\input\data.txt" /input
```

> *Lưu ý: Thay "C:\input\data.txt" bằng nơi lưu trữ file trong máy*

Vào trang quản lý NameNode http://localhost:9870/ để kiểm tra file


localhost:9870/dfshealth.html#tab-overview
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Browse the file system
Logs
Log Level
Metrics
Configuration
Process Thread Dump
Overview 'localhost:9000' (active)
Started: Sun Apr 24 17:07:06 +0700 2022
Version: 3.3.0, raa96f1071bfd050f8bac59cf2a01ec470da649af
Compiled: Tue Jul 07 01:44:00 +0700 2020 by brahma from branch-3.3.0
Cluster ID: CID-002a012d-a224-4ec1-be30-f7066f7804ee
Block Pool ID: BP-469905206-192.168.23.120-1650051230140
Summary
Security is off.
Safemode is off.
3 files and directories, 1 blocks (1 replicated blocks, 0 erasure coded block groups) = 4 total filesystem object(s).

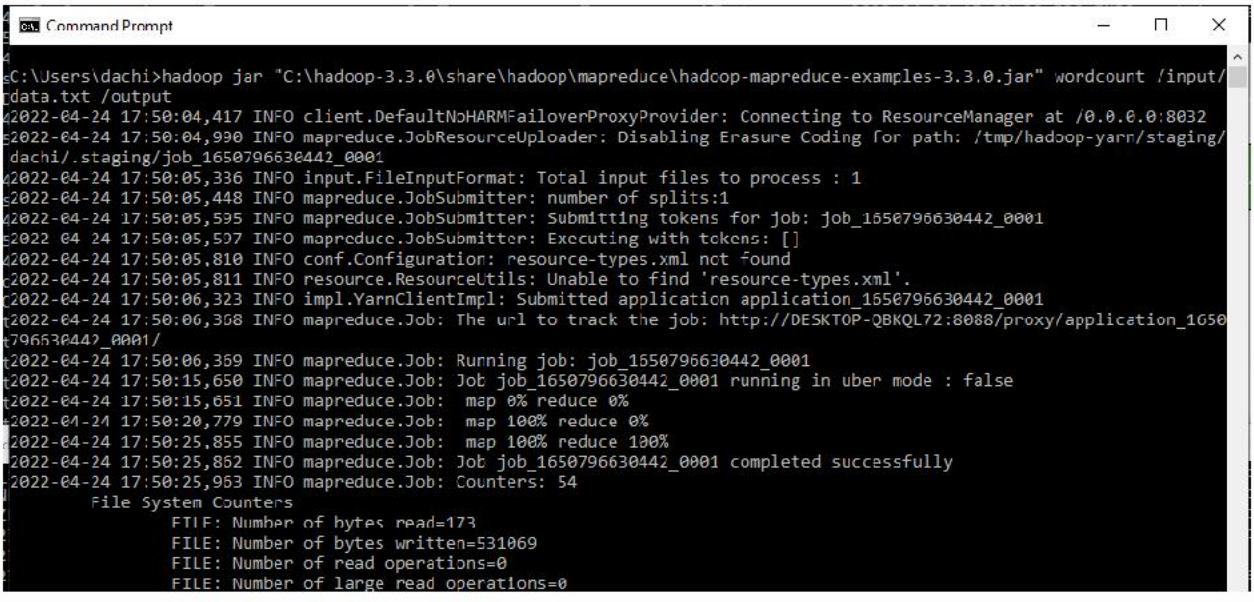

localhost:98/0/explorer.html#/
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Browse Directory
/
Go!
Show 25 entries
Search:
Permission
Owner
Group
Size
Last Modified
Replication
Block Size
Name
drwxr-xr-x
dachi
supergroup
0 B
Apr 24 17:41
0
0 B
input
Showing 1 to 1 of 1 entries
Previous
1
Next
Hadoop, 2020



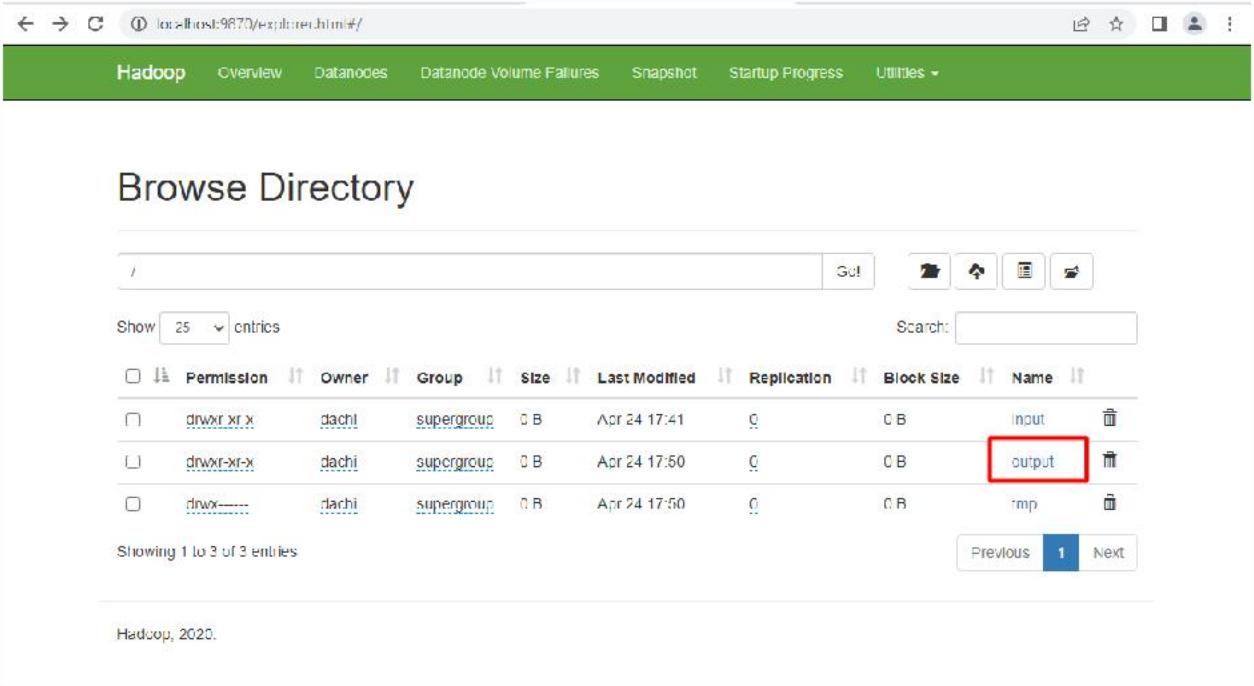

Bước 3: Chạy chương trình MapReduce và xem kết quả

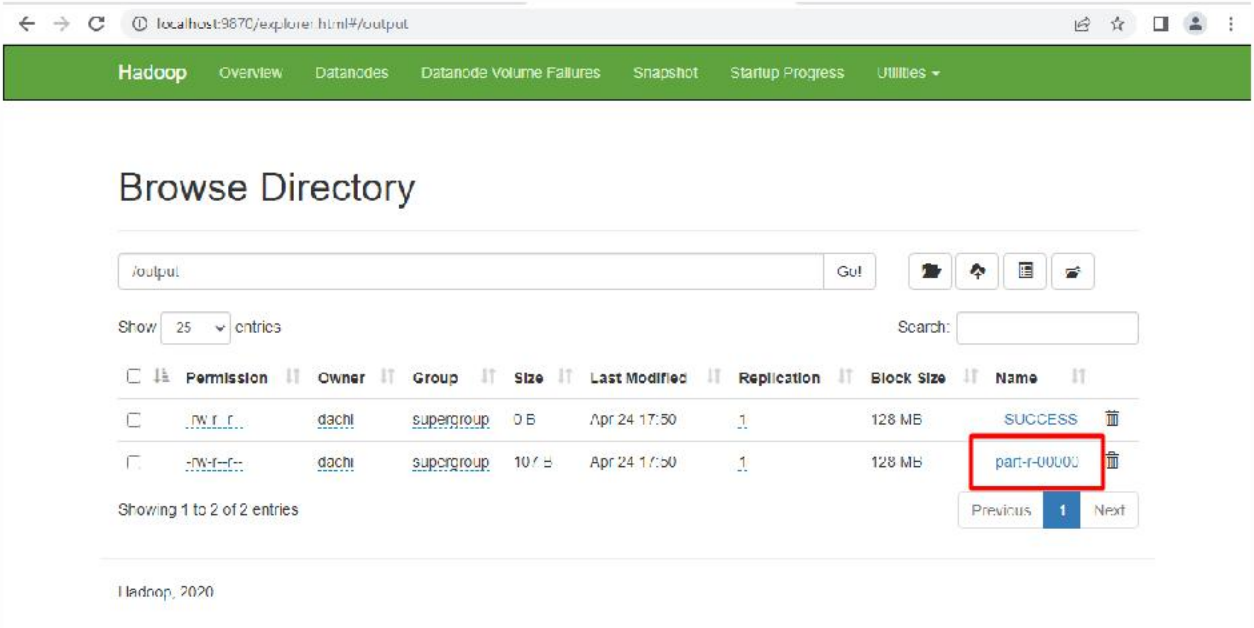
Chương trình mẫu MapReduce của Hadoop nằm tại **C:\hadoop-3.3.0\share\hadoop\mapreduce\hadoop-mapreduce-examples-3.3.0.jar**. Ta sẽ thử nghiệm đầu vào chương trình là file **data.txt** và kết quả sẽ được lưu tại folder **/output**, lệnh thực hiện"

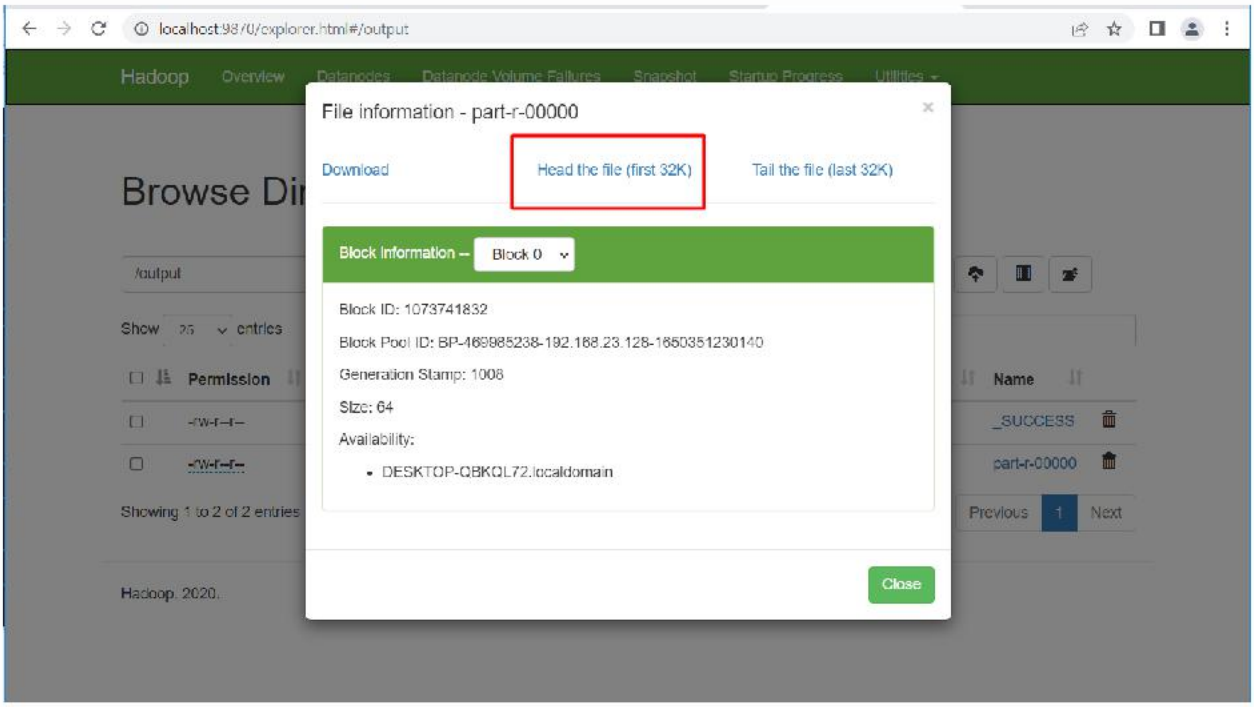
```
hadoop jar "C:\hadoop-3.3.0\share\hadoop\mapreduce\hadoop-mapreduce-examples-3.3.0.jar" wordcount /input/data.txt /output
```

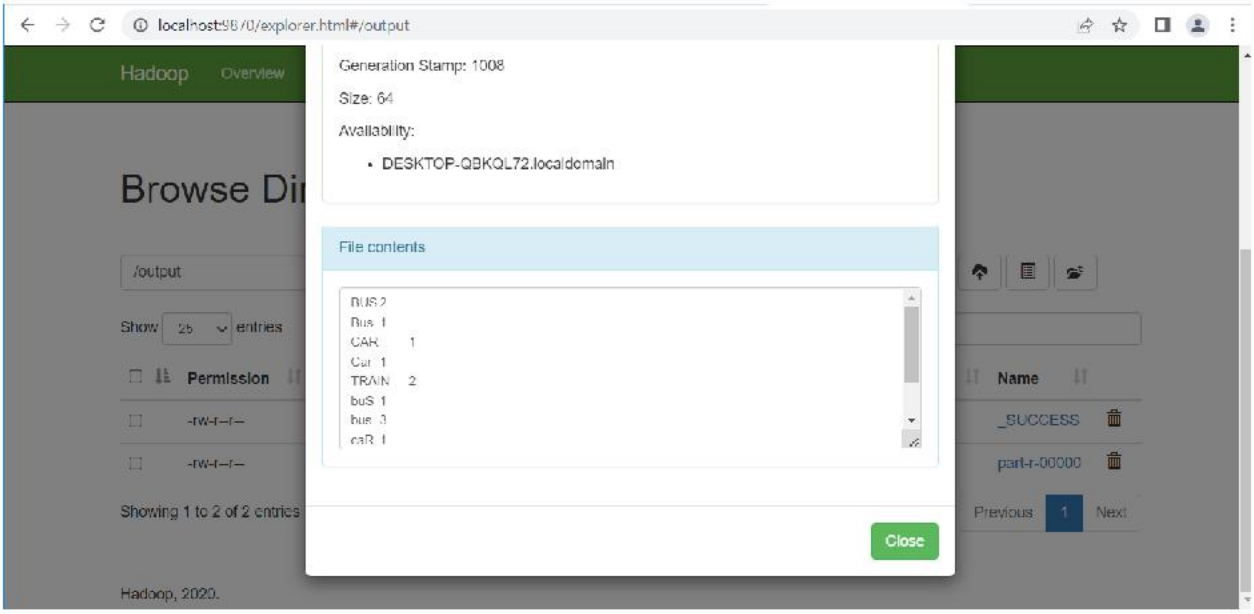



Xem kết quả thu được:


localhost:9870/explorer.html#/
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Browse Directory
/
Go!
Show 25 entries
Search:
Permission
Owner
Group
Size
Last Modified
Replication
Block Size
Name
drwxr-xr-x
dachi
supergroup
0 B
Apr 24 17:41
0
0 B
input
drwxr-xr-x
dachi
supergroup
0 B
Apr 24 17:50
0
0 B
output
drwx------
dachi
supergroup
0 B
Apr 24 17:50
0
0 B
tmp
Showing 1 to 3 of 3 entries
Previous
1
Next
Hadoop, 2020.


localhost:9870/explorer.html#/output
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Browse Directory
/output
Go!
Show 25 entries
Search:
Permission
Owner
Group
Size
Last Modified
Replication
Block Size
Name
-rw-r--r--
dachi
supergroup
0 B
Apr 24 17:50
1
128 MB
SUCCESS
-rw-r--r--
dachi
supergroup
10 B
Apr 24 17:50
1
128 MB
part-r-00000
Showing 1 to 2 of 2 entries
Previous
1
Next
Hadoop, 2020

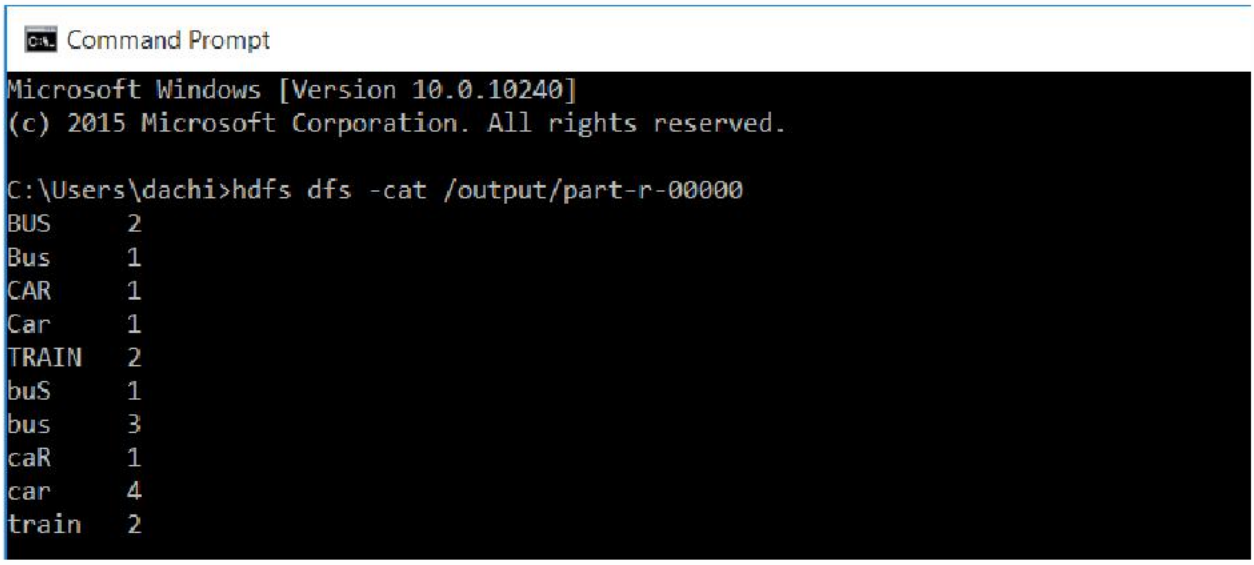



Hoặc dùng lệnh cmd:

```
hdfs dfs -cat /output/part-r-00000
```

> *Lưu ý: thay /output/part-r-00000 thành đường dẫn file muốn xem*

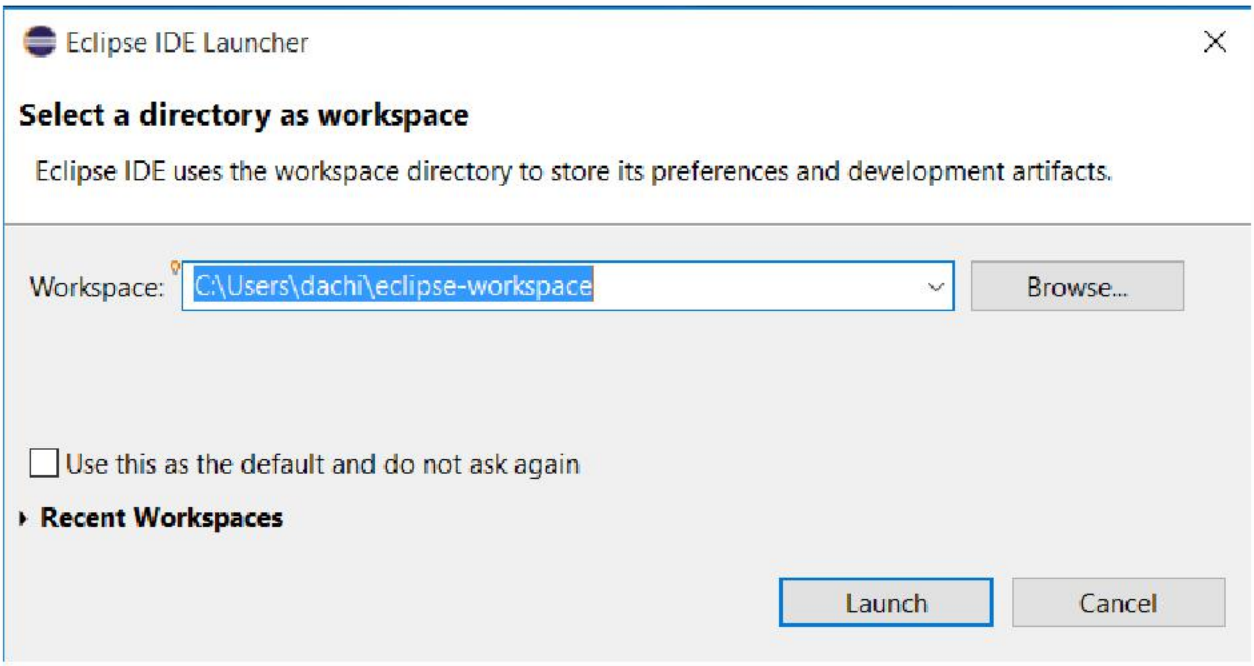



Như vậy ta đã chạy thành công chương trình mẫu MapReduce của Hadoop cung cấp.

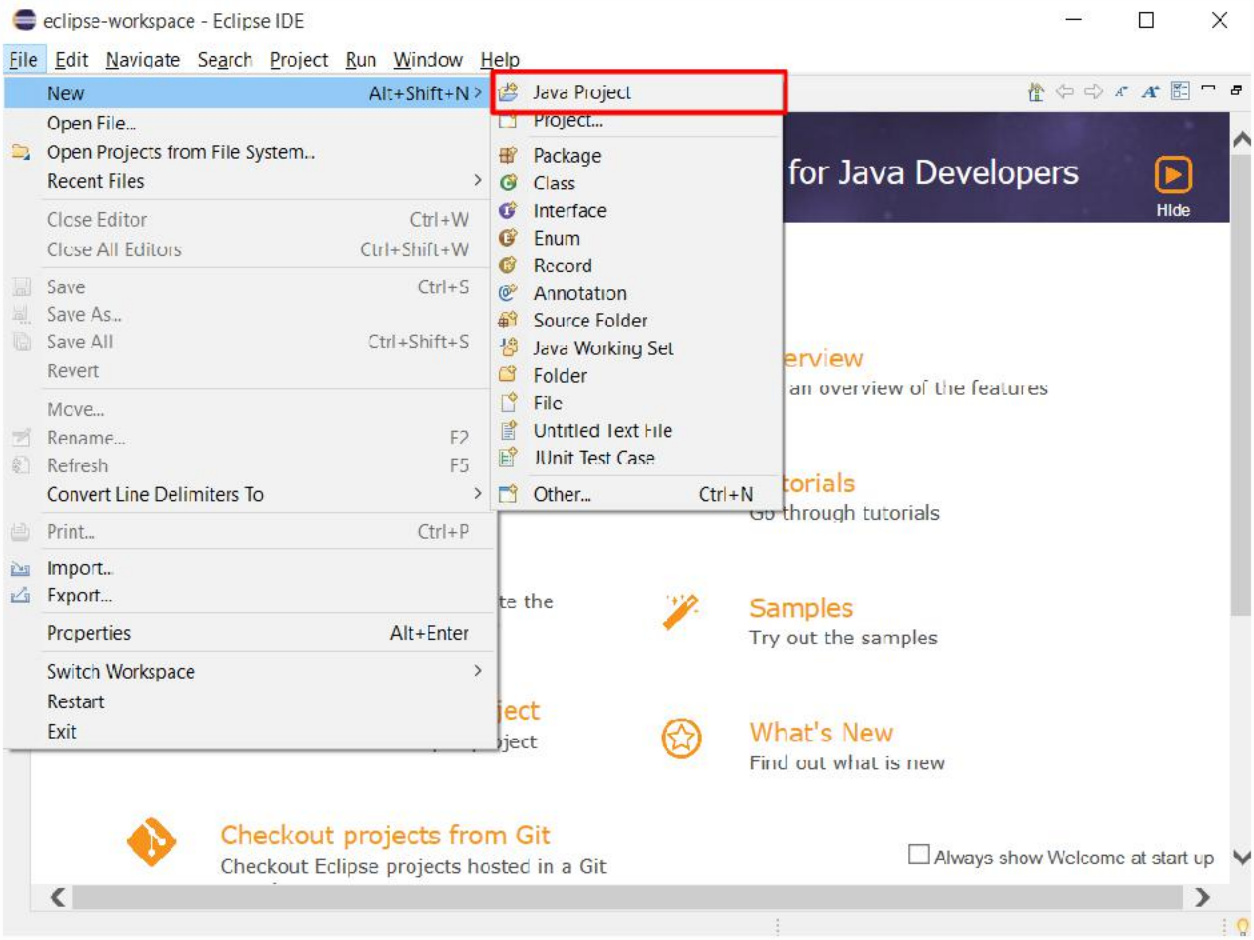
# II. Lập trình chương trình WordCount bằng Eclipse

## Bước 1: Tạo project

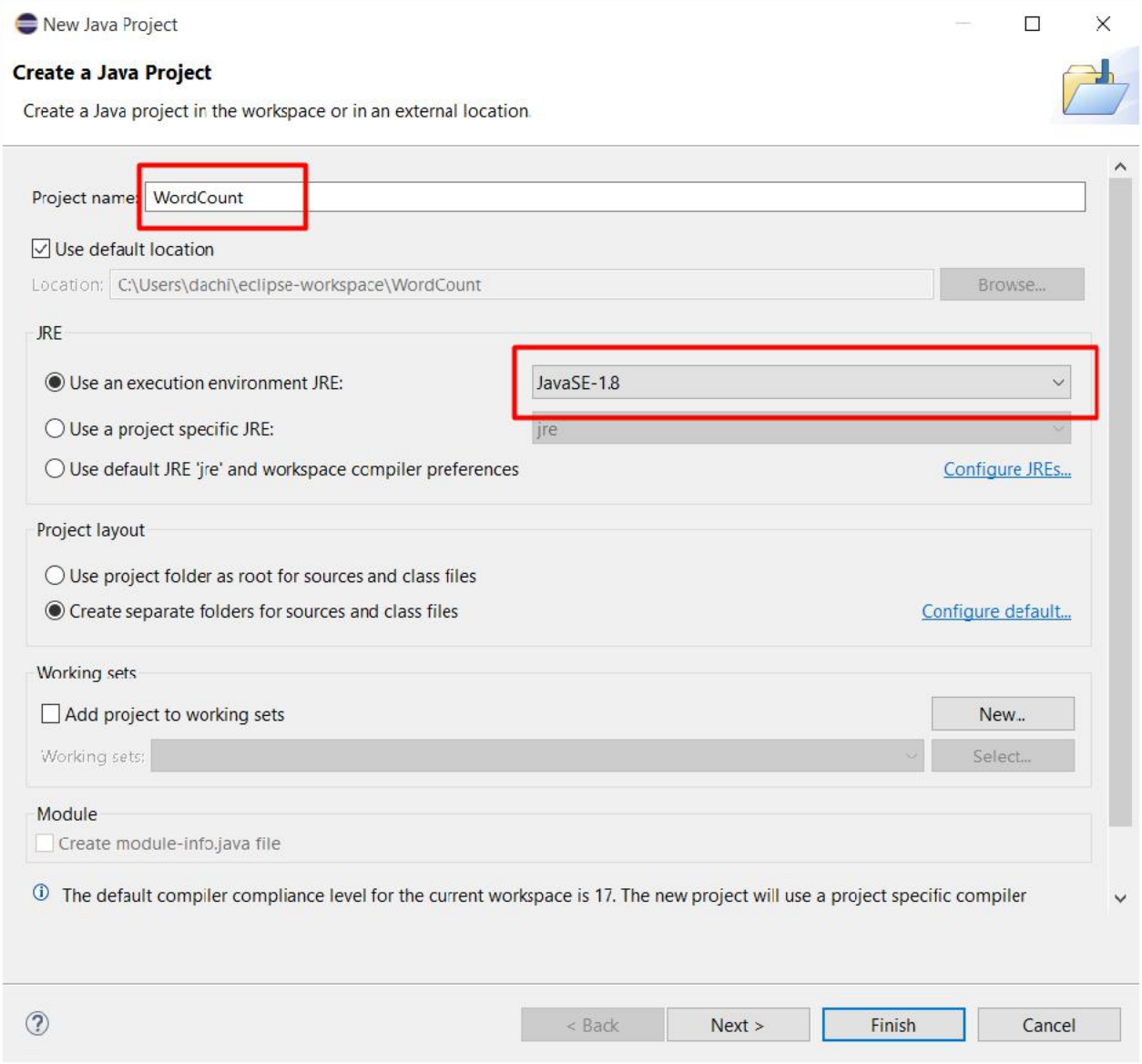
Mở chương trình Eclipse. Chọn **workspace** (nên để mặc định)




Tạo project Java, chọn **File > New > Java Project**

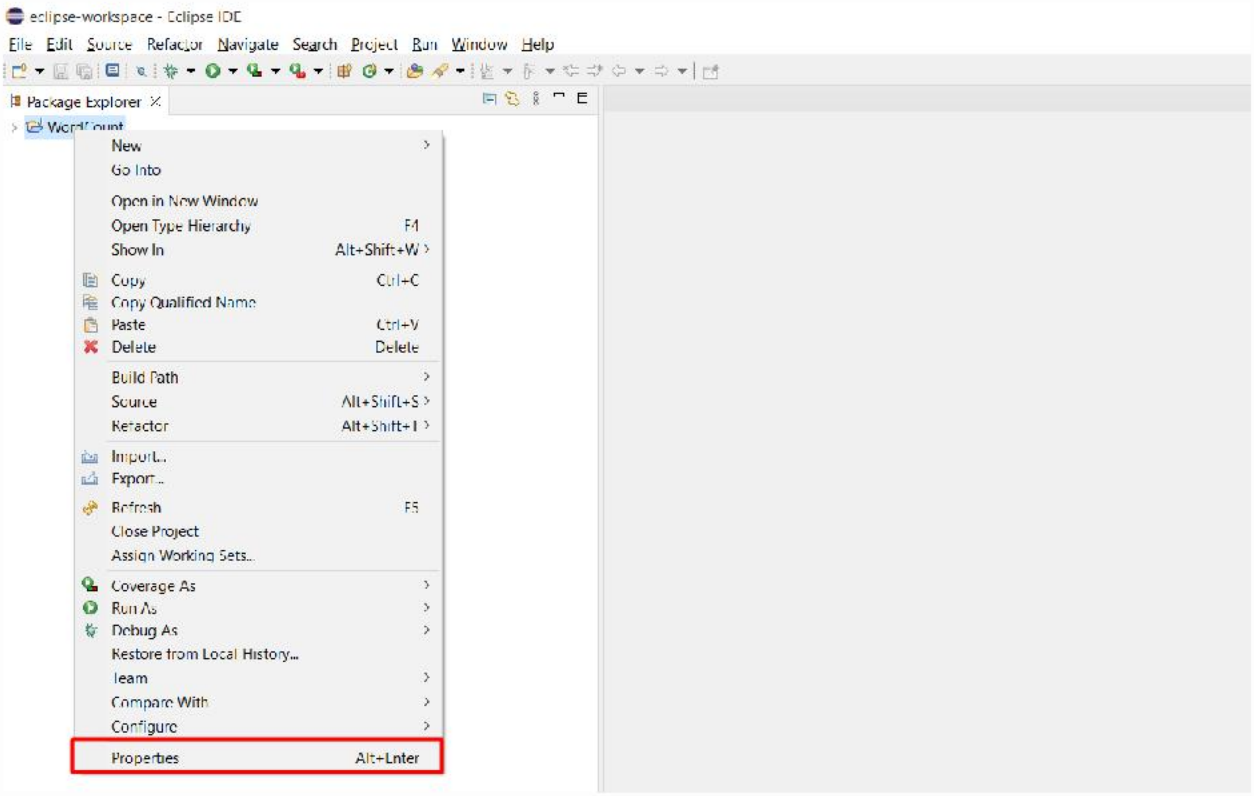



Đặt tên project là **WordCount** và chọn môi trường là **JavaSE-1.8**. Xong ấn **Finish**.

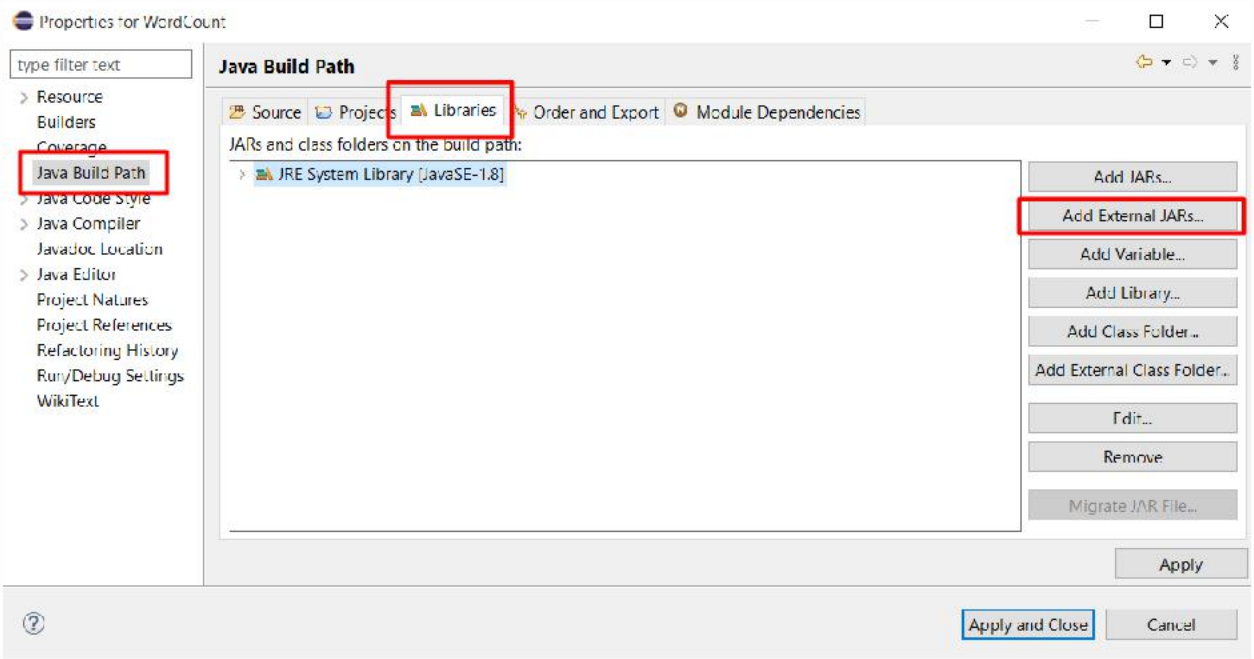



## Bước 2: Thêm thư viện cần thiết để chạy MapReduce

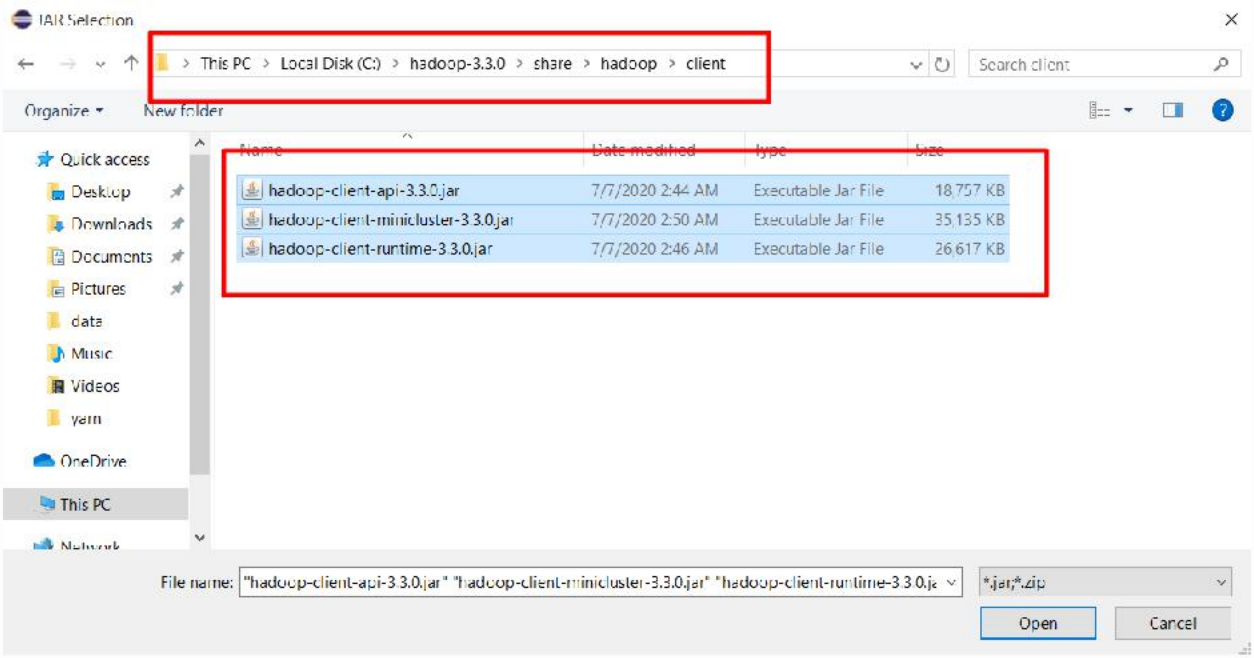
Chuột phải vào project **WordCount** chọn **Properties**



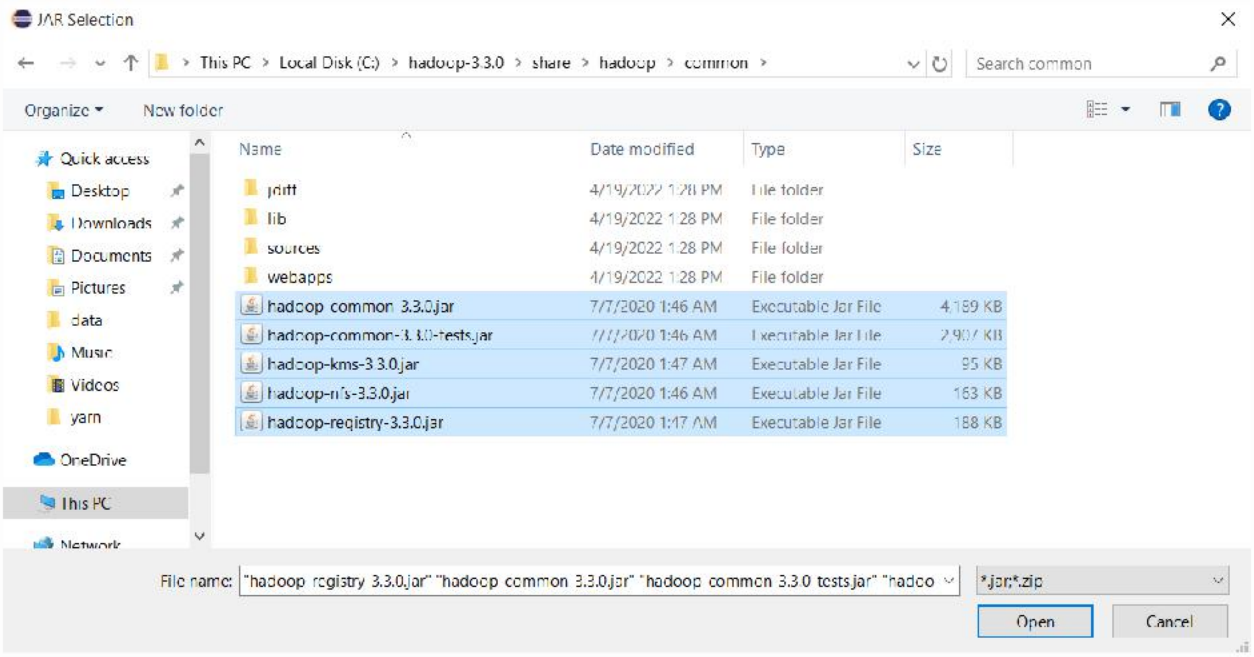

Chọn **Java Build Path**, chọn tab **Libraries** và bấm **Add External JARs**



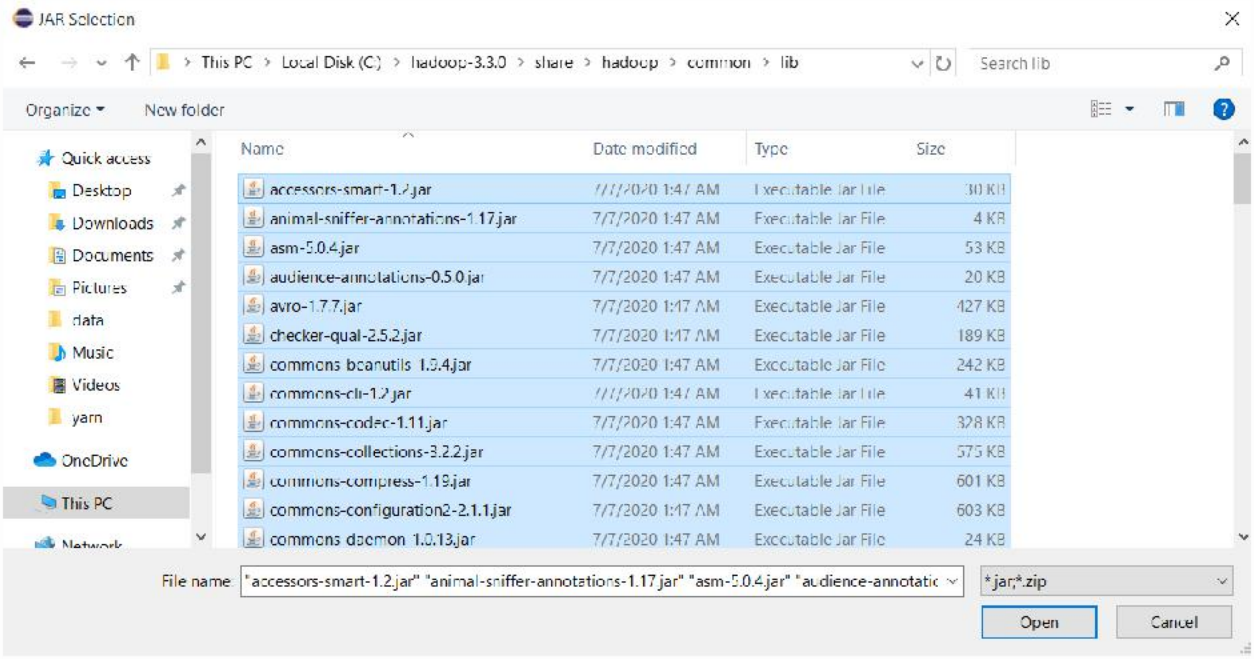

Chọn tất cả file trong thư mục **C:\hadoop-3.3.0\share\hadoop\client** và ấn **Open**



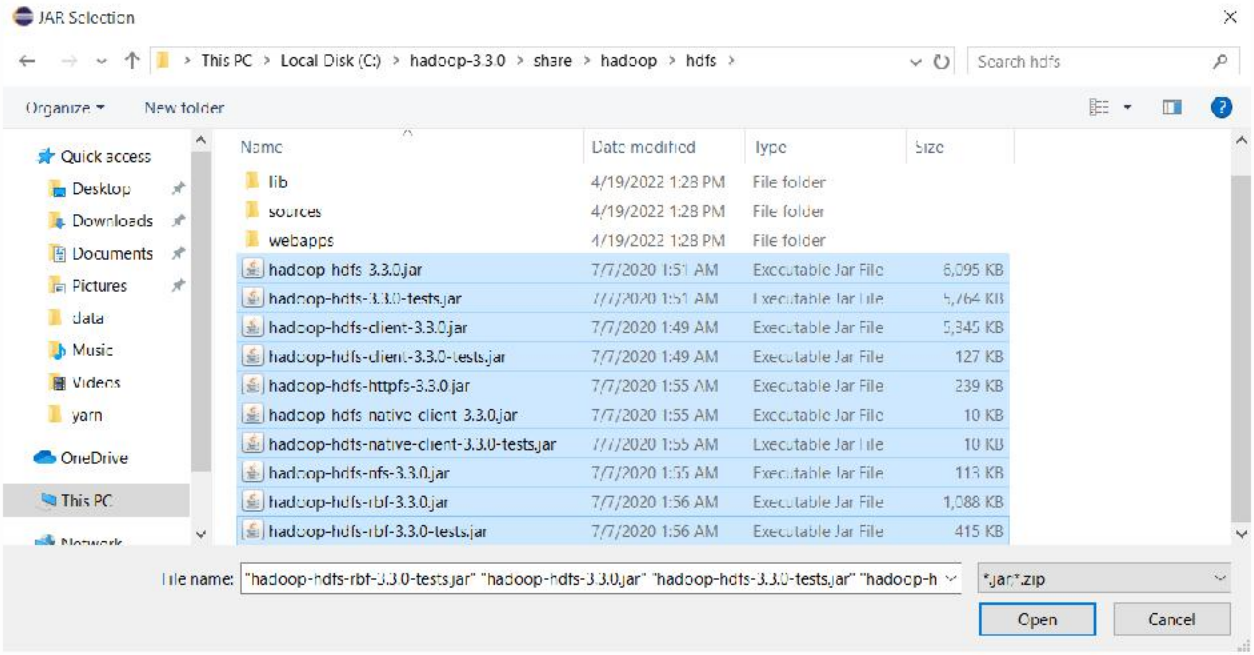

Tương tự chọn tất cả file trong thư mục **C:\hadoop-3.3.0\share\hadoop\common** và ấn **Open**



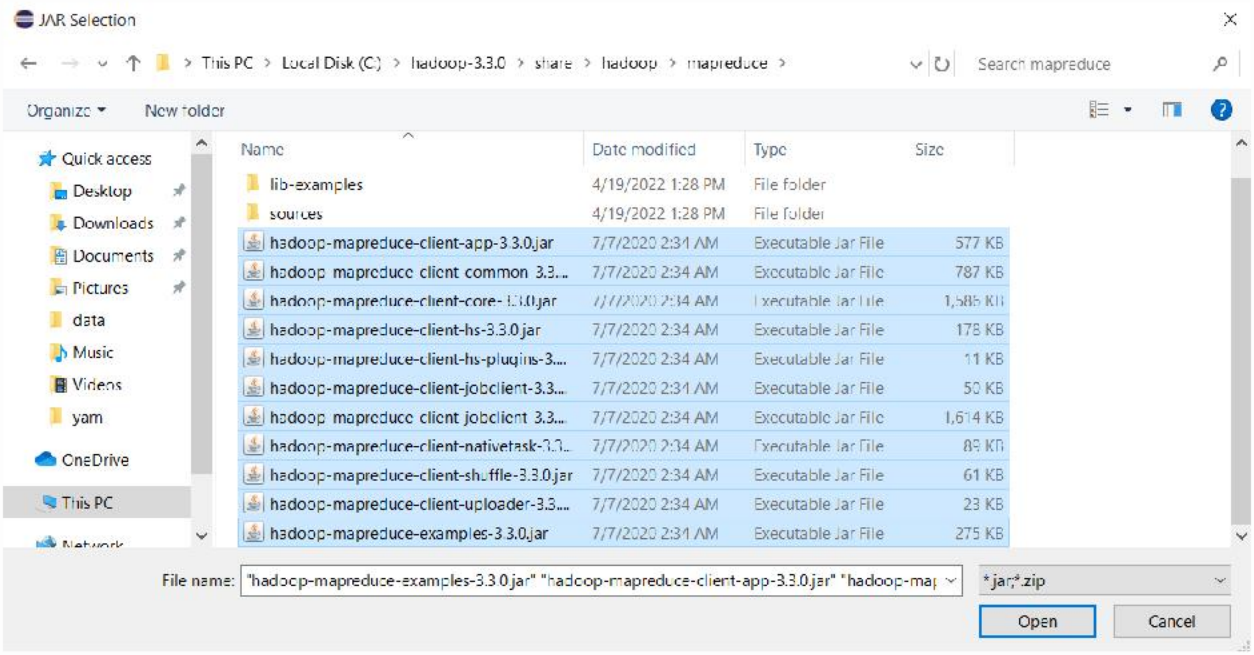

Chọn tất cả file trong thư mục **C:\hadoop-3.3.0\share\hadoop\common\lib** và ấn **Open**



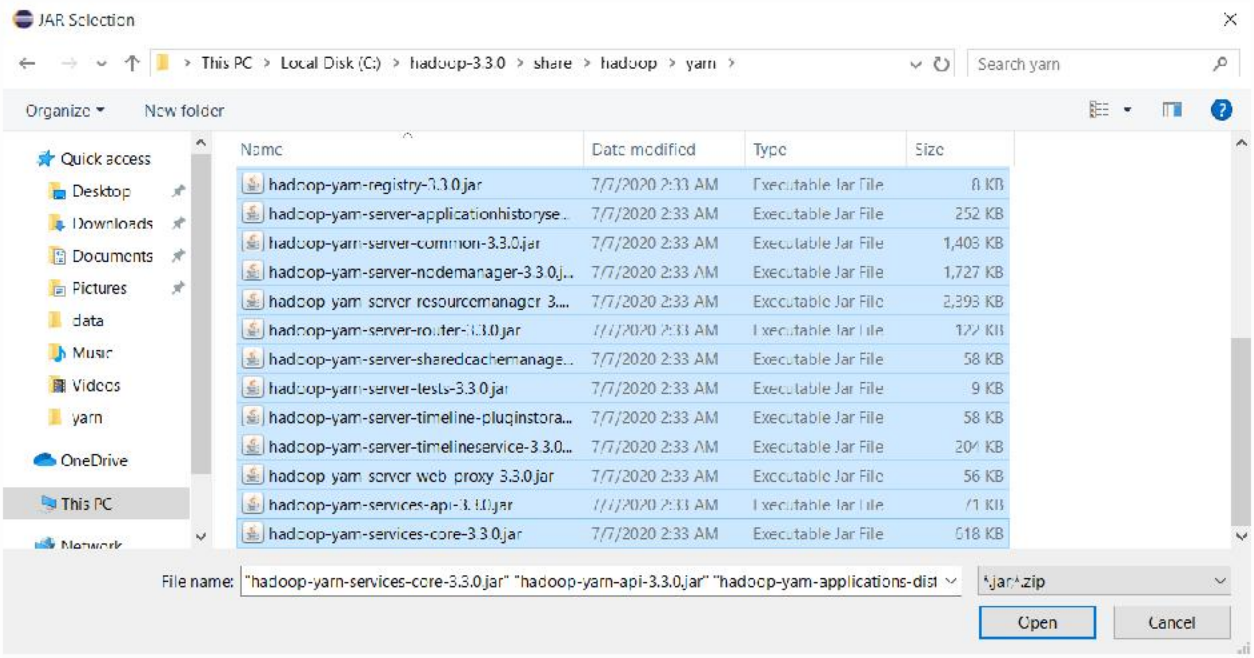

Chọn tất cả file trong thư mục **C:\hadoop-3.3.0\share\hadoop\hdfs** và ấn **Open**



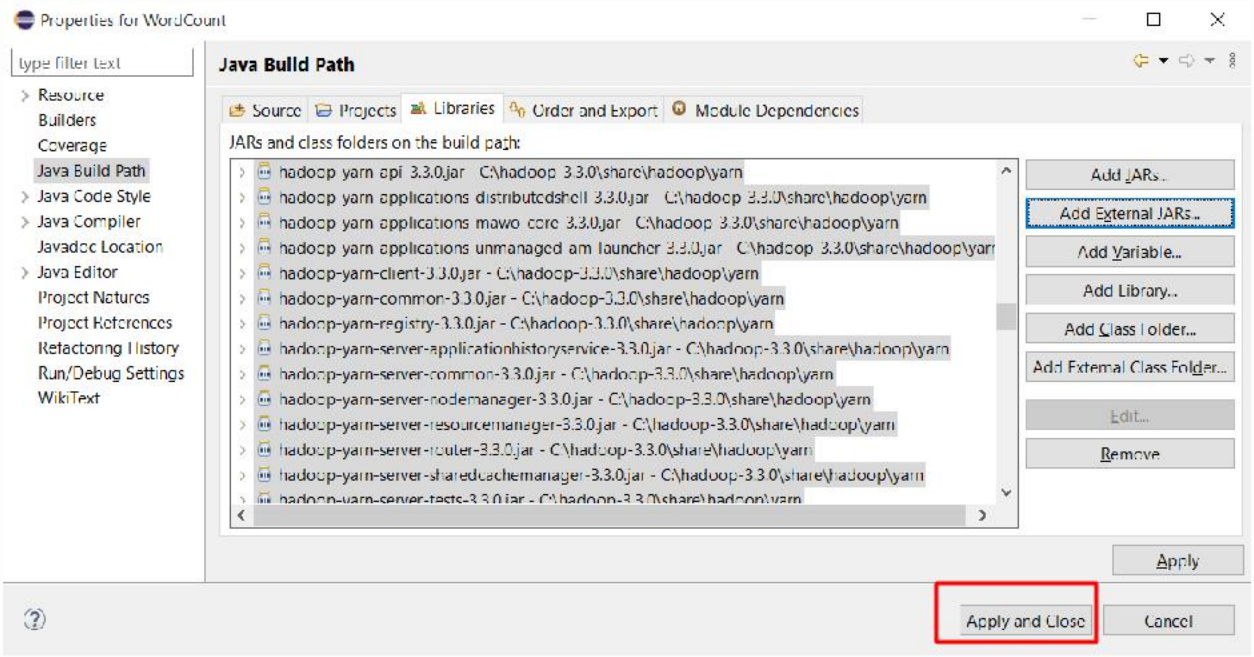

Chọn tất cả file trong thư mục **C:\hadoop-3.3.0\share\hadoop\mapreduce** và ấn **Open**




Chọn tất cả file trong thư mục **C:\hadoop-3.3.0\share\hadoop\yarn** và ấn **Open**

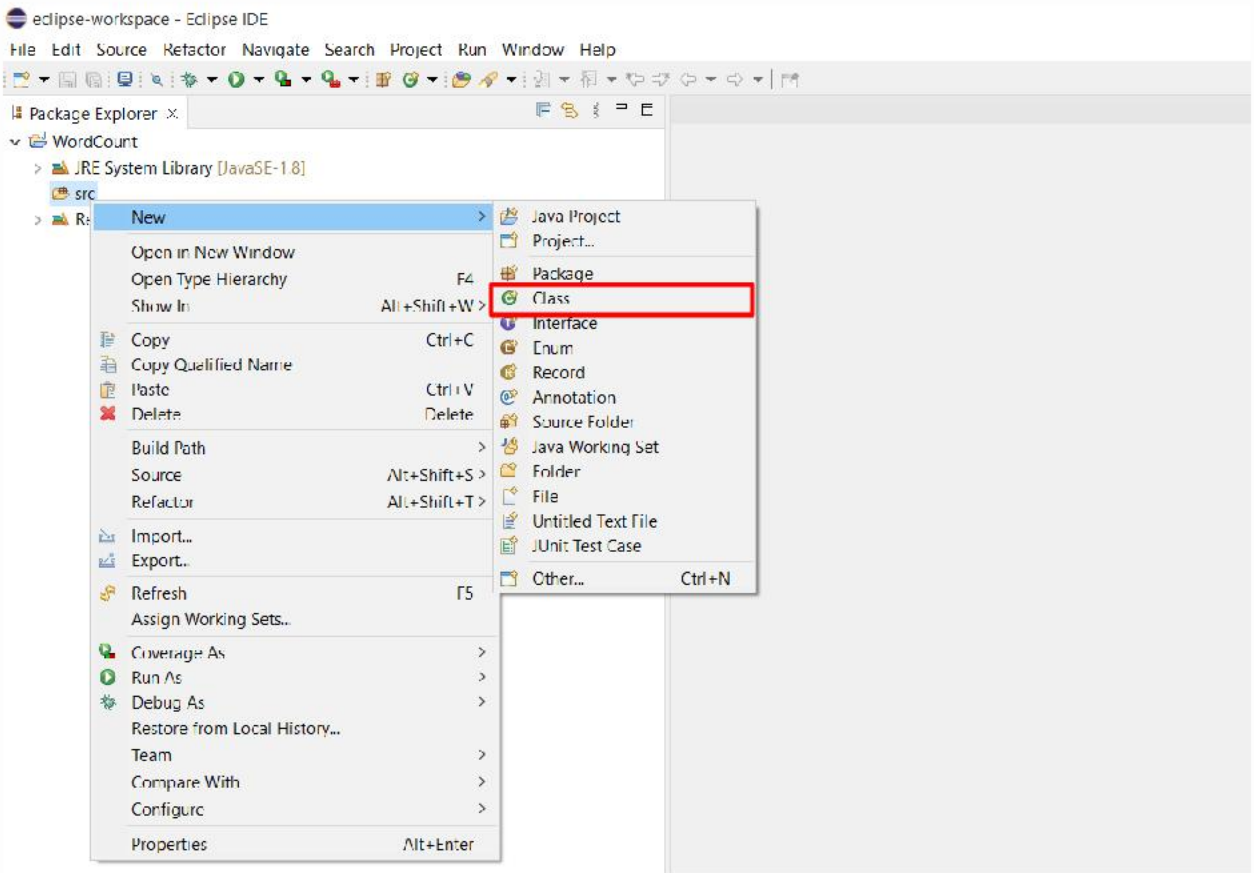



Ấn **Apply and Close**

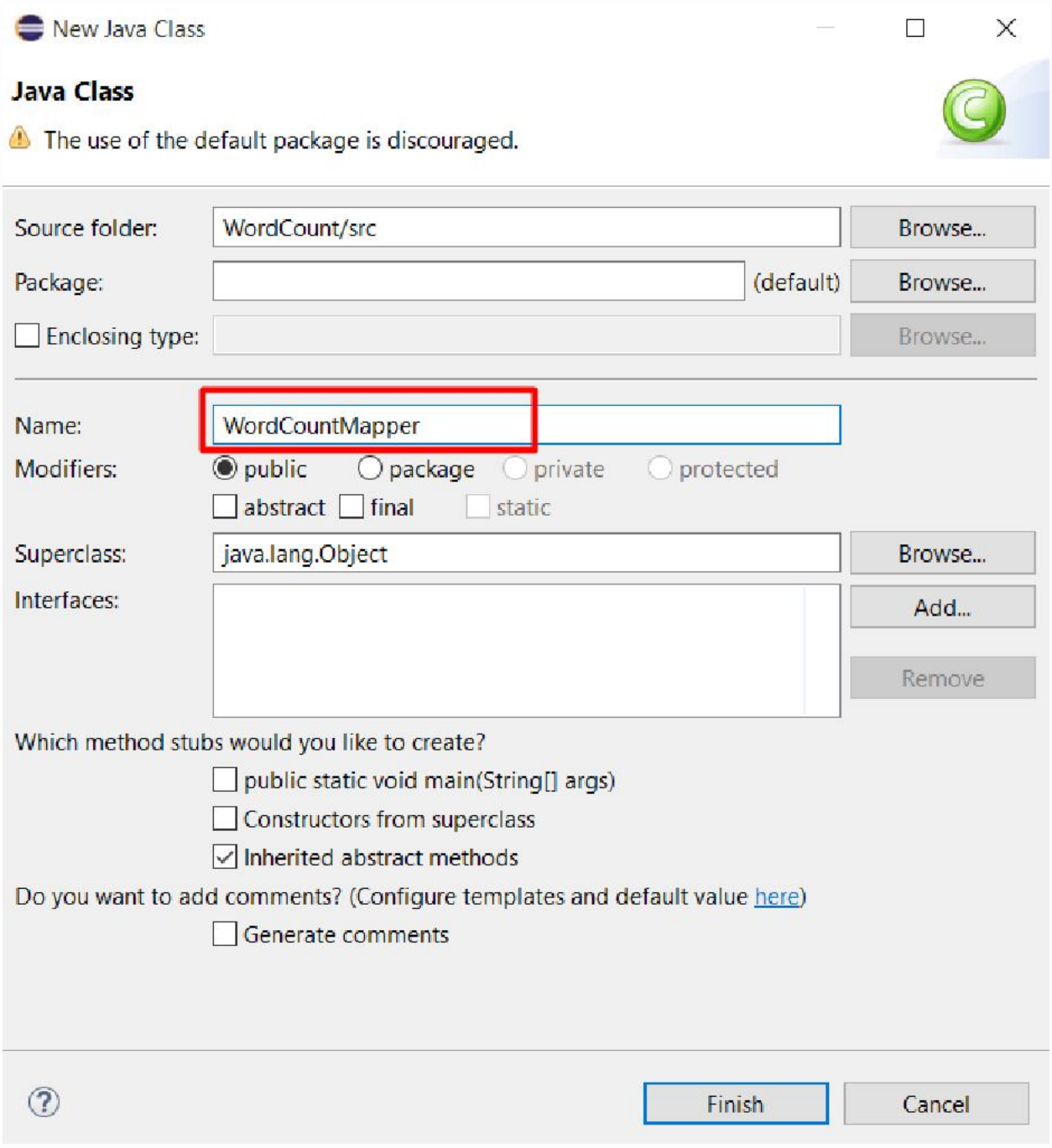



## Bước 3: Tạo class xử lý tác vụ MapReduce

Double click vào project **WordCount**, chuột phải vào **src** và chọn **New > Class**



Tạo class để xử lý nhiệm vụ **Map**, đặt tên là **WordCountMapper**




Nội dung bên trong file **WordCountMapper.java**:

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
```

```
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.io.LongWritable;

public class WordCountMapper extends Mapper <LongWritable, Text, Text, IntWritable>
{
  private Text wordToken = new Text();
  public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException
  {
    StringTokenizer tokens = new StringTokenizer(value.toString()); //Dividing
String into tokens
    while (tokens.hasMoreTokens())
    {
      wordToken.set(tokens.nextToken());
      context.write(wordToken, new IntWritable(1));
    }
  }
}
```

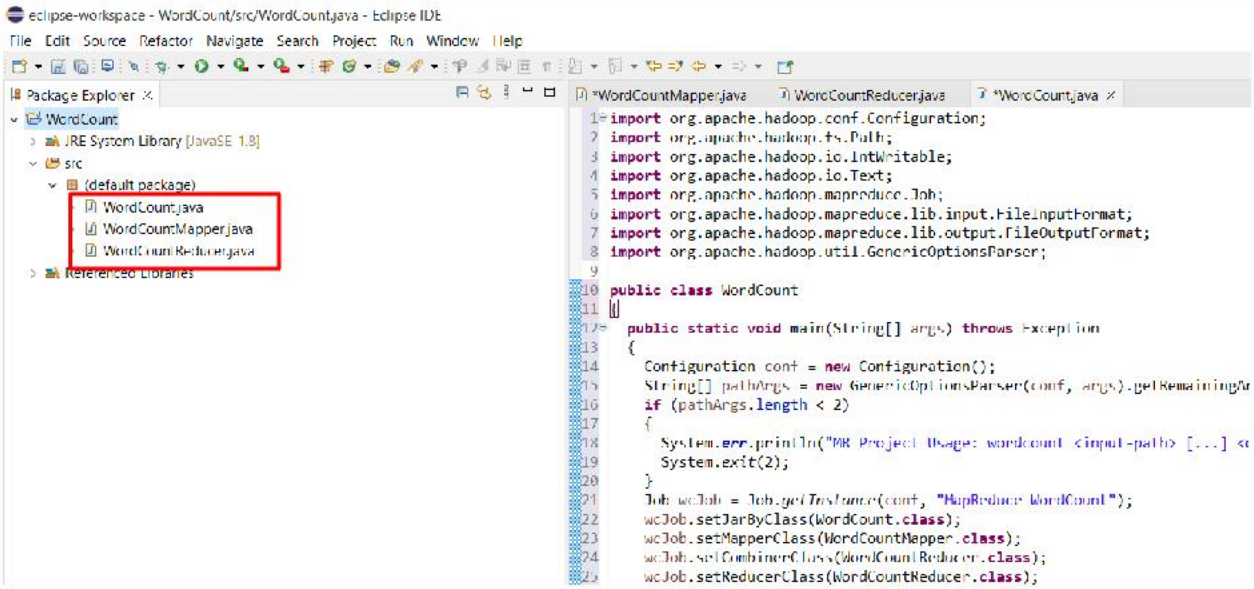
Tương tự tạo class xử lý nhiệm vụ **Reduce**, đặt tên là **WordCountReducer**:

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class WordCountReducer extends Reducer <Text, IntWritable, Text, IntWritable>
{
  private IntWritable count = new IntWritable();
  public void reduce(Text key, Iterable<IntWritable> values, Context context) throws
IOException, InterruptedException
  {
    int valueSum = 0;
    for (IntWritable val : values)
    {
      valueSum += val.get();
    }
    count.set(valueSum);
    context.write(key, count);
  }
}
```

Và tạo class **WordCount** chứa hàm **main** để khởi chạy chương trình:

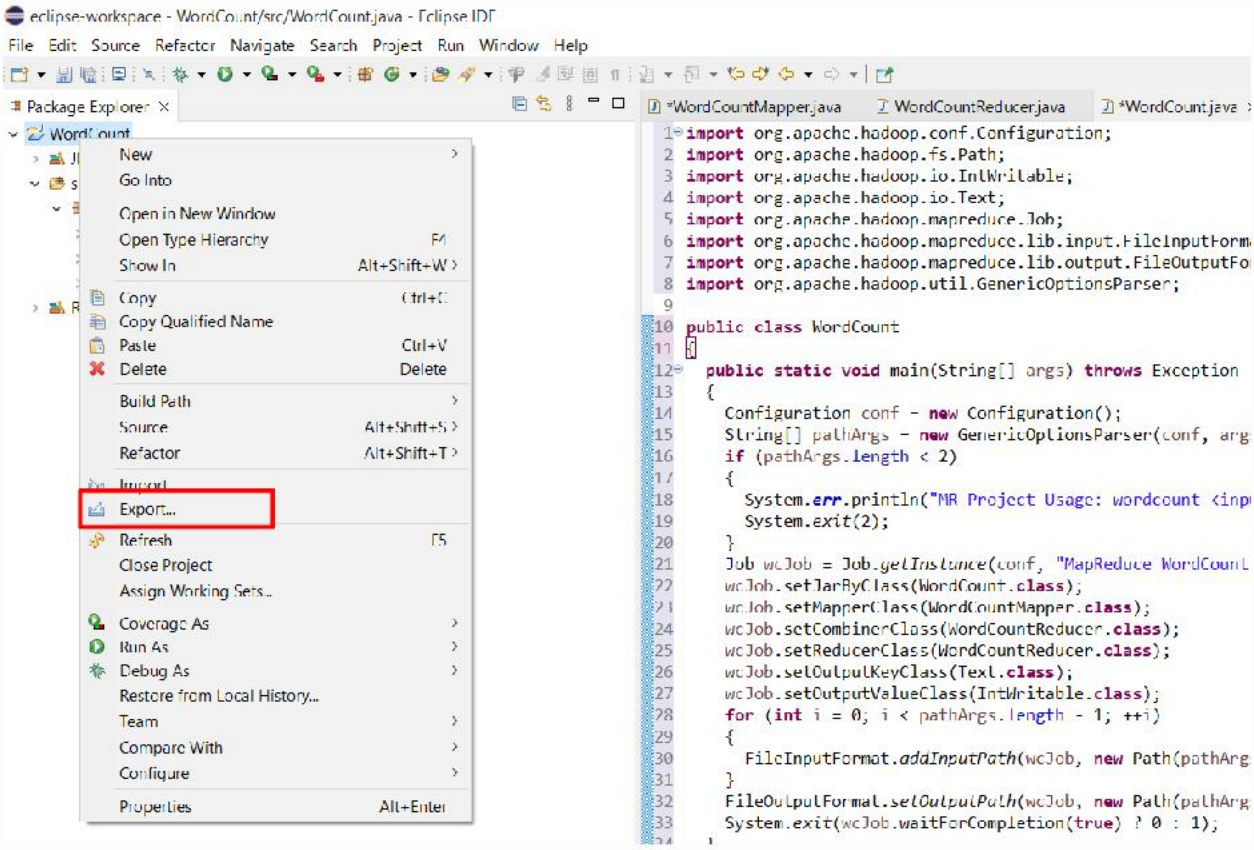
```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
```

```java
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;

public class WordCount
{
  public static void main(String[] args) throws Exception
  {
    Configuration conf = new Configuration();
    String[] pathArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
    if (pathArgs.length < 2)
    {
      System.err.println("MR Project Usage: wordcount <input-path> [...] <output-
path>");
      System.exit(2);
    }
    Job wcJob = Job.getInstance(conf, "MapReduce WordCount");
    wcJob.setJarByClass(WordCount.class);
    wcJob.setMapperClass(WordCountMapper.class);
    wcJob.setCombinerClass(WordCountReducer.class);
    wcJob.setReducerClass(WordCountReducer.class);
    wcJob.setOutputKeyClass(Text.class);
    wcJob.setOutputValueClass(IntWritable.class);
    for (int i = 0; i < pathArgs.length - 1; ++i)
    {
      FileInputFormat.addInputPath(wcJob, new Path(pathArgs[i]));
    }
    FileOutputFormat.setOutputPath(wcJob, new Path(pathArgs[pathArgs.length - 1]));
    System.exit(wcJob.waitForCompletion(true) ? 0 : 1);
  }
}
```

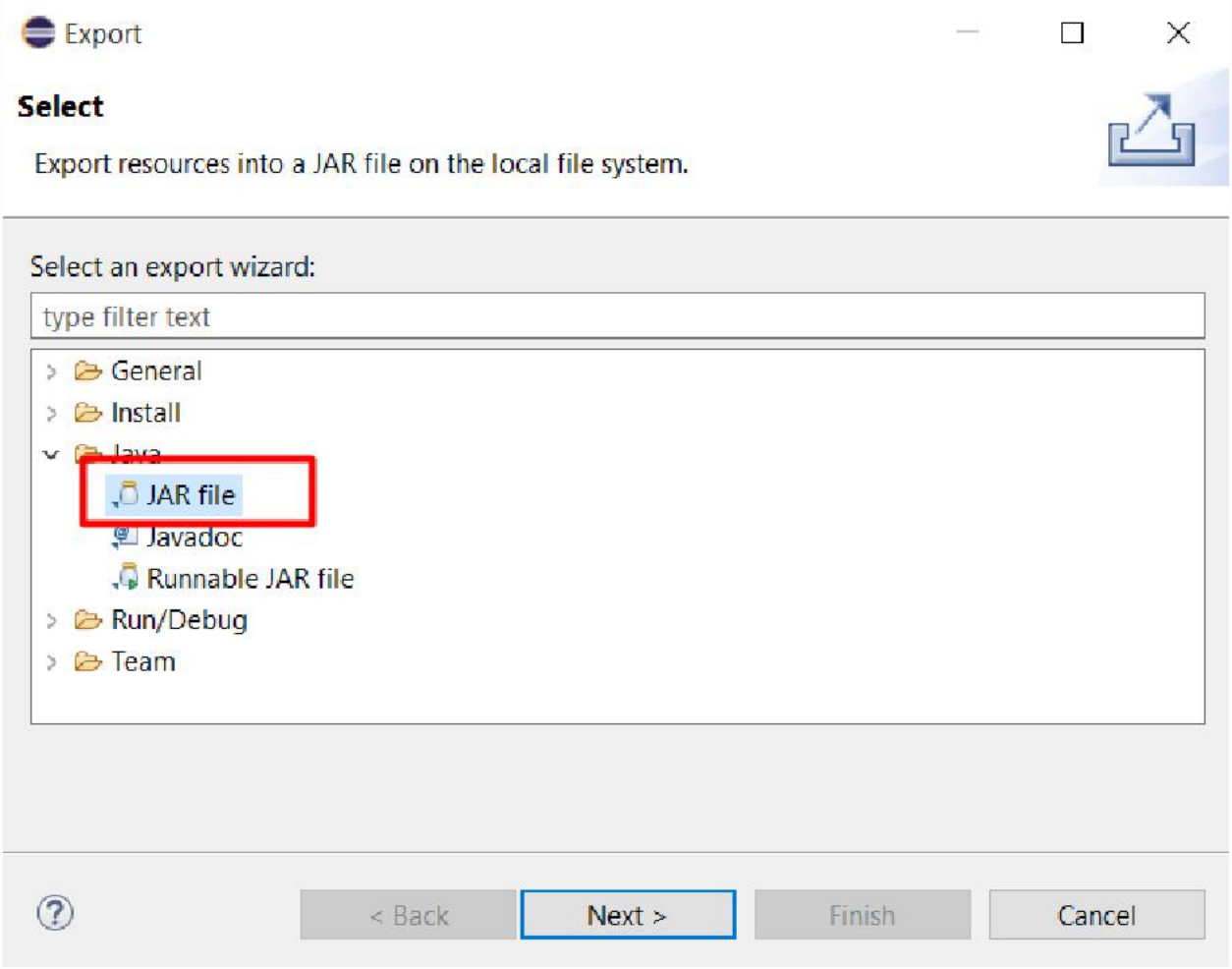



## Bước 4: Tạo file JAR

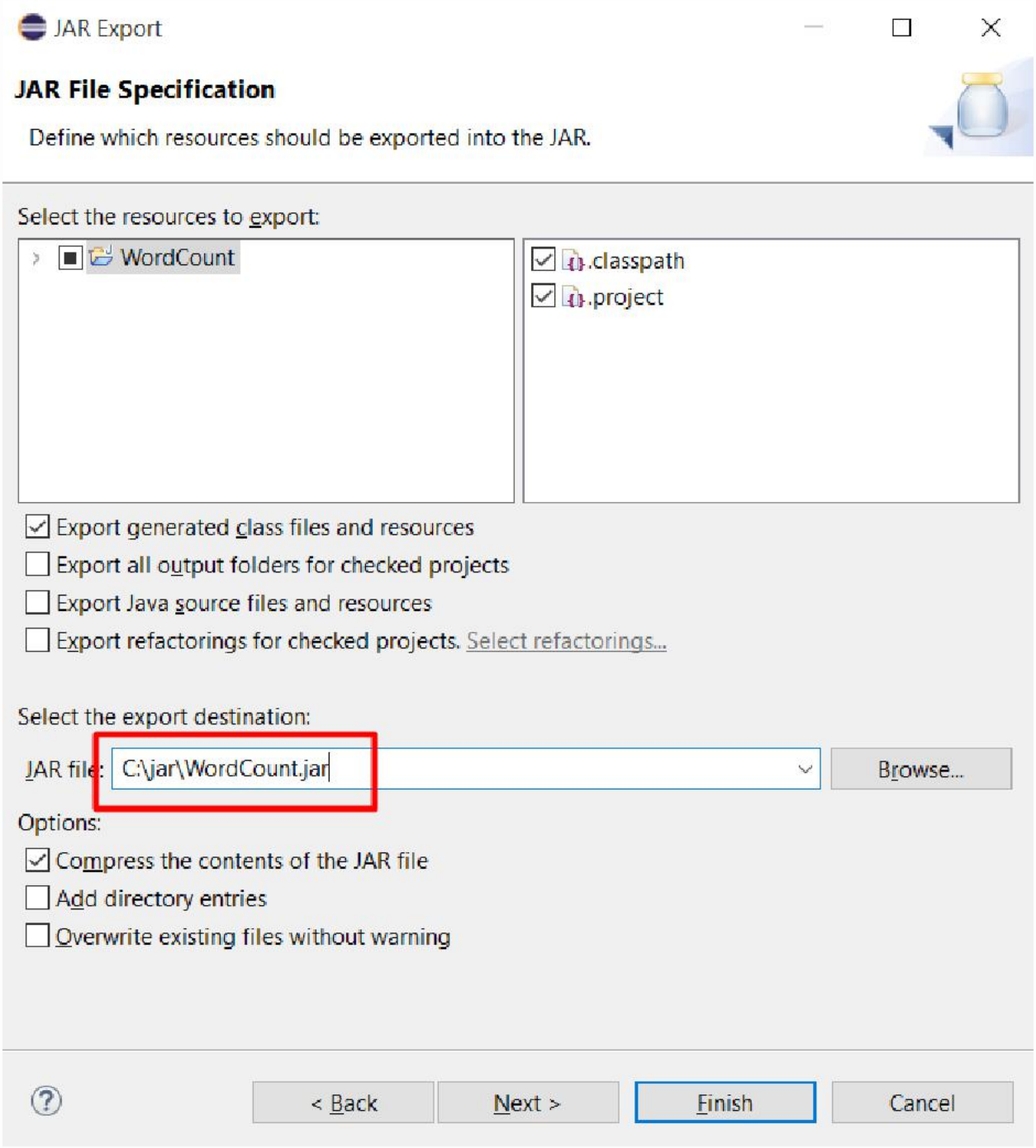
### Chuột phải vào project **WordCount** chọn **Export**

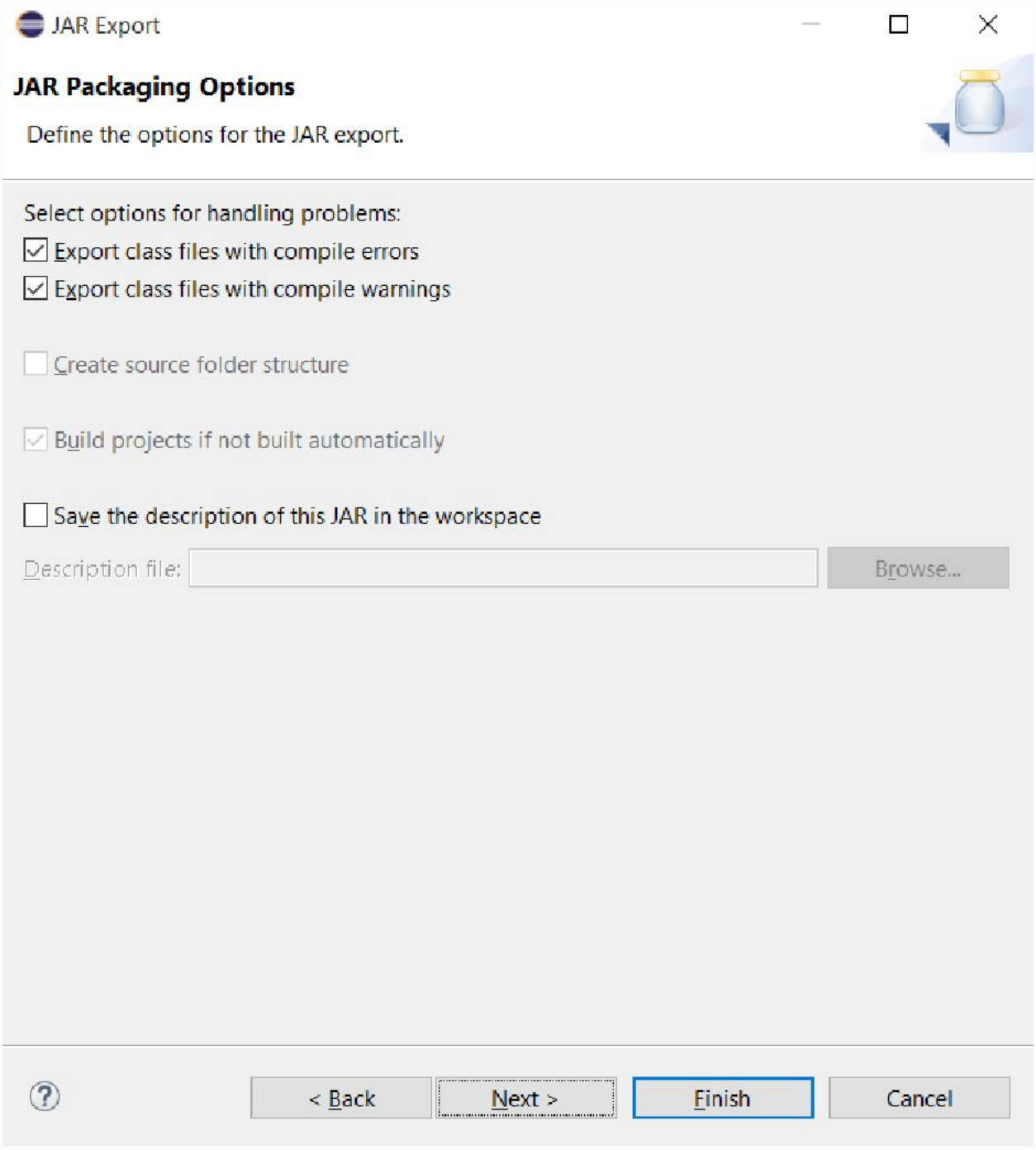



### Chọn **Java** > **JAR File** rồi bấm **Next**

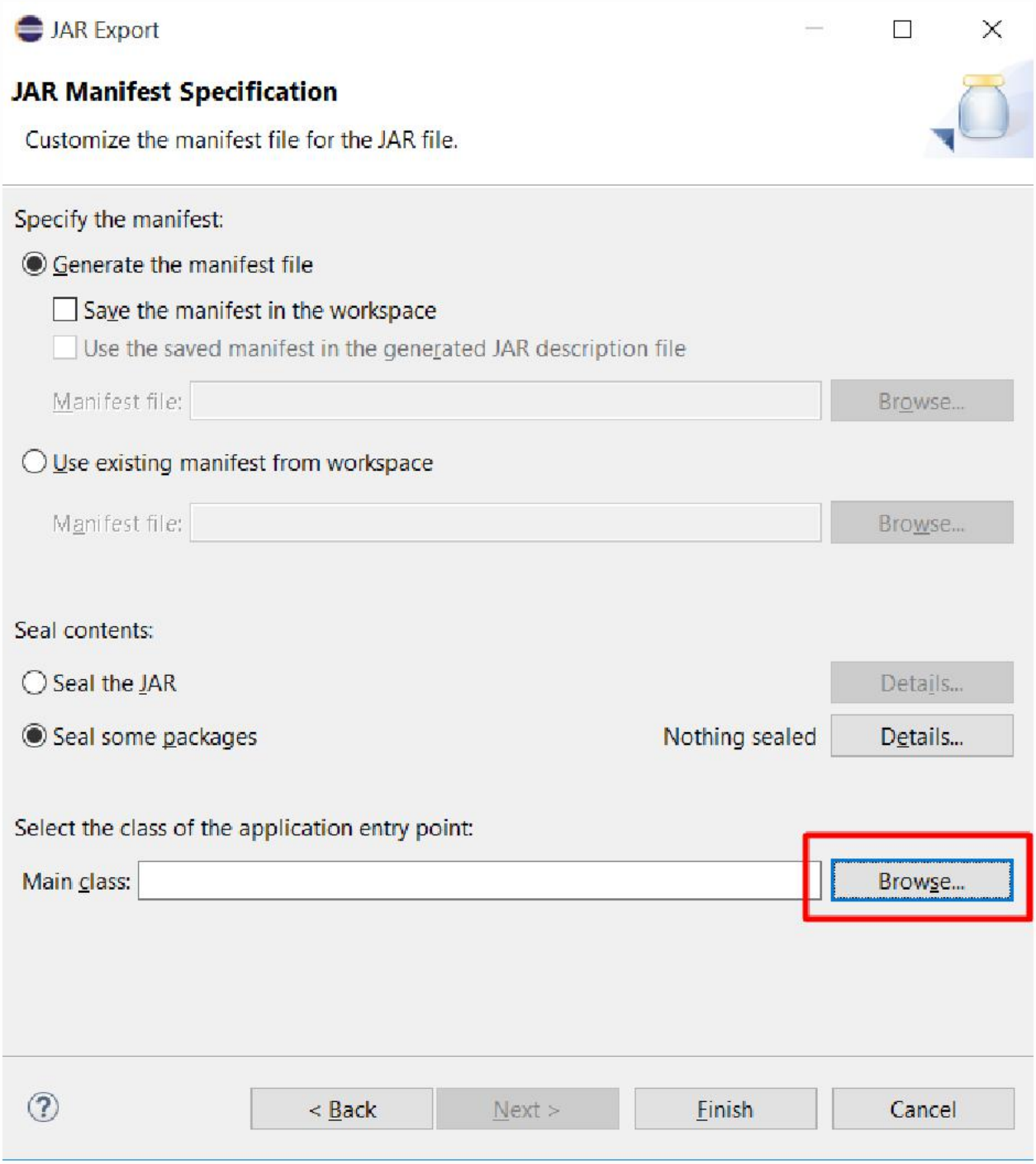



Chọn đường dẫn lưu file JAR và bấm **Next**

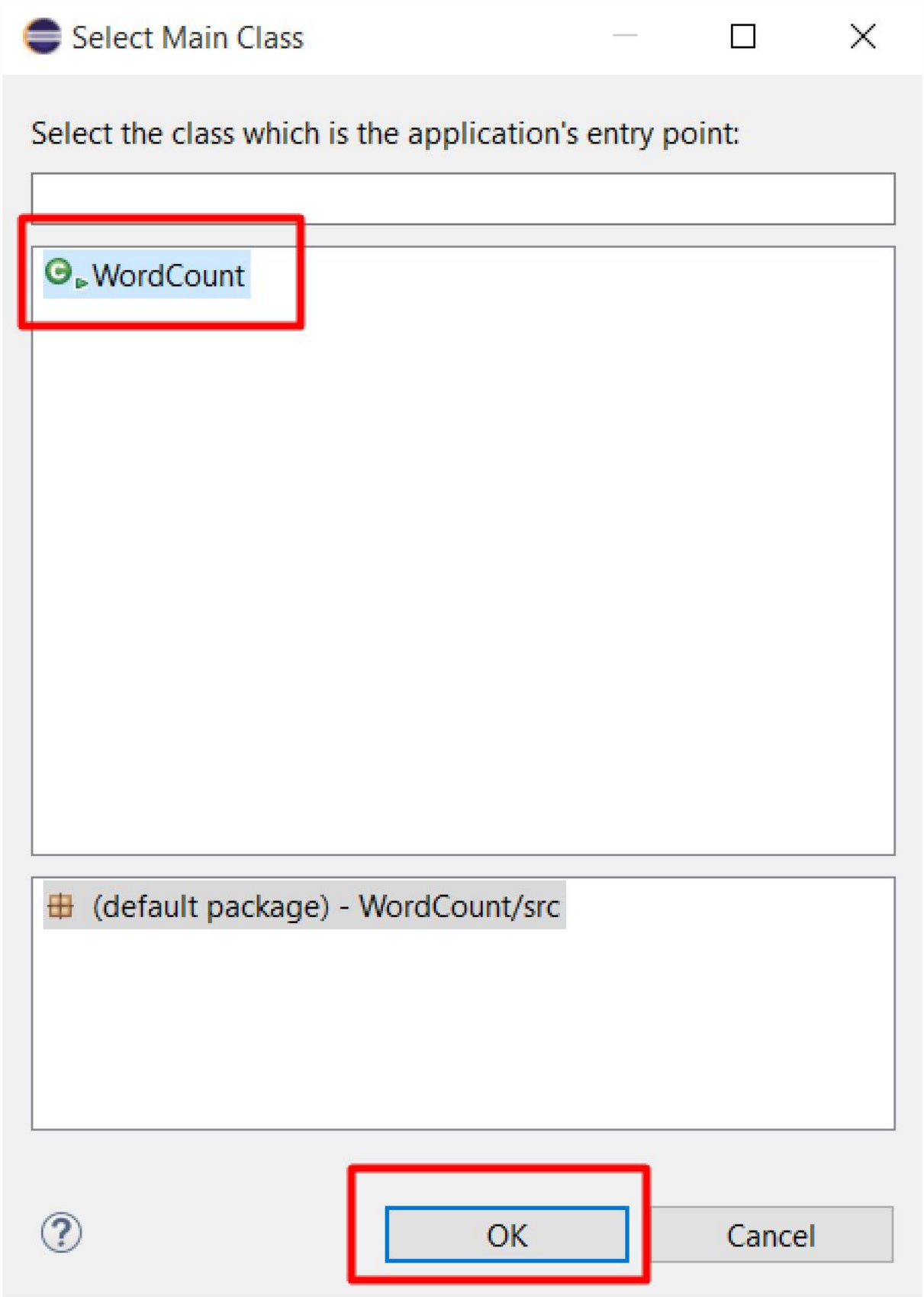

JAR Export
JAR File Specification
Define which resources should be exported into the JAR.
Select the resources to export:
WordCount
.classpath
.project
Export generated class files and resources
Export all output folders for checked projects
Export Java source files and resources
Export refactorings for checked projects. Select refactorings...
Select the export destination:
JAR file: C:\jar\WordCount.jar
Browse...
Options:
Compress the contents of the JAR file
Add directory entries
Overwrite existing files without warning
< Back
Next >
Finish
Cancel



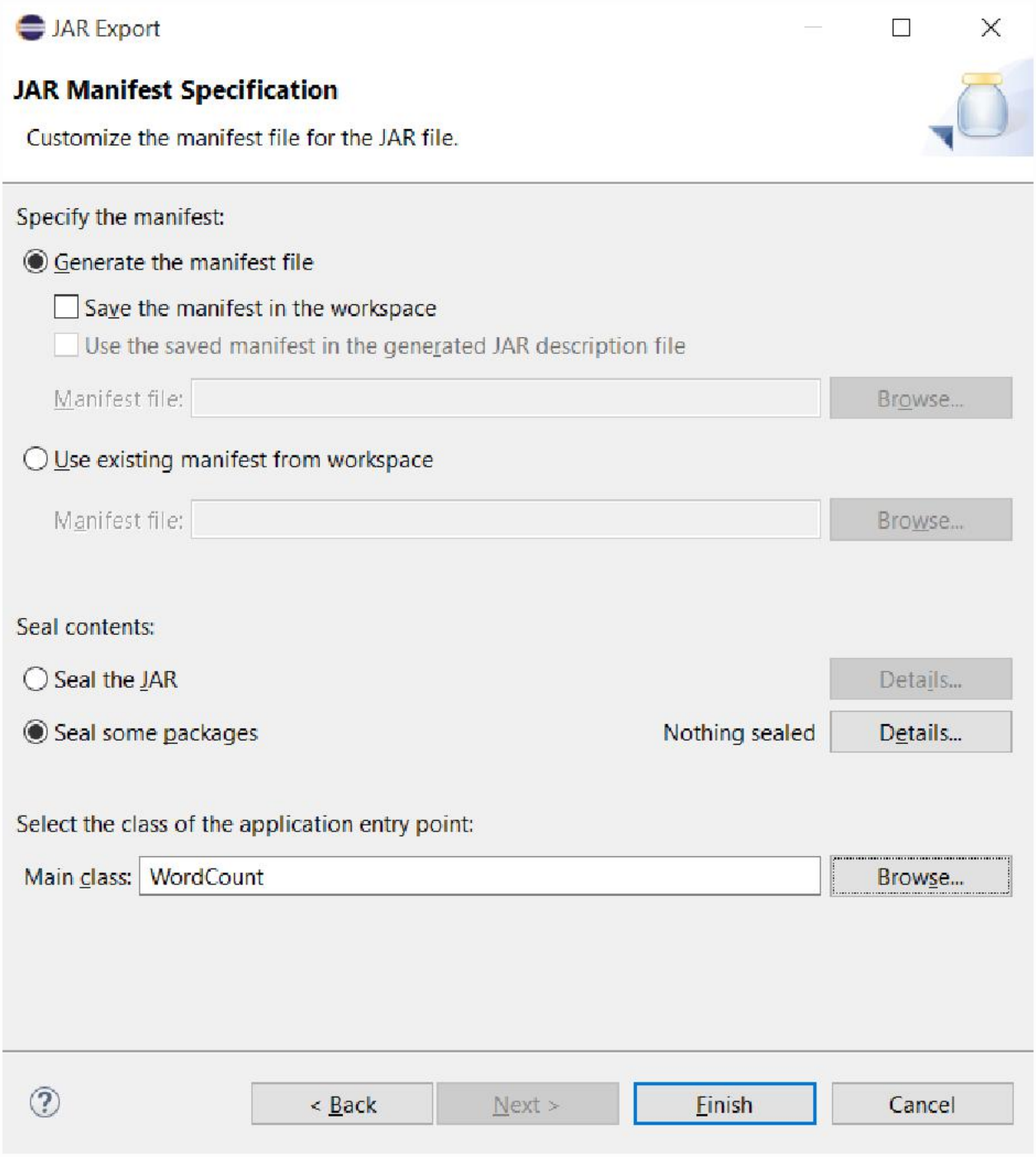

Bấm **Browser** để chọn file **main**



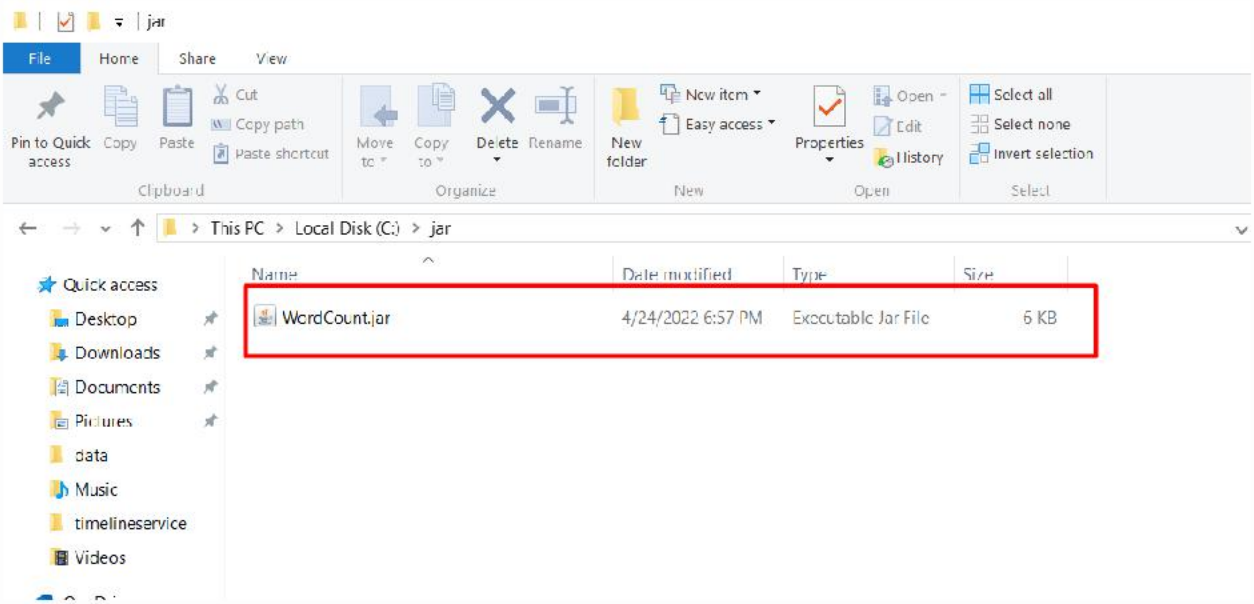

Chọn **WordCount** và bấm **OK**


Select Main Class
Select the class which is the application's entry point:
WordCount
(default package) - WordCount/src
OK
Cancel

Bấm **Finish** để thực hiện quá trình **Export**



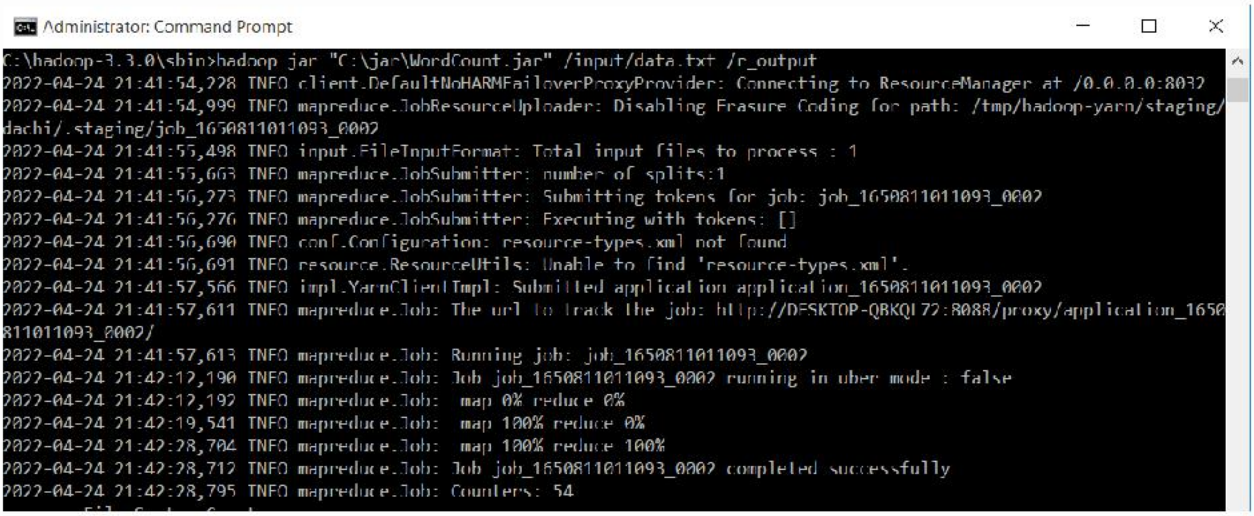

Vào thư mục chứa lưu file JAR vừa tạo và kiểm tra kết quả

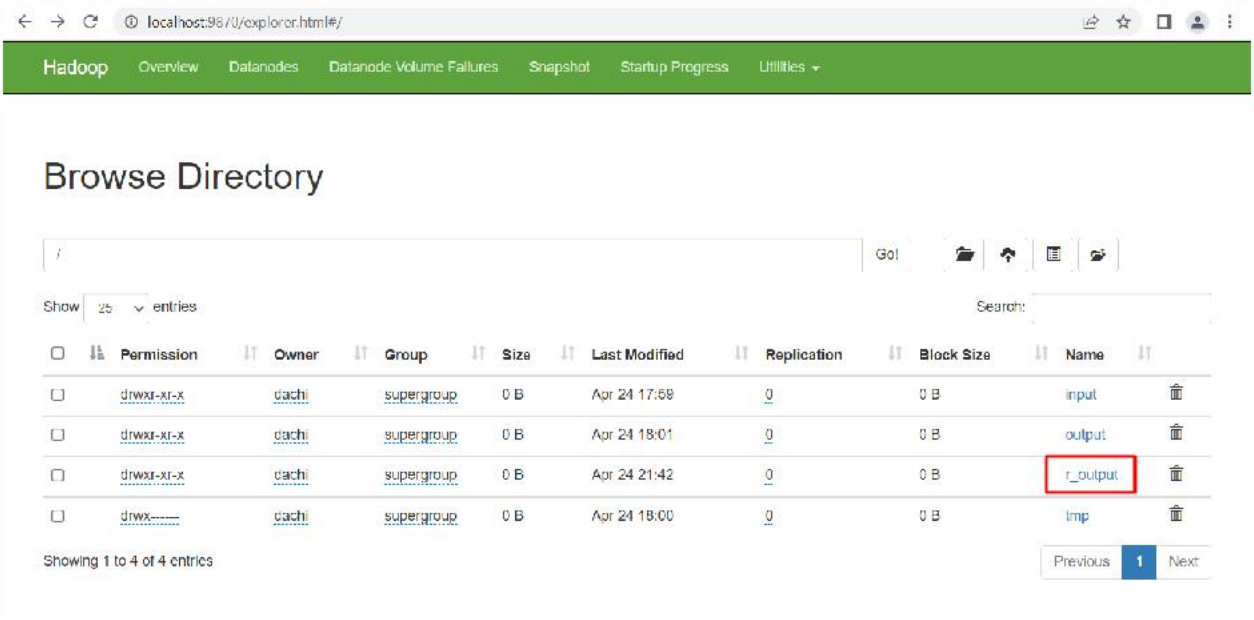


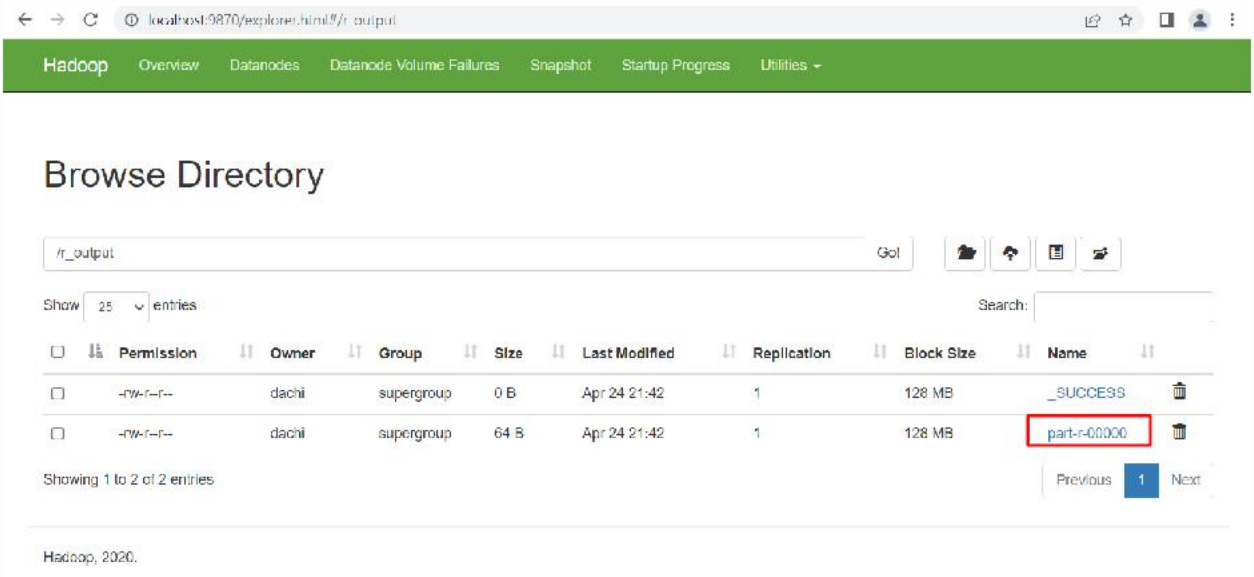

Thử nghiệm trên file dữ liệu **data.txt** đã tạo ở trên, và kết quả thu được lưu tại thư mục **r_output**. Chạy lệnh sau:

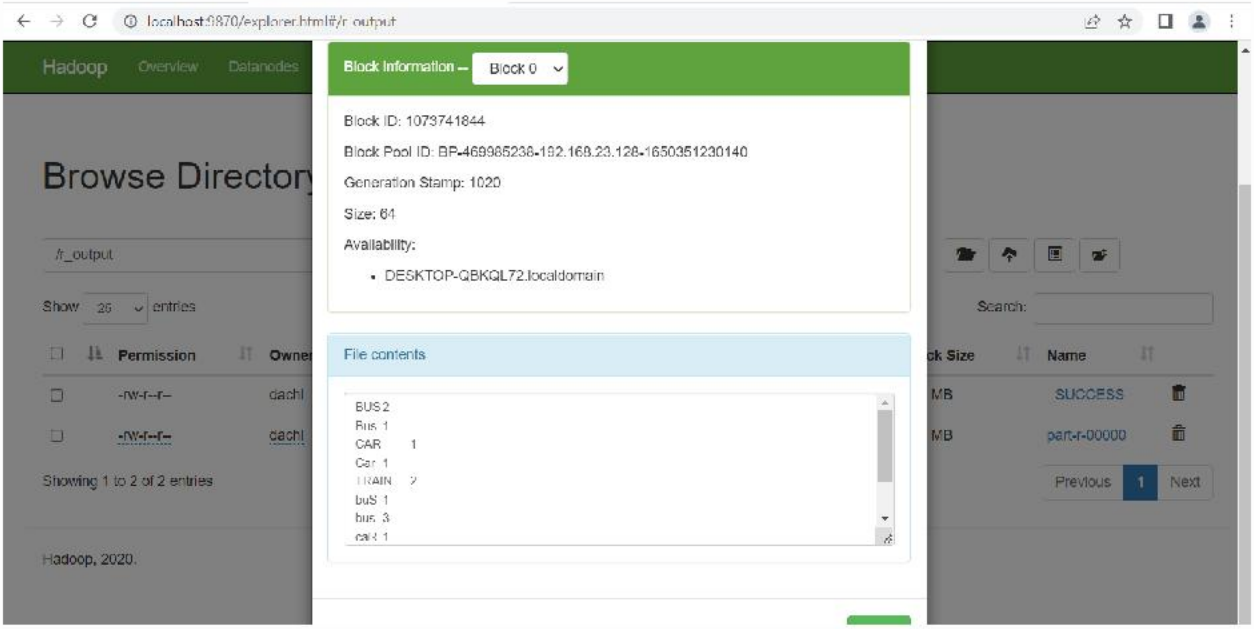
```
hadoop jar "C:\jar\WordCount.jar" /input/data.txt /r_output
```

> *Lưu ý: Thay "C:\jar\WordCount.jar" bằng đường dẫn chứa file JAR ở trên máy*




localhost:9870/explorer.html#/
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Browse Directory
/
Go!
Show 25 entries
Search:
Permission
Owner
Group
Size
Last Modified
Replication
Block Size
Name
drwxr-xr-x dachi supergroup 0 B Apr 24 17:59 0 0 B input
drwxr-xr-x dachi supergroup 0 B Apr 24 18:01 0 0 B output
drwxr-xr-x dachi supergroup 0 B Apr 24 21:42 0 0 B r_output
drwx------ dachi supergroup 0 B Apr 24 18:00 0 0 B tmp
Showing 1 to 4 of 4 entries
Previous
1
Next


localhost:9870/explorer.html#/r_output
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Browse Directory
/r_output
Go!
Show 25 entries
Search:
Permission
Owner
Group
Size
Last Modified
Replication
Block Size
Name
-rw-r--r-- dachi supergroup 0 B Apr 24 21:42 1 128 MB _SUCCESS
-rw-r--r-- dachi supergroup 64 B Apr 24 21:42 1 128 MB part-r-00000
Showing 1 to 2 of 2 entries
Previous
1
Next
Hadoop, 2020.


localhost:9870/explorer.html#/r_output
Hadoop
Overview
Datanodes
Browse Directory
/r_output
Show 25 entries
Permission
-rw-r--r--
-rw-r--r--
Showing 1 to 2 of 2 entries
Hadoop, 2020.
Block Information -- Block 0
Block ID: 1073741844
Block Pool ID: BP-469985238-192.168.23.128-1650351230140
Generation Stamp: 1020
Size: 64
Availability:
DESKTOP-QBKQL72.localdomain
File contents
BUS 2
Bus 1
CAR 1
Car 1
TRAIN 2
buS 1
bus 3
car 1
Search:
Name
SUCCESS
part-r-00000
Previous
1
Next