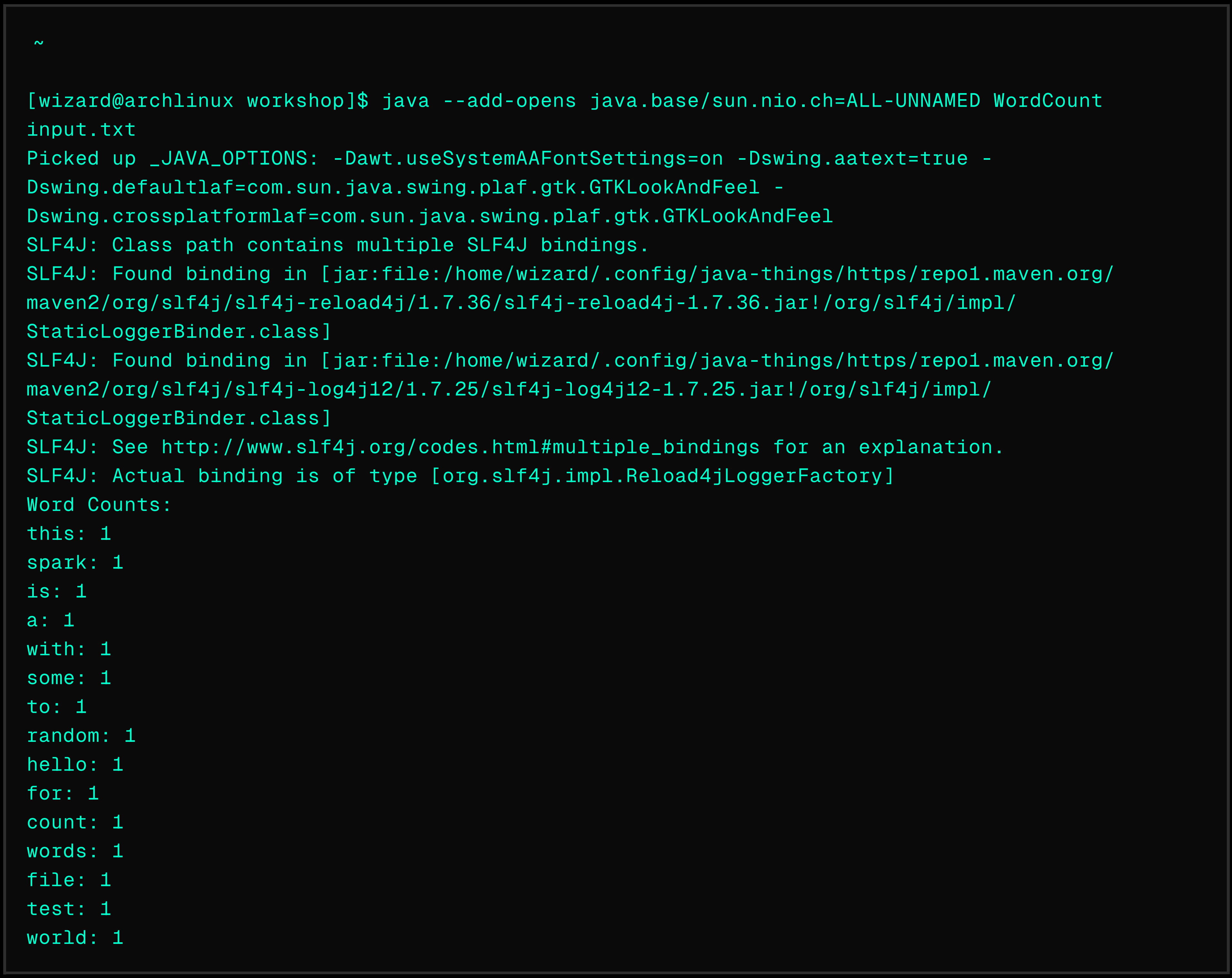
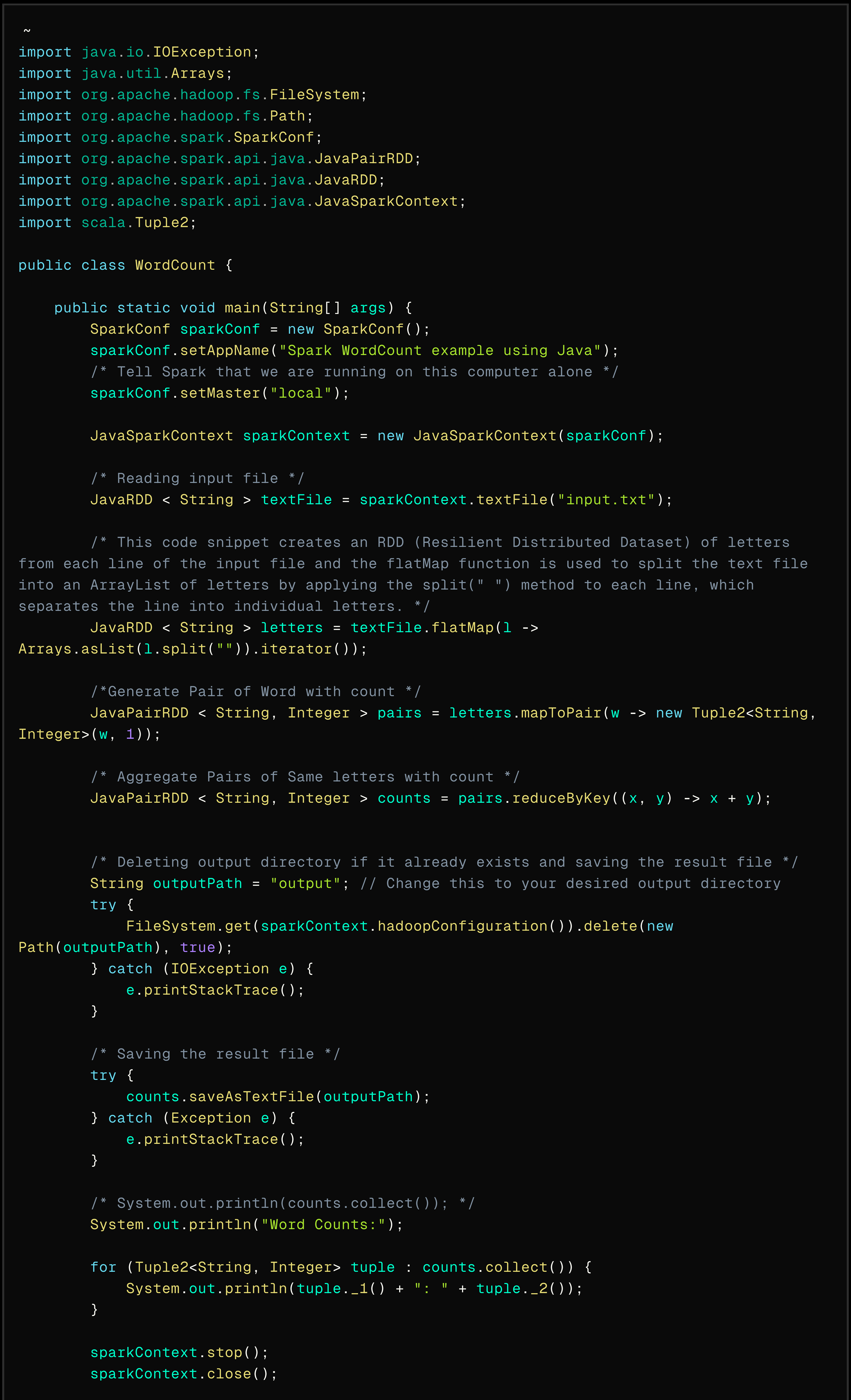
Word Counts:

```
 import java.io.IOException;
import java.util.Arrays;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.spark.SparkConf;
import org.apache.spark.api.java.JavaPairRDD;
import org.apache.spark.api.java.JavaRDD;
import org.apache.spark.api.java.JavaSparkContext;
import scala.Tuple2;

public class WordCount {
    public static void main(String[] args) {
        SparkConf sparkConf = new SparkConf();
        sparkConf.setAppName("Spark WordCount example using Java");
        /* Tell Spark that we are running on this computer alone */
        sparkConf.setMaster("local");
        JavaSparkContext sparkContext = new JavaSparkContext(sparkConf);
        /* Reading input file */
        JavaRDD < String > textFile = sparkContext.textFile("input.txt");
        /* This code snippet creates an RDD (Resilient Distributed Dataset) of words from
each line of the input file and the flatMap function is used to split the text file into
an ArrayList of words by applying the split(" ") method to each line, which separates the
line into individual words. */
        JavaRDD < String > words = textFile.flatMap(l -> Arrays.asList(l.split("
")).iterator());
        /*Generate Pair of Word with count */
        JavaPairRDD < String, Integer > pairs = words.mapToPair(w -> new Tuple2<String,
Integer>(w, 1));
        /* Aggregate Pairs of Same Words with count */
        JavaPairRDD < String, Integer > counts = pairs.reduceByKey((x, y) -> x + y);
        /* Deleting output directory if it already exists and saving the result file */
        String outputPath = "output"; // Change this to your desired output directory
        try {
            FileSystem.get(sparkContext.hadoopConfiguration()).delete(new
Path(outputPath), true);
        } catch (IOException e) {
            e.printStackTrace();
        }
        /* Saving the result file */
        try {
            counts.saveAsTextFile(outputPath);
        } catch (Exception e) {
            e.printStackTrace();
        }
        /* System.out.println(counts.collect()); */
        System.out.println("Word Counts:");

        for (Tuple2<String, Integer> tuple : counts.collect()) {
            System.out.println(tuple._1() + ": " + tuple._2());
        }

        sparkContext.stop();
        sparkContext.close();
    }
}
```



```
 ~

[wizard@archlinux workshop]$ java --add-opens java.base/sun.nio.ch=ALL-UNNAMED WordCount
input.txt
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true -
Dswing.defaultlaf=com.sun.java.swing.plaf.gtk.GTKLookAndFeel -
Dswing.crossplatformlaf=com.sun.java.swing.plaf.gtk.GTKLookAndFeel
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/wizard/.config/java-things/https/repo1.maven.org/
maven2/org/slf4j/slf4j-reload4j/1.7.36/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/
StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/wizard/.config/java-things/https/repo1.maven.org/
maven2/org/slf4j/slf4j-log4j12/1.7.25/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/
StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Reload4jLoggerFactory]
Word Counts:
this: 1
spark: 1
is: 1
a: 1
with: 1
some: 1
to: 1
random: 1
hello: 1
for: 1
count: 1
words: 1
file: 1
test: 1
world: 1
```

Letter Counts:



```
~
import java.io.IOException;
import java.util.Arrays;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.spark.SparkConf;
import org.apache.spark.api.java.JavaPairRDD;
import org.apache.spark.api.java.JavaRDD;
import org.apache.spark.api.java.JavaSparkContext;
import scala.Tuple2;

public class WordCount {

    public static void main(String[] args) {
        SparkConf sparkConf = new SparkConf();
        sparkConf.setAppName("Spark WordCount example using Java");
        /* Tell Spark that we are running on this computer alone */
        sparkConf.setMaster("local");

        JavaSparkContext sparkContext = new JavaSparkContext(sparkConf);

        /* Reading input file */
        JavaRDD < String > textFile = sparkContext.textFile("input.txt");

        /* This code snippet creates an RDD (Resilient Distributed Dataset) of letters
from each line of the input file and the flatMap function is used to split the text file
into an ArrayList of letters by applying the split(" ") method to each line, which
separates the line into individual letters. */
        JavaRDD < String > letters = textFile.flatMap(l ->
Arrays.asList(l.split("")).iterator());

        /*Generate Pair of Word with count */
        JavaPairRDD < String, Integer > pairs = letters.mapToPair(w -> new Tuple2<String,
Integer>(w, 1));

        /* Aggregate Pairs of Same letters with count */
        JavaPairRDD < String, Integer > counts = pairs.reduceByKey((x, y) -> x + y);


        /* Deleting output directory if it already exists and saving the result file */
        String outputPath = "output"; // Change this to your desired output directory
        try {
            FileSystem.get(sparkContext.hadoopConfiguration()).delete(new
Path(outputPath), true);
        } catch (IOException e) {
            e.printStackTrace();
        }

        /* Saving the result file */
        try {
            counts.saveAsTextFile(outputPath);
        } catch (Exception e) {
            e.printStackTrace();
        }

        /* System.out.println(counts.collect()); */
        System.out.println("Word Counts:");

        for (Tuple2<String, Integer> tuple : counts.collect()) {
            System.out.println(tuple._1() + ": " + tuple._2());
        }

        sparkContext.stop();
        sparkContext.close();
        }
}
```

```
 ~

[wizard@archlinux workshop]$ java --add-opens java.base/sun.nio.ch=ALL-UNNAMED WordCount
input.txt
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true -
Dswing.defaultlaf=com.sun.java.swing.plaf.gtk.GTKLookAndFeel -
Dswing.crossplatformlaf=com.sun.java.swing.plaf.gtk.GTKLookAndFeel
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/wizard/.config/java-things/https/repo1.maven.org/
maven2/org/slf4j/slf4j-reload4j/1.7.36/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/
StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/wizard/.config/java-things/https/repo1.maven.org/
maven2/org/slf4j/slf4j-log4j12/1.7.25/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/
StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Reload4jLoggerFactory]
Word Counts:
d: 3
w: 3
s: 6
e: 4
p: 1
a: 3
t: 6
i: 4
k: 1
u: 1
h: 3
 : 14
o: 8
n: 2
f: 2
r: 5
l: 4
m: 2
c: 1
```