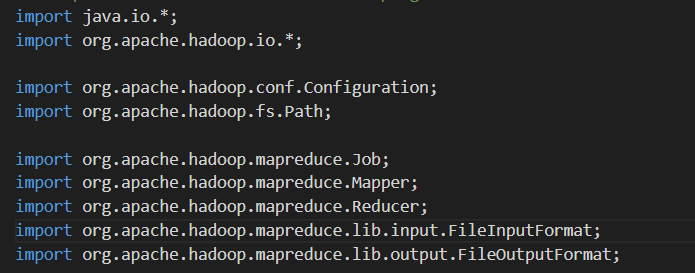
**Custom Writable:**

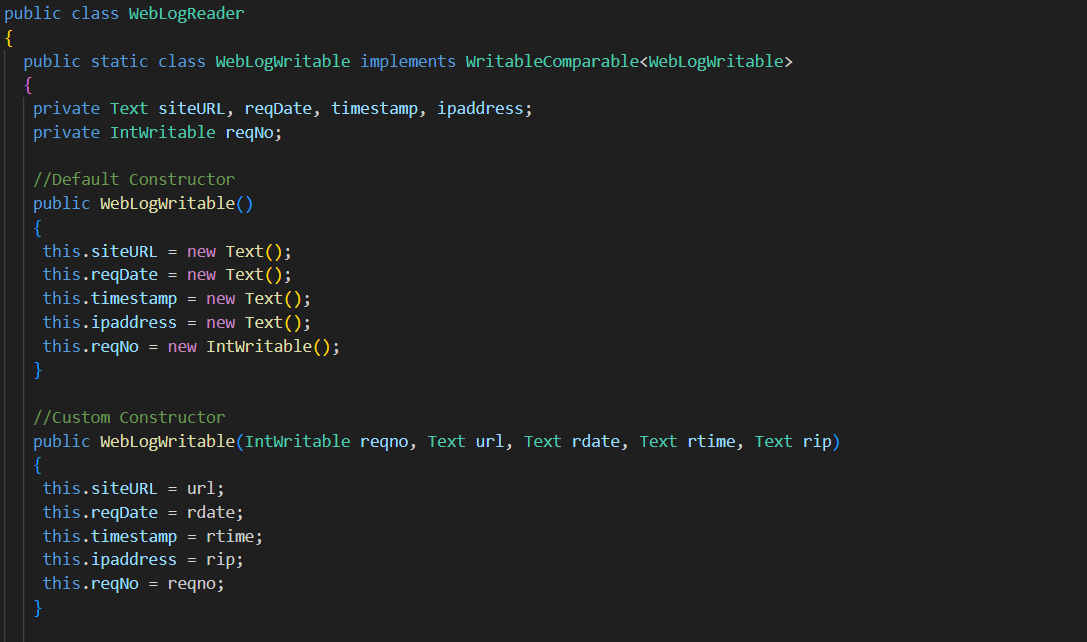
* Objective - take a weblog file and generate counts of IP access using a custom writable

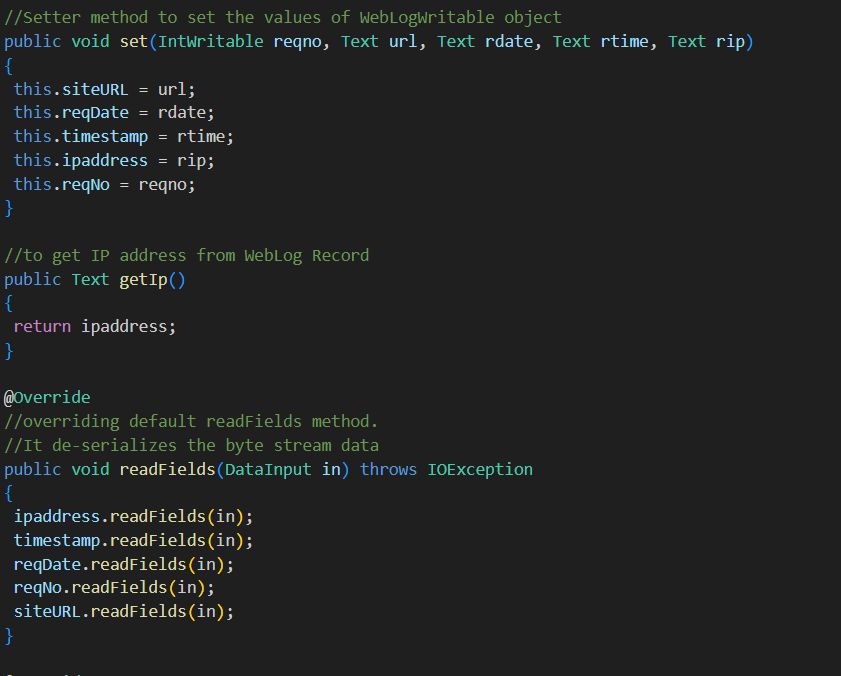
1. rename this file to WebLogReader.java
2. hdfs dfs -put ~/src/weblog.txt /weblog.txt
3. mkdir -m 755 bigram\_classes
4. javac -classpath $(hadoop classpath) -d bigram\_classes WebLogReader.java
5. jar -cvf ${HOME}/scripts/WebLogReader.jar -C bigram\_classes/ .
6. hadoop jar ${HOME}/scripts/WebLogReader.jar WebLogReader /weblog.txt /WebLog.res

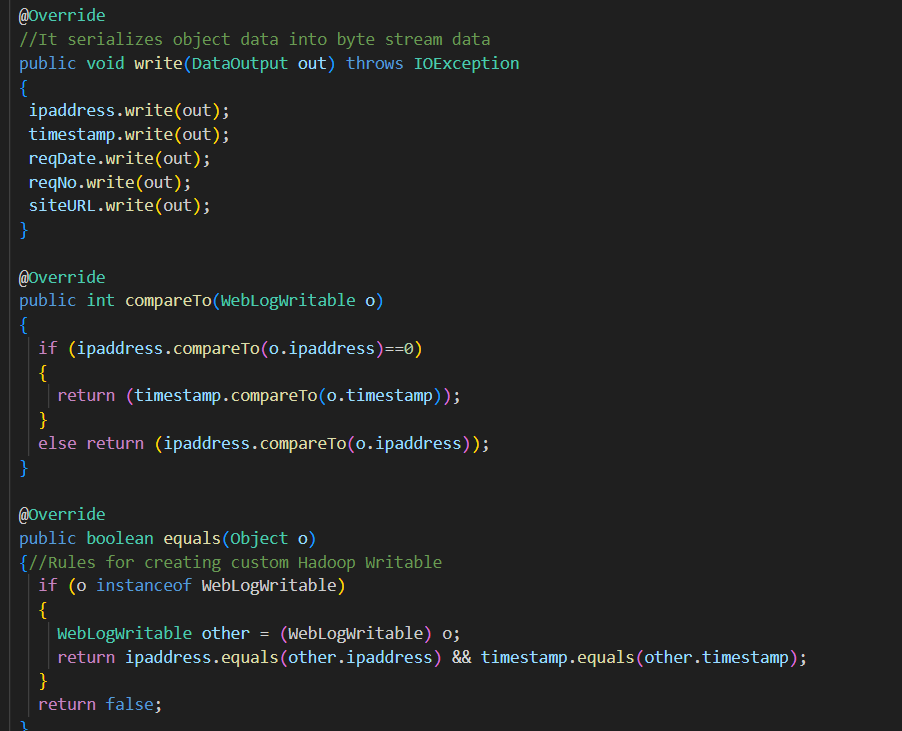
1.save the file as WebLogReader.java



* Includes necessary Hadoop classes for I/O operations, MapReduce framework, and configuration.

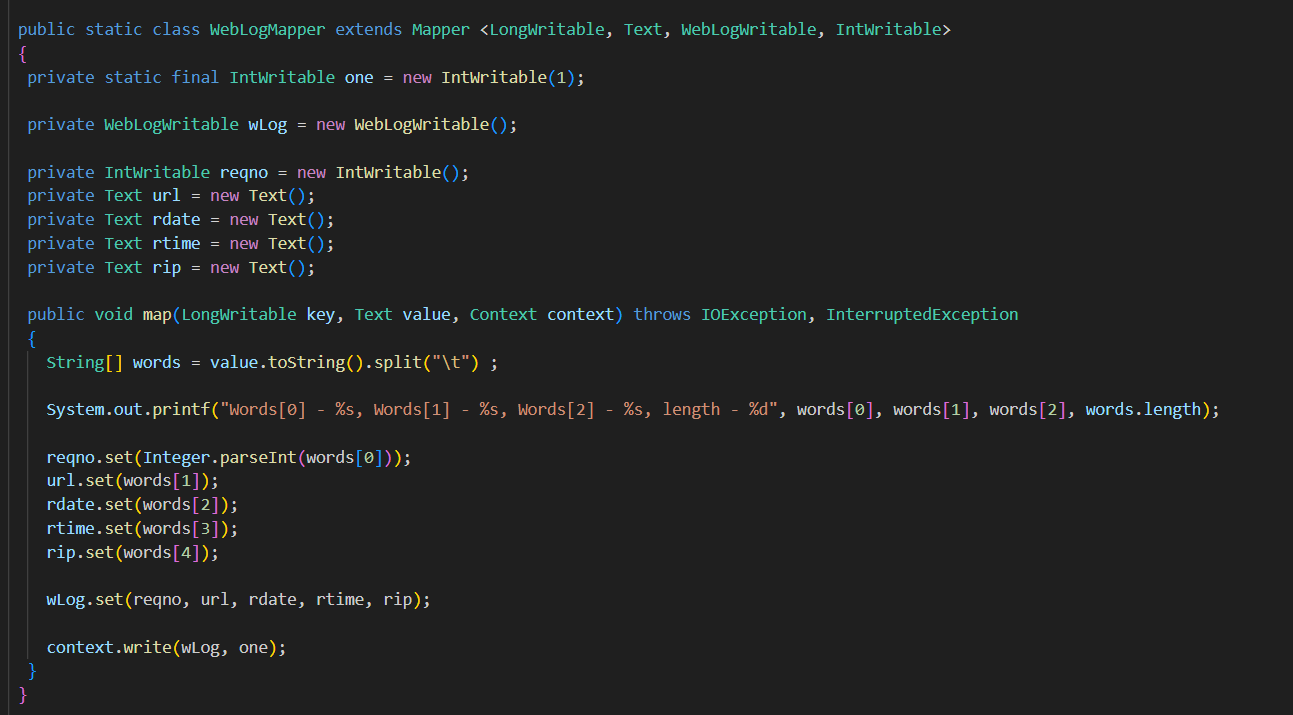




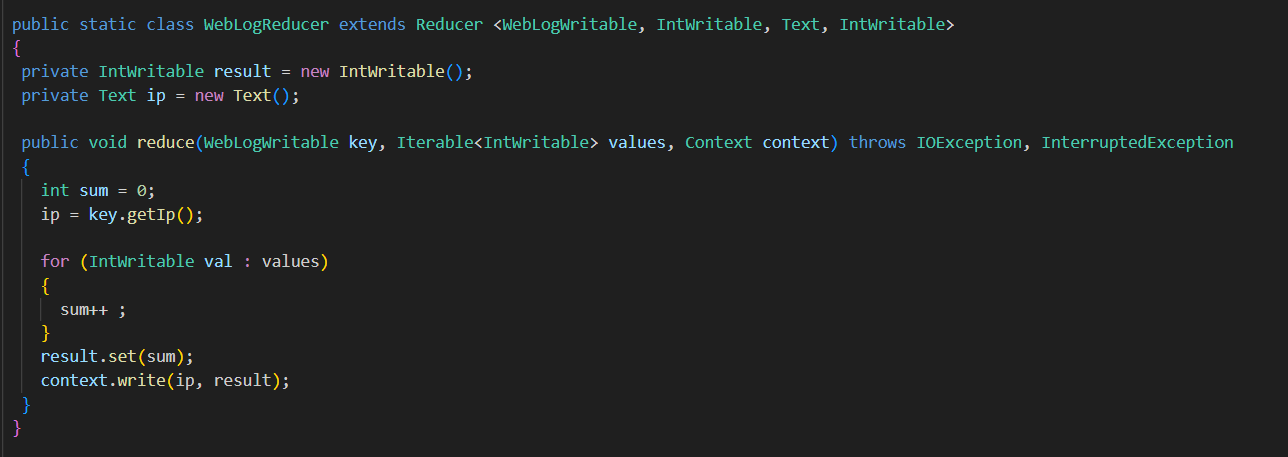




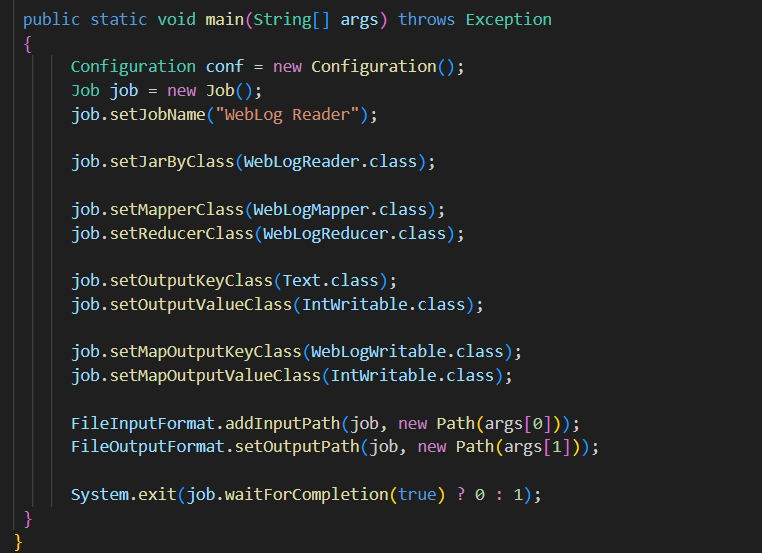
* Defines a custom writable object that encapsulates the data from a weblog record
* Fields: siteURL, reqDate, timestamp, ipaddress, and reqNo.
* readFields and write: Handle serialization and deserialization of the object.
* compareTo: Defines how WebLogWritable objects are compared, which is used for sorting.
* equals and hashCode: Override these methods for object comparison and hashing, though hashCode is not used here.



* Processes each line of the input file, creating a WebLogWritable object for each record and emitting it with a count of 1.



* Aggregates the counts for each IP address.
* reduce Method: Sums up the occurrences of each IP address and writes the final count.



* setJarByClass: Sets the JAR file that contains the job classes.
* setMapperClass and setReducerClass: Specifies the mapper and reducer classes.
* setMapOutputKeyClass and setMapOutputValueClass: Specifies the types for the output of the mapper.
* FileInputFormat.addInputPath and FileOutputFormat.setOutputPath: Sets the input and output paths.
* Implements a MapReduce job with a custom writable class (WebLogWritable) to process weblog data. The mapper processes records and emits them with a count of 1, while the reducer aggregates these counts to provide a count of IP accesses.
* Upload the input data to HDFS, compile and package the Java code, and then run the MapReduce job.