|  |
| --- |
|  |
| **Solution-**   1. **Mapper Class-** Mapper class has got 4 methods- 2. **Setup** 3. **Map** 4. **Run** 5. **Clean up** 6. **Map- Method-** In this method the input is taken from the file line by line and converts in the key value format and sent to the reducer. Map mehtod is called for each input split spawned by the input format fro the job. 7. **Setup Method-**  As name suggests it is the fisrt mehtod executed before any of the other method is executed, this method is used for the setting up initial setups if needed befror the actual mapping starts. e.g finding the cache file and using the that file can be done in the setup pahse of the mapping. 8. **Run Method-** The default run() method simply takes each key / value pair supplied by the context and calls the map() method. Very rarerly this method is over ridden. 9. **Clean up method-** During cleanup() is that you clean up any resources you may have allocated. There are other uses too, which is to flush out any accumulation of aggregate results.  |  | | --- | | Flow | |  | Setup ---> Map -----> Clean up | |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| 1. **Reducer Class-**   As we know the reducer code reads the outputs generated by the different mappers as <key,value> pairs. The reducer interface excepts four generics, which define the types of the input and output key value pairs. The first two paramenters define the intermediate key and value types, the second two deine the final output key and value types.   |  | | --- | |  | |  | | | Reducer has got same Setup , run and clean up method as in that of the mapper class with same functionality.  Reduce methods- The main task of the reducer class is to perform user operations on all the mapper key value pairs sort and shuffle results and to combine these results into one output.   |  | | --- | |  | |  | |  | |  | | | |  | | |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |