Problem:

Write a secondary sort program to generate the top 2 maximum temperatures corresponding to every year from the temperature dataset.

Codes:

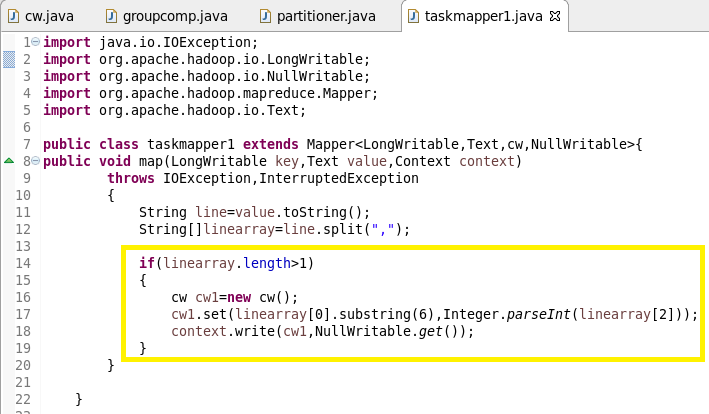
Task class:



Composite key will consider both year and temperature as key.

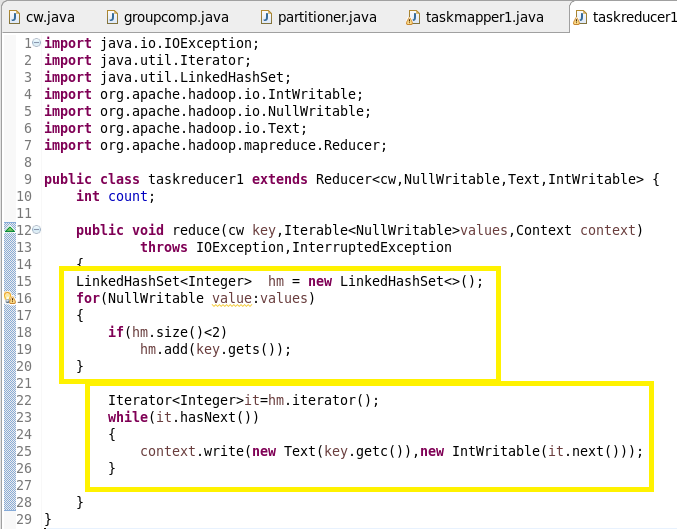
GroupingComparator class will help reducer to group all the keys coming to it based on natural key instead of composite key.

Mapper class:



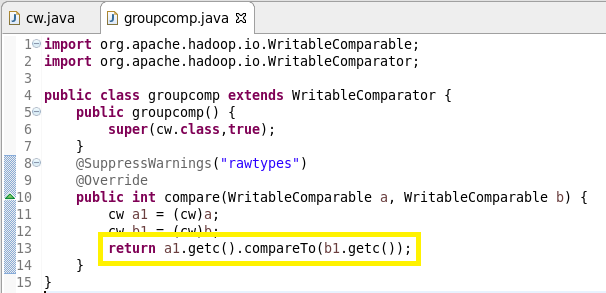
Mapper will convert the data in key and value pair where key is composite key (combination of year and temperature where year is the natural key) and value is nullwritable (secondary sort).

Reducer class:



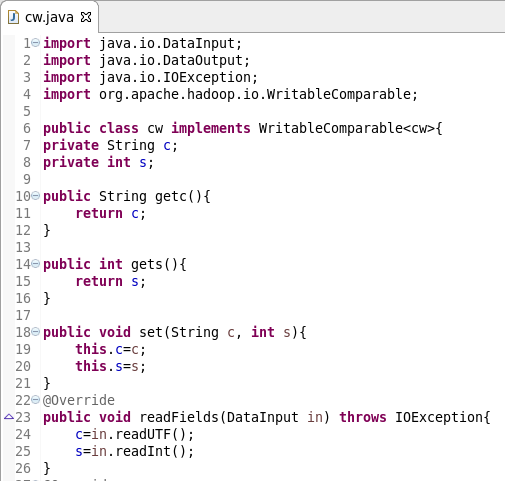
In reducer we are finding out the two maximum temperatures corresponding to each year.

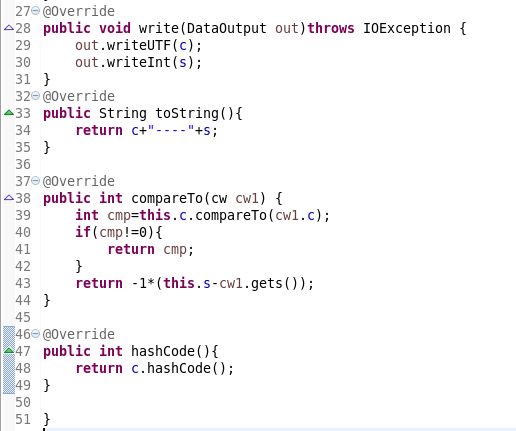
Groupcomparator class:



Once the data reaches a reducer, all data is grouped by key. Since we have a composite key, we need to make sure records are grouped solely by the natural key. This is accomplished by writing a custom Groupcomparator. We have a Comparator object only considering the year field (i.e. c) of the cw class for the purposes of grouping the records together.

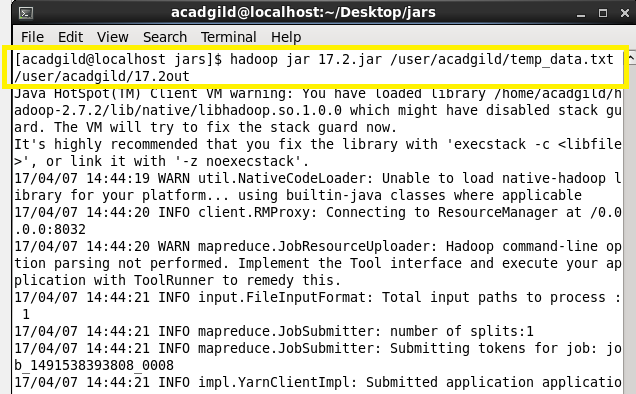
Writable comparable class:





Writable comparable will help us to create composite key and arrange them in descending order based on temperature which is the value. Thus we achieved secondary sort i.e., sorting based on values.

Running jar:



Output:

