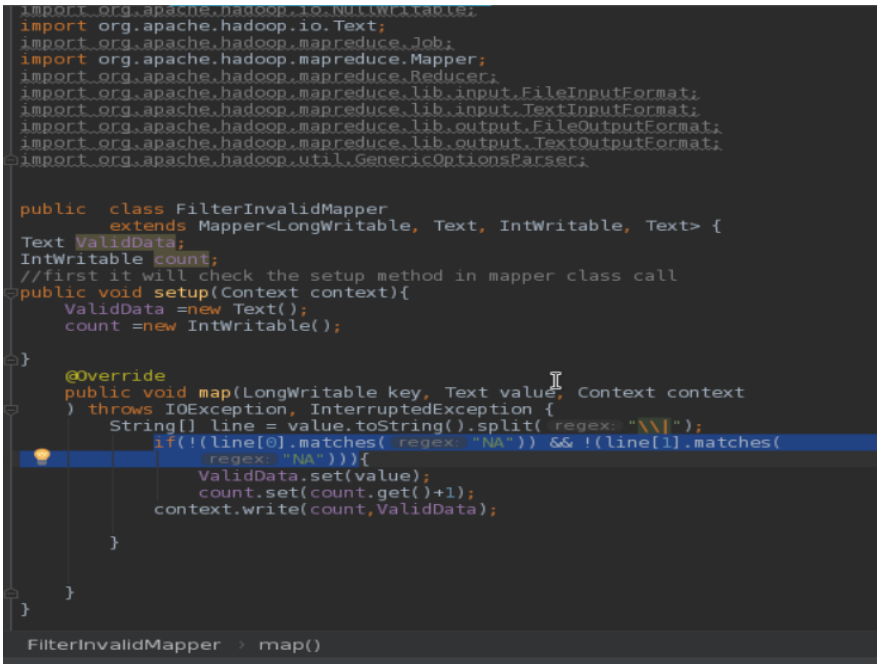
Solutions Assignment 4

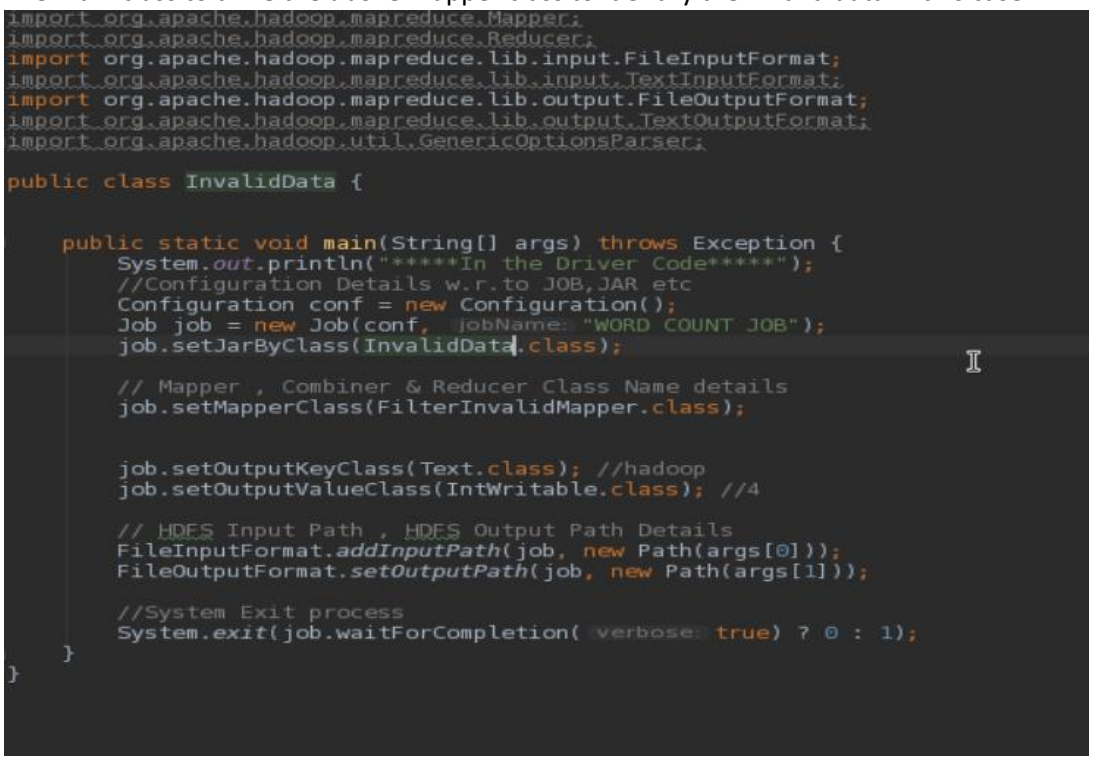
Q: Write a Map Reduce program to filter out the invalid records. Map only job will fit for this context.?

Create Map Program in Intellij IDEA

- The project package used for this task is default. This task is done by Mapper class alone,



- The main class to drive the above mapper class to identify the invalid data in this case “NA”.



- Execute the jar

```
n /var/spool/mail/acadgild
 yarn]$ hadoop jar DO-IT.jar InvalidData /hadoop/television.txt /hadoop/invalidout
```

- To view the out put file.

Input	Output
Samsung|Optima|14|Madhya Pradesh|132401|14200 Onida|Lucid|18|Uttar Pradesh|232401|16200 Akai|Decent|16|Kerala|922401|12200 Lava|Attention|20|Assam|454601|24200 Zen|Super|14|Maharashtra|619082|9200 Samsung|Optima|14|Madhya Pradesh|132401|14200 Onida|Lucid|18|Uttar Pradesh|232401|16200 Onida|Decent|14|Uttar Pradesh|232401|16200 Onida|NA|16|Kerala|922401|12200 Lava|Attention|20|Assam|454601|24200 Zen|Super|14|Maharashtra|619082|9200 Samsung|Optima|14|Madhya Pradesh|132401|14200 NA|Lucid|18|Uttar Pradesh|232401|16200 Samsung|Decent|16|Kerala|922401|12200 Lava|Attention|20|Assam|454601|24200 Samsung|Super|14|Maharashtra|619082|9200 Samsung|Super|14|Maharashtra|619082|9200 Samsung|Super|14|Maharashtra|619082|9200	1 Samsung|Optima|14|Madhya Pradesh|132401|14200 2 Onida|Lucid|18|Uttar Pradesh|232401|16200 3 Akai|Decent|16|Kerala|922401|12200 4 Lava|Attention|20|Assam|454601|24200 5 Zen|Super|14|Maharashtra|619082|9200 6 Samsung|Optima|14|Madhya Pradesh|132401|14200 7 Onida|Lucid|18|Uttar Pradesh|232401|16200 8 Onida|Decent|14|Uttar Pradesh|232401|16200 9 Lava|Attention|20|Assam|454601|24200 10 Zen|Super|14|Maharashtra|619082|9200 11 Samsung|Optima|14|Madhya Pradesh|132401|14200 12 Samsung|Decent|16|Kerala|922401|12200 13 Lava|Attention|20|Assam|454601|24200 14 Samsung|Super|14|Maharashtra|619082|9200 15 Samsung|Super|14|Maharashtra|619082|9200 16 Samsung|Super|14|Maharashtra|619082|9200

2. Write a Map Reduce program to calculate the total units sold for each Company

- Mapper class for each company details and for total units input.

```
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;


public  class productmapper
        extends Mapper<LongWritable, Text, Text, IntWritable> {
    Text company;
    IntWritable count;
    //first it will check the setup method in mapper class call
    public void setup(Context context){
        company =new Text();
        count =new IntWritable( value: 1);

    }
    @Override
    public void map(LongWritable key, Text value, Context context
    ) throws IOException, InterruptedException {
        String[] line = value.toString().split( regex: "\\|");
        if(!(line[0].matches( regex: "NA")) && !(line[1].matches(
                regex: "NA"))){
            company.set(line[0]);
            context.write(company,count);

        }


    }
}
```

- Reducer class or total units and for each company

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

import java.io.IOException;

public class productreducer extends Reducer<Text,IntWritable,Text,IntWritable> {
   public  void reduce(Text key,Iterable<IntWritable> values,Context context) throws IOException,InterruptedException {
        int sum = 0;
        for (IntWritable value: values){
            sum+=value.get();


        }
context.write(key,new IntWritable(sum));
    }
}
```

- Driver class program

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class Company {
    public static void main(String[] args) throws Exception {
        System.out.println("*****In the Driver Code*****");
        //Configuration Details w.r.to JOB,JAR etc
        Configuration conf = new Configuration();
        Job job = new Job(conf,  jobName: "per each company JOB");
        job.setJarByClass(Invalid1.class);
        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(IntWritable.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        job.setMapperClass(productmapper.class);
        job.setReducerClass(productreducer.class);
        job.setCombinerClass(productreducer.class);
        job.setNumReduceTasks(1);//set the reduce tasks to one
        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion( verbose: true) ? 0 : 1);
    }
}
```

- Jar file run on the Hadoop to get the each company wise total units.

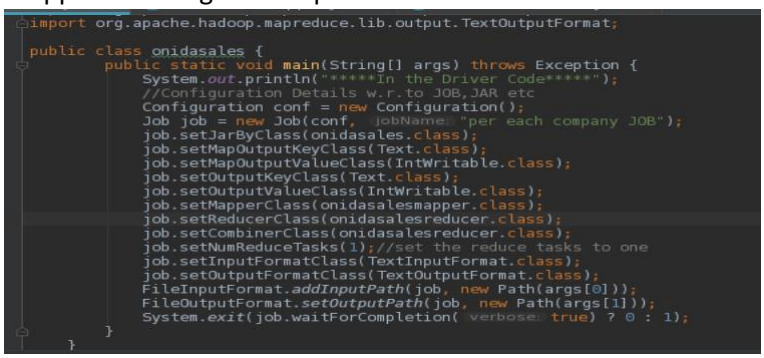
```
[acadgild@localhost yarn]$ hadoop jar DO-IT.jar Company /hadoop/television.txt /hadoop/companyout
```

Output

```
[acadgild@localhost yarn]$ hadoop fs -cat  /hadoop/companyout/part-r-00000
18/05/19 23:54:24 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform
asses where applicable
Akai    1
Lava    3
Onida   3
Samsung 7
Zen     2
```

3. Write a Map Reduce program to calculate the total units sold in each state for Onida company.

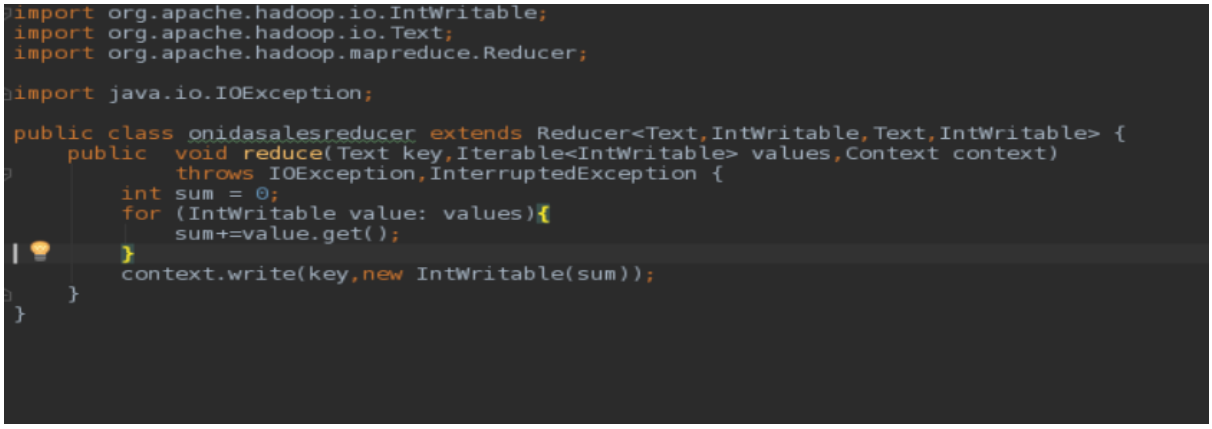
- Mapper class to get the input to the reducer class.



```
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class onidasales {
        public static void main(String[] args) throws Exception {
            System.out.println("*****In the Driver Code*****");
            //Configuration Details w.r.to JOB,JAR etc
            Configuration conf = new Configuration();
            Job job = new Job(conf,  jobName: "per each company JOB");
            job.setJarByClass(onidasales.class);
            job.setMapOutputKeyClass(Text.class);
            job.setMapOutputValueClass(IntWritable.class);
            job.setOutputKeyClass(Text.class);
            job.setOutputValueClass(IntWritable.class);
            job.setMapperClass(onidasalesmapper.class);
            job.setReducerClass(onidasalesreducer.class);
            job.setCombinerClass(onidasalesreducer.class);
            job.setNumReduceTasks(1);//set the reduce tasks to one
            job.setInputFormatClass(TextInputFormat.class);
            job.setOutputFormatClass(TextOutputFormat.class);
            FileInputFormat.addInputPath(job, new Path(args[0]));
            FileOutputFormat.setOutputPath(job, new Path(args[1]));
            System.exit(job.waitForCompletion( verbose: true) ? 0 : 1);
        }
    }
```

- Reducer class to get the total units sold for each onida use the Mapper class output.



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

import java.io.IOException;

public class onidasalesreducer extends Reducer<Text,IntWritable,Text,IntWritable> {
    public  void reduce(Text key,Iterable<IntWritable> values,Context context)
            throws IOException,InterruptedException {
        int sum = 0;
        for (IntWritable value: values){
            sum+=value.get();
        }
        context.write(key,new IntWritable(sum));
    }
}
```

- Driver class to start the class and placed the combiner class(mini reducer).

```
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class onidasales {
        public static void main(String[] args) throws Exception {
            System.out.println("*****In the Driver Code*****");
            //Configuration Details w.r.to JOB,JAR etc
            Configuration conf = new Configuration();
            Job job = new Job(conf,  jobName: "per each company JOB");
            job.setJarByClass(onidasales.class);
            job.setMapOutputKeyClass(Text.class);
            job.setMapOutputValueClass(IntWritable.class);
            job.setOutputKeyClass(Text.class);
            job.setOutputValueClass(IntWritable.class);
            job.setMapperClass(onidasalesmapper.class);
            job.setReducerClass(onidasalesreducer.class);
            job.setCombinerClass(onidasalesreducer.class);
            job.setNumReduceTasks(1);//set the reduce tasks to one
            job.setInputFormatClass(TextInputFormat.class);
            job.setOutputFormatClass(TextOutputFormat.class);
            FileInputFormat.addInputPath(job, new Path(args[0]));
            FileOutputFormat.setOutputPath(job, new Path(args[1]));
            System.exit(job.waitForCompletion( verbose: true) ? 0 : 1);
        }
    }
```

- Jar file to execute the get the required output.

```
[acadgild@localhost yarn]$ hadoop jar DO-IT.jar /hadoop/television.txt /hadoop/salesout
```

- OutPut file from hdfs.

Uttar Pradesh 3