

```

1  import java.io.IOException;
2  import java.util.StringTokenizer;
3
4  import org.apache.hadoop.conf.Configuration;
5  import org.apache.hadoop.fs.Path;
6  import org.apache.hadoop.io.IntWritable;
7  import org.apache.hadoop.io.Text;
8  import org.apache.hadoop.mapreduce.Job;
9  import org.apache.hadoop.mapreduce.Mapper;
10 import org.apache.hadoop.mapreduce.Reducer;
11 import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
12 import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
13
14 public class solution4 {
15
16     public static void main(String[] args) throws Exception {
17         Configuration conf = new Configuration();
18         Job job = Job.getInstance(conf, "Total deposit");
19         job.setJarByClass(solution4.class);
20         job.setMapperClass(TokenizerMapper.class);
21         job.setCombinerClass(solution4Reducer.class);
22         job.setReducerClass(solution4Reducer.class);
23         job.setOutputKeyClass(Text.class);
24         job.setOutputValueClass(IntWritable.class);
25         FileInputFormat.addInputPath(job, new Path(args[0]));
26         FileOutputFormat.setOutputPath(job, new Path(args[1]));
27         System.exit(job.waitForCompletion(true) ? 0 : 1);
28     }
29
30     public static class TokenizerMapper
31         extends Mapper<Object, Text, Text, IntWritable>{
32
33         private Text account = new Text();
34         private Text date = new Text();
35         private final static IntWritable amount = new IntWritable(0);
36         private Text year = new Text();
37         private Text account_year = new Text();
38         private Integer int_amount;
39
40         public void map(Object key, Text value, Context context
41             ) throws IOException, InterruptedException {
42             StringTokenizer itr = new StringTokenizer(value.toString());
43             while (itr.hasMoreTokens()) {
44                 account.set(itr.nextToken());
45                 date.set(itr.nextToken());
46                 String string_account = account.toString();
47                 String string_year = date.toString().substring(7);
48
49                 account_year.set(string_account+string_year);
50                 int_amount = Integer.parseInt(itr.nextToken());
51
52                 if (int_amount > 0) {
53                     amount.set(int_amount);
54                     context.write(account_year, amount);
55                 }
56             }
57         }
58     }
59
60     public static class solution4Reducer
61         extends Reducer<Text,IntWritable,Text,IntWritable> {
62         private IntWritable result = new IntWritable();
63
64         public void reduce(Text key, Iterable<IntWritable> values,
65             Context context
66             ) throws IOException, InterruptedException {
67             int total = 0;
68
69             for (IntWritable val : values)

```

```
70         total = total + val.get();
71
72         result.set(total);
73         context.write(key, result);
74     }
75 }
76 }
```