实验目的：

1）理解 mapreduce 执行原理
2）理解 map,reduce 阶段
3）熟悉统计的原理
4）熟悉去重原理
5）熟悉平均函数的使用
6）了解排序的原理
7）二次排序的原理
8）了解 mapreduce 处理日志的原理

实验环境：

本次环境是：centos7.5 + jdk1.8.0_79 + hadoop2.4.1 + eclipse
日志文件 source.txt 存放在/root/simple/目录下

实验技术原理：

1. MR1-1 词频统计

首先对待处理的信息进行拆分，拆分之后在 map 阶段，把拆分的每个单词作为 map 方法的输出键，而 map 的方法输出的值设置为 1，最后在 reduce 阶段对每个键的值集合进行遍历并把遍历的值进行相加，输出结果即可。

2.MR1-2 单词个数

首先对待处理的信息进行拆分，拆分之后在 map 阶段，拆分后计算出单词个数并作为 map 方法的输出值，而 map 的方法输出键作为 NullWritable 即可，最后在 reduce 阶段对每个键的值集合进行遍历并把遍历的值进行相加，输出结果即可。

3.MR2-2 成绩统计

【案例 3.1】 mapreduce 学生总成绩报表

【案例 3.2】mapreduce 学生各科平均成绩报表

Map 处理的是一个纯文本文件，文件中存放的数据时每一行表示一个学生的姓名和他相应一科成绩。Mapper 处理的数据是由 InputFormat 分解过的数据集，其中 InputFormat 的作用是将数据集切割成小数据集 InputSplit，每一个 InputSplit 将由一个 Mapper 负责处理。此外，InputFormat 中还提供了一个 RecordReader 的实现，并将一个 InputSplit 解析成<key,value>对提供给了 map 函数。InputFormat 的默认值是 TextInputFormat，它针对文本文件，按行将文本切割成 InputSlit，并用 LineRecordReader 将 InputSplit 解析成<key,value>对，key 是行在文本中的位置，value 是文件中的一行。Map 的结果会通过 partion 分发到 Reducer，Reducer 做完 Reduce 操作后，将通过以格式 OutputFormat 输出。Mapper 最终处理的结果对<key,value>，会送到 Reducer 中进行合并，合并的时候，有相同 key 的键/值对则送到同一个 Reducer 上。Reducer 是所有用户定制 Reducer 类地基础，它的输入是 key 和这个 key 对应的所有 value 的一个迭代器，同时还有 Reducer 的上下文。Reduce 的结果由 Reducer.Context 的 write 方法输出到文件中。

4.MR3 MRShuffle

【案例 4.1】mapreduce 整数排序并指定顺序编号

本试验要求对一系列数据进行排序你，自然就会想到只要把这些数据作为 map 阶段的输出键即可，排序后的数据在输出时，在数值前面添加一个数字的自然顺序编号，此时只要考虑在 reduce 阶段定义一个全局变量，每次执行 reduce 方法时，就是对 map 阶段的 key 进行处理，此时全局变量就可以作为编号，每执行一次 reduce，全局变量就会增加 1。

【案例 4.2】mapreduce 两列数据的二次排序

在 MapReduce 操作时，我们知道传递的<key,value>会按照 key 的大小进行排序，最后输出的结果是按照 key 排过序的。有的时候我们在 key 排序的基础上，对 value 也进行排序。这种需求就是二次排序。二次排序是在框架在对 key2 排序后再对 reduce 输出结果的结果 value3 进行二次排序的需求。在 map 阶段，使用 job.setInputFormatClass 定义的 InputFormat 将输入的数据集分割成小数据块 splites，同时 InputFormat 提供一个 RecordReder 的实现。本例子中使用的是 TextInputFormat，他提供的 RecordReader 会将文本的字节偏移量作为 key，这一行的文本作为 value。

核心总结：

1、map 最后阶段进行 partition 分区，一般使用 job.setPartitionerClass 设置的类，如果没有自定义 Key 的 hashCode()方法进行排序。

2、（第一次排序）每个分区内部调用 job.setSortComparatorClass 设置的 key 的比较函数类进行排序，如果没有则使用 Key 的实现的 compareTo 方法。

3、（第二次排序）当 reduce 接收到所有 map 传输过来的数据之后，调用 job.setSortComparatorClass 设置的 key 比较函数类对所有数据对排序，如果没有则使用 Key 的实现的 compareTo 方法。

4、紧接着使用 job.setGroupingComparatorClass 设置的分组函数类，进行分组，同一个 Key 的 value 放在一个迭代器里面。

【案例 4.3】mapreduce 多条数据去重处理

数据去重的最终目标是让原始数据中出现次数超过一次的数据在输出文件中只出现一次。具体就是 reduce 的输入应该以数据作为 key，而对 value-list 则没有要求。当 reduce 接收到一个<key，value-list>时就直接将 key 复制到输出的 key 中，并将 value 设置成空值。

【案例 4.4】mapreduce 非结构化日志文件处理

需求：根据 tomcat 日志计算 url 访问了情况，具体的 url 如下，

要求：区别统计 GET 和 POST URL 访问量

结果为：访问方式、URL、访问量

5.MR4 MR 调优

【案例 5.1】mapreducetopN 排名

原理：可以在每一个 map 端，就求出开始 TopN，这样可以减少在 reduce 端的

压力，即在每一个 map 中先求出 Top10，然后将这 top10,传给 reduce，让后 reduce 把所有全部的传来的数据中找出 top10

【案例 5.2】mapreduce 统计拨打公共服务号码的电话信息

原理：首先对待处理的信息进行拆分，拆分之后在 map 阶段，把公共服务电话作为 map 方法的输出键，用户电话作为 map 方法的输出值，最后在 reduce 阶段对每个键的值集合进行遍历并按指定规则拼接起来，输出结果即可。

【案例 5.3】mapreduce 学生信息按年龄分区

原理：首先把学生的信息进行封装成新的类型，作为 map’阶段的输出值，因为不需要按键排序，所以键只需要设置为 NullWritable 类型即可。在 reduce 阶段对所有的值进行遍历之后直接输出即可

【案例 5.4】mapreduce 电话号码流量封装类作为键并排序

原理：首先按电话号码记录信息的封装类作为键进行排序，同时为电话记录的封装类型指定所属的数据类型（Writable），重写排序方法和规则。

【案例 5.5】mapreduce 电话号码流量统计并分区

原理：首先按电话号码作为键进行排序，相同键的内容形成一个集合，然后把相同键的所有内容值进行流量相加，最后按照指定分区条件进行分区输出。

实验过程

1. MR1-1 词频统计

一、项目准备阶段

1.1 在 linux 系统的命令终端上切换到~/simple 目录，执行命令：touch source.txt 创建一个文件。

```
(base) [root@ferry ~]# cd ~/simple
(base) [root@ferry simple]# touch source.txt
(base) [root@ferry simple]# ls
core-site.xml  Hadoop-2.4.1      hw  Hw.c  mapper.sh  soft        test.txt
hadoop-2.4.1   HelloWorld.class  Hw  jdk   reducer.sh  source.txt  word.txt
(base) [root@ferry simple]#
```

1.2 在 simple 目录下，执行命令：vi ~/simple/source.txt 编辑该文件，并把数据的信息内容拷贝到该文件中，然后在 simple 目录可以查看到 source.txt 文件。

```
(base) [root@ferry simple]# vi source.txt
(base) [root@ferry simple]# cat source.txt
The ASF provides an established framework
for intellectual property and financial
contributions that simultaneously limits
potential legal exposure for
our project committers
The ASF provides an established framework
for intellectual property and financial
contributions that simultaneously limits
potential legal exposure for
our project committers
```

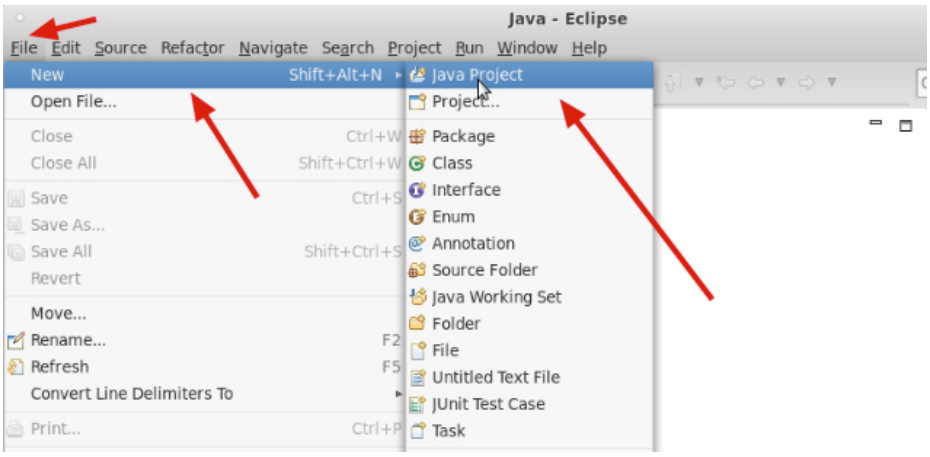
1.3 本案例需要用到 hadoop 的存储和计算，所以在编写程序之前需要先启动 Hadoop 服务，在命令终端执行命令：start-all.sh 把 hdfs 和 yarn 服务启动。

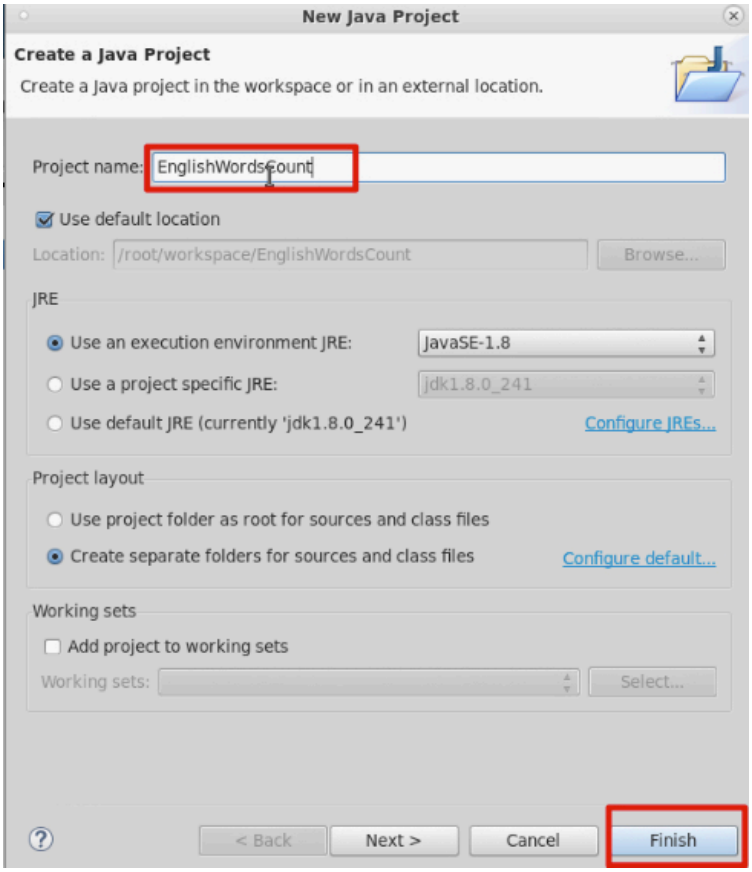
```
(base) [root@ferry simple]# start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
20/05/28 20:04:53 WARN util.NativeCodeLoader: Unable to load native-hadoop library for
atform... using builtin-java classes where applicable
Starting namenodes on [ferry.szu]
ferry.szu: starting namenode, logging to /root/simple/hadoop-2.4.1/logs/hadoop-root-nam
```

通过 jps 命令可以看到服务已启动

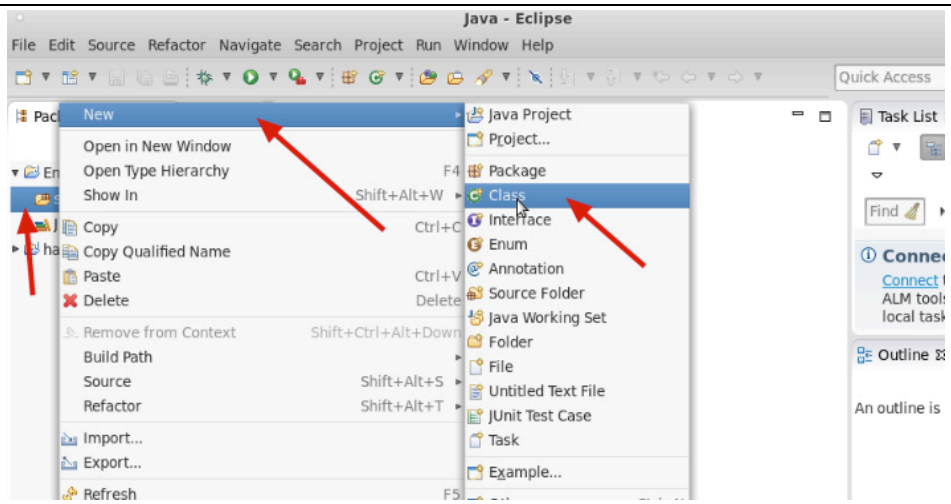
二 程序编写

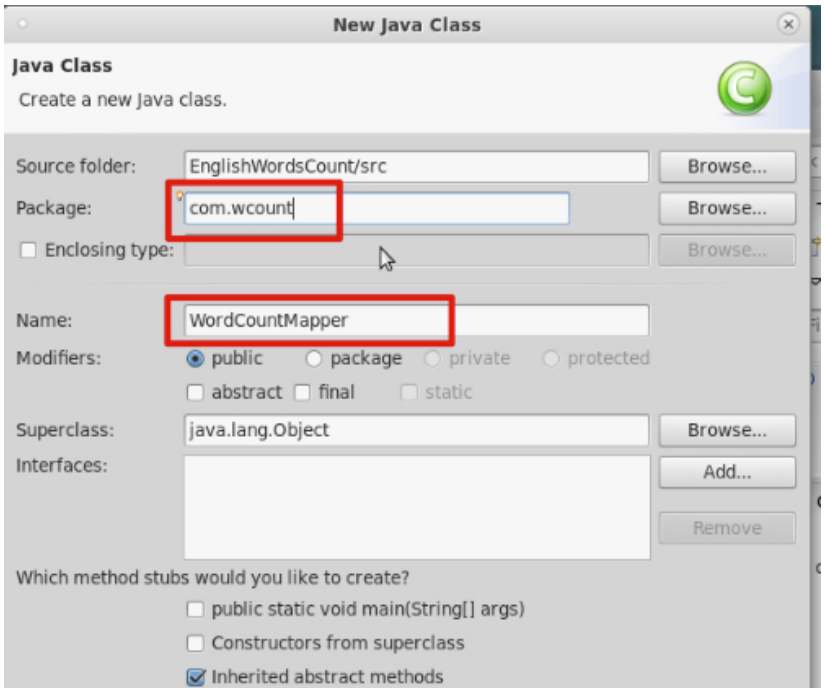
2.1 在 eclipse 中的项目列表中，右键点击，选择“new“—>”Java Project…”新建一个项 目“EnglishWordsCount” 。





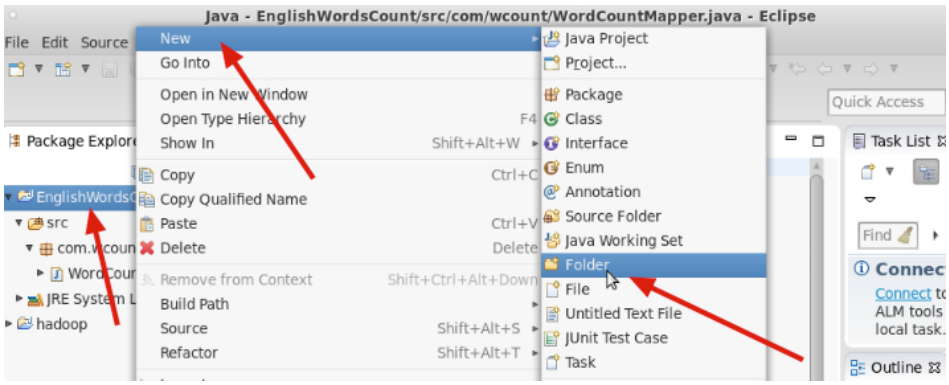
2.2 在项目 src 目录下，右键点击，选择“新建”创建一个类文件名称为“WordCountMapper” 并指定包名” com.wcount” 。

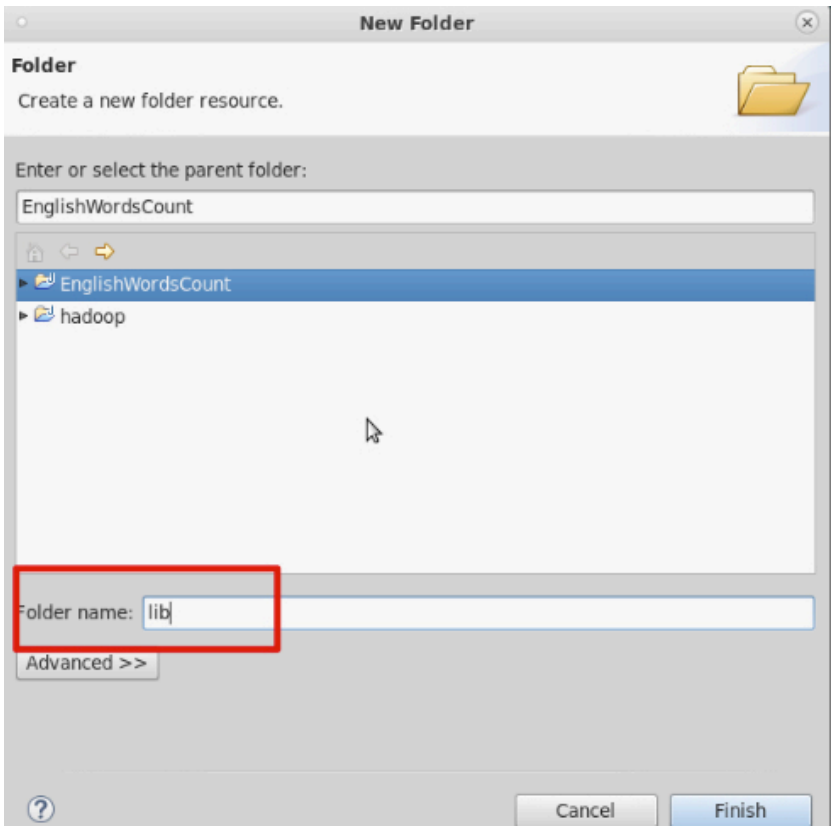




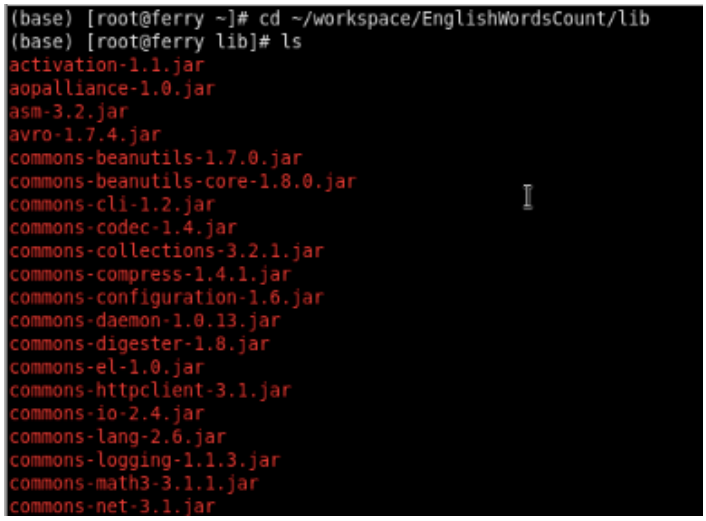
2.3 在编写“WordCountMapper”类之前需要把 hadoop 相关的 jar 包导入，首先右击 EnglishWordsCount 项目选择“New”—“Folder”创建一个 lib 文件夹并将 hadoop 相关的 jar 包导入到此 lib 文件夹中。需要的 jar 包包括：

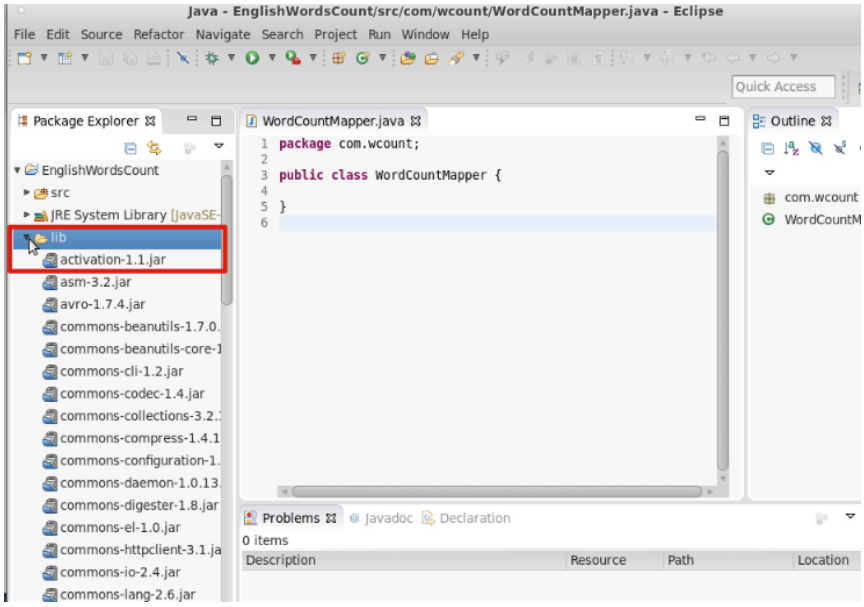
hadoop-2.4.1/share/hadoop/hdfs/hadoop-hdfs-2.4.1.jar
hadoop-2.4.1/share/hadoop/hdfs/lib/ 所有 jar 包
hadoop-2.4.1/share/hadoop/common/hadoop-common-2.4.1.jar
hadoop-2.4.1/share/hadoop/common/lib/ 所有 jar 包
hadoop-2.4.1/share/hadoop/mapreduce/ 除
hadoop-mapreduce-examples-2.4.1.jar 之外的 jar 包
hadoop-2.4.1/share/hadoop/mapreduce/lib/ 所有 jar 包
hadoop-2.4.1/share/hadoop/yarn 所有 jar 包
hadoop-2.4.1/share/hadoop/yarn/lib/所有 jar 包



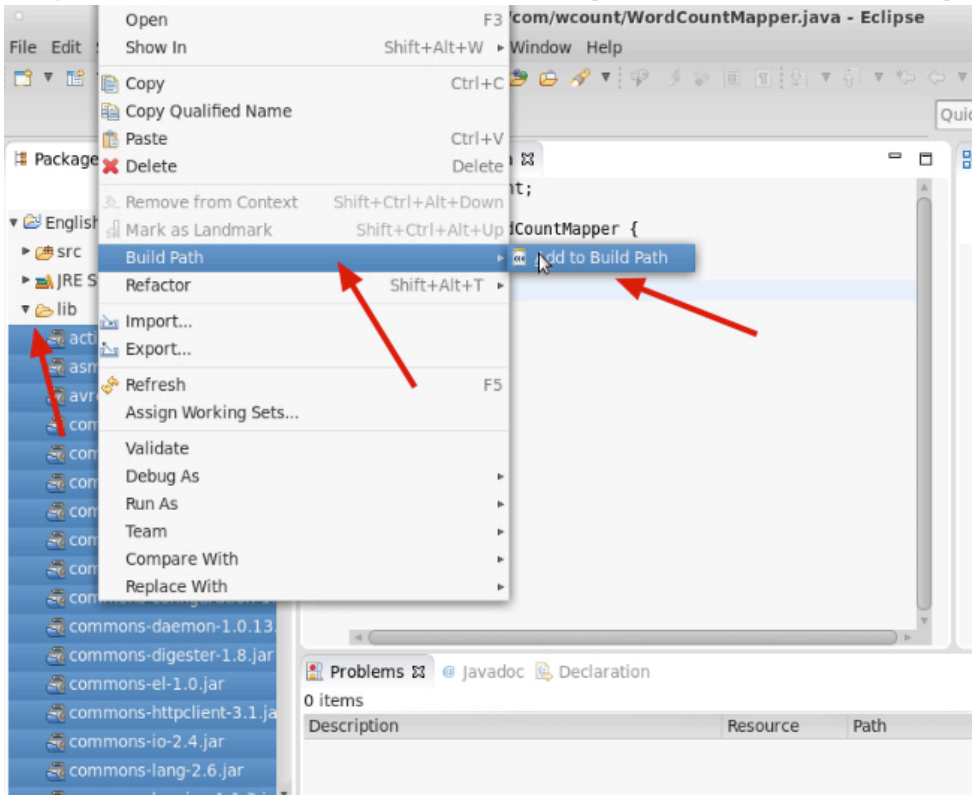


完成导入，可以查看如下图所示：

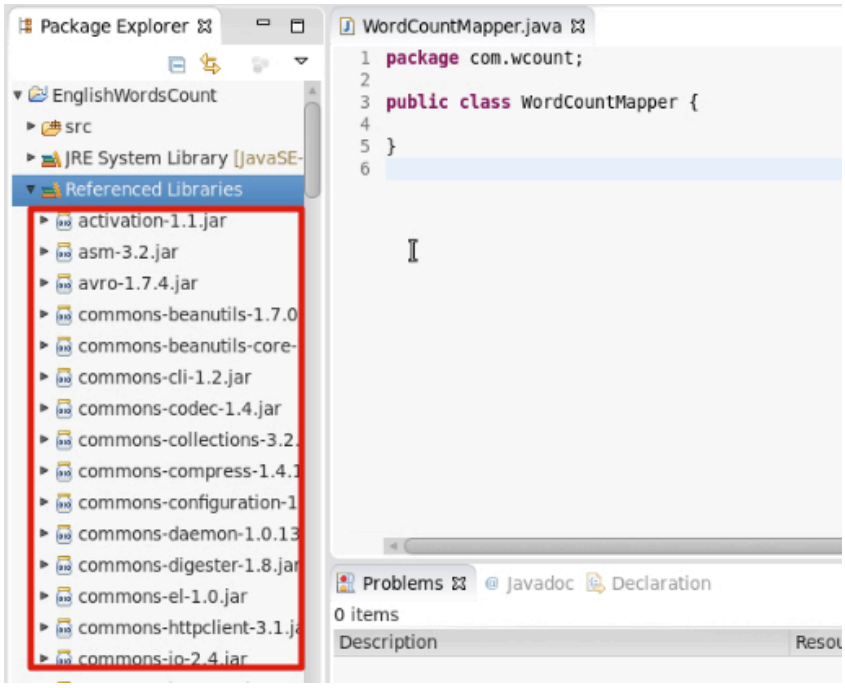




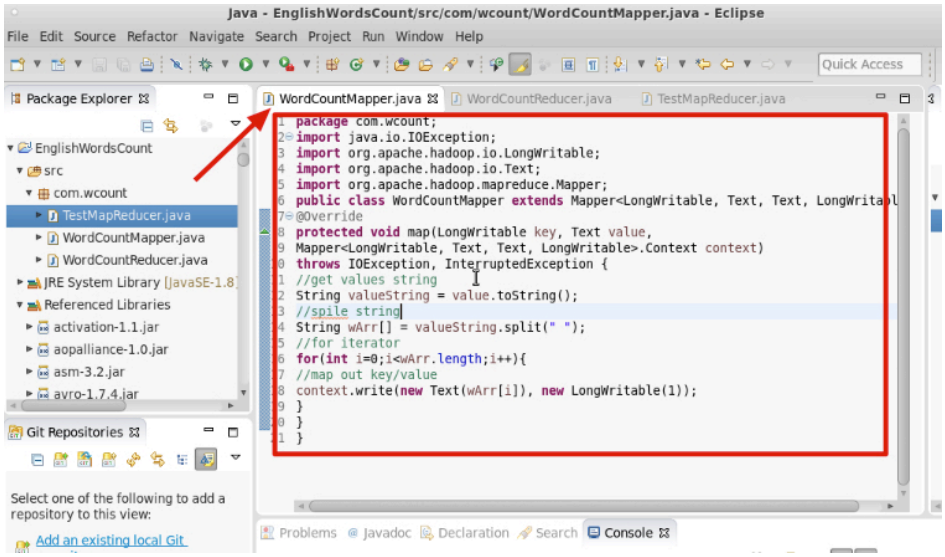
2.4 把 lib 下所有的 jar 包导入到环境变量，首先全选 lib 文件夹下的 jar 包文件，右键点击， 选择“build path”-->“add to build path”



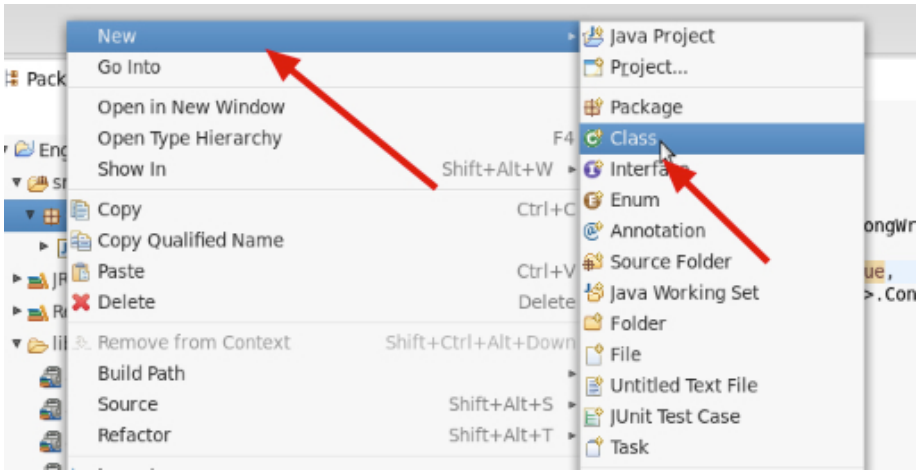
添加后，发现在项目下很多奶瓶图表的 jar 包。

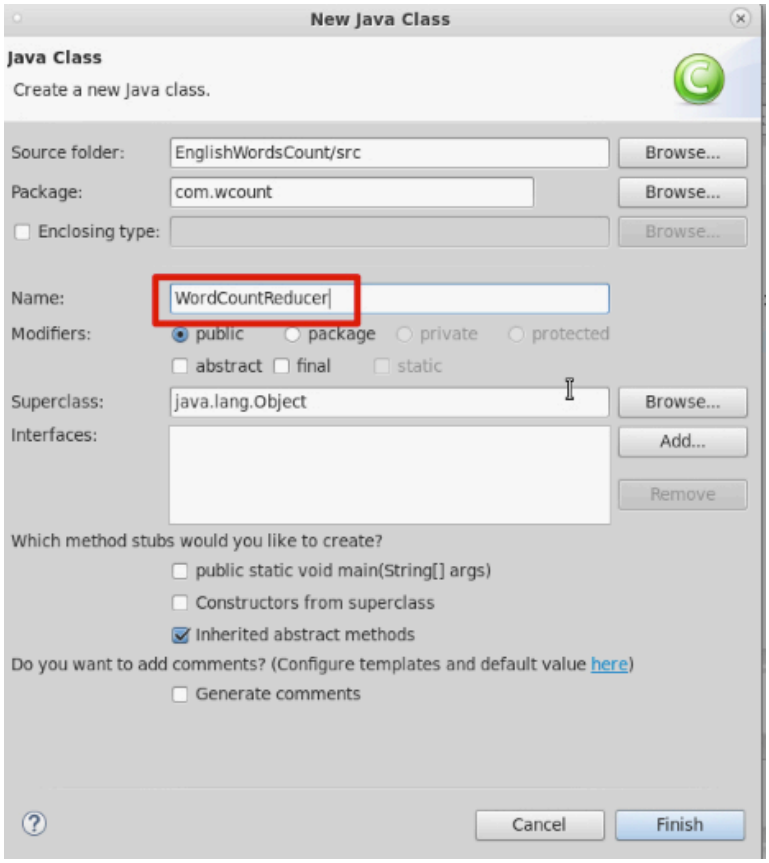


2.5 让类“WordCountMapper”继承类 Mapper 同时指定需要的参数类型，根据业务逻辑修改 map 类的内容如下。

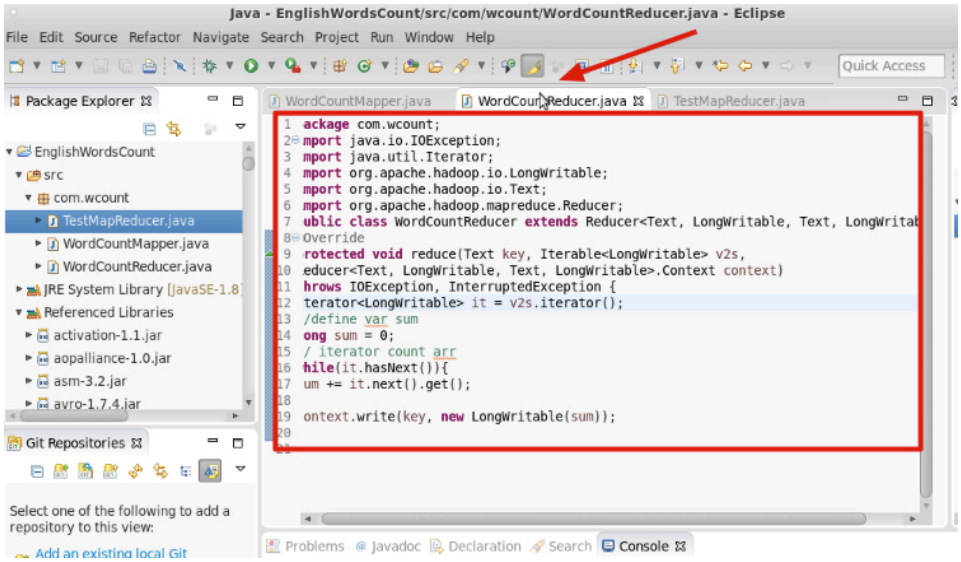


2.6 在项目 src 目录下指定的包名” com.wcount ”下右键点击， 新建一个类名为 “WordCountReducer “并继承 Reducer 类

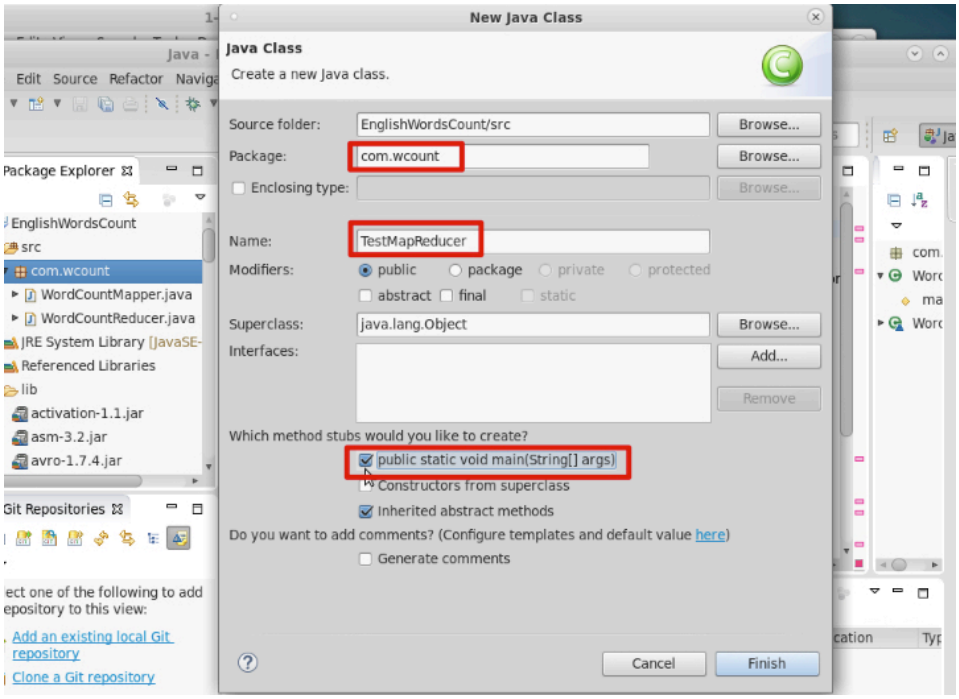




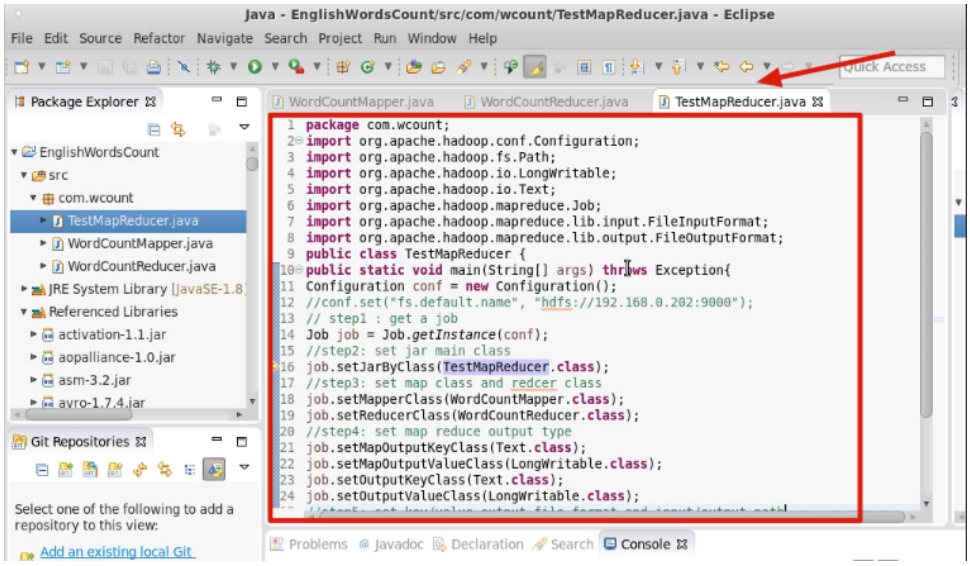
然后添加该类中的代码内容如下所示。



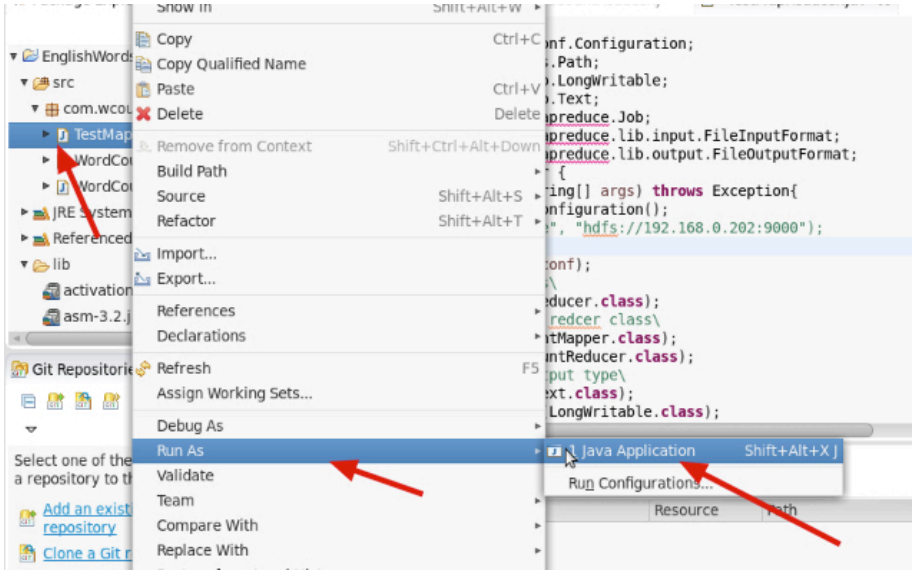
2.7 在项目 src 目录下指定的包名” com.wcount”下右键点击，新建一个测试主类名为” TestMapReducer”并指定 main 主方法。如图所示



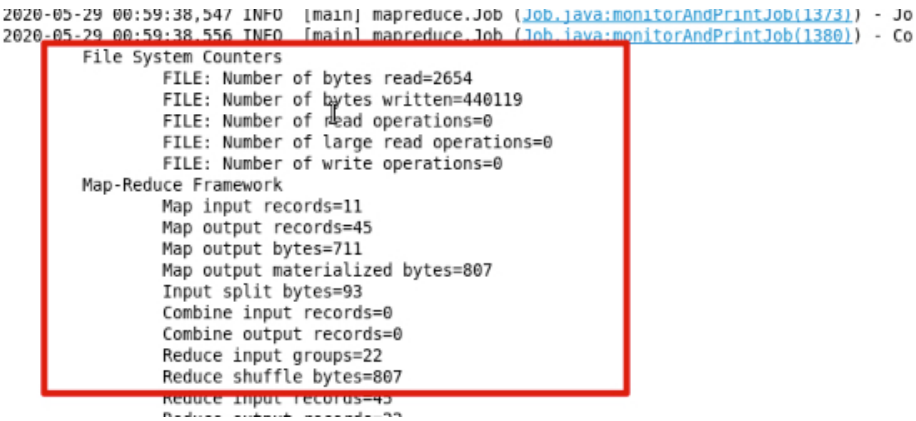
2.8 编写测试代码如下所示。



2.9 按照以上的步骤，把 mapper 和 reducer 阶段以及测试代码编写完毕之后，选中测试类” TestMapReducer “，右键点击选择”Run as”--->” Java Application”，

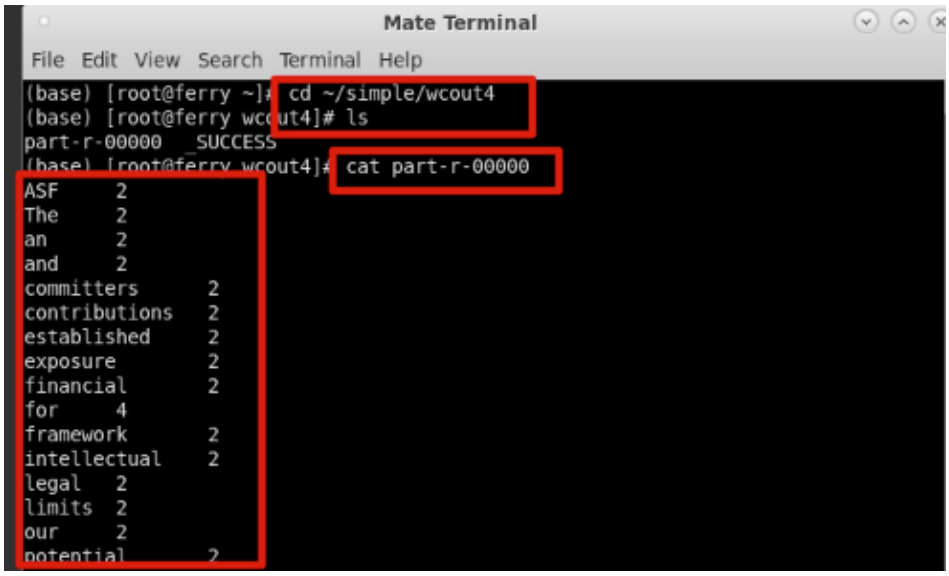


查看控制台显示内容，查看是否正确执行。如图所示



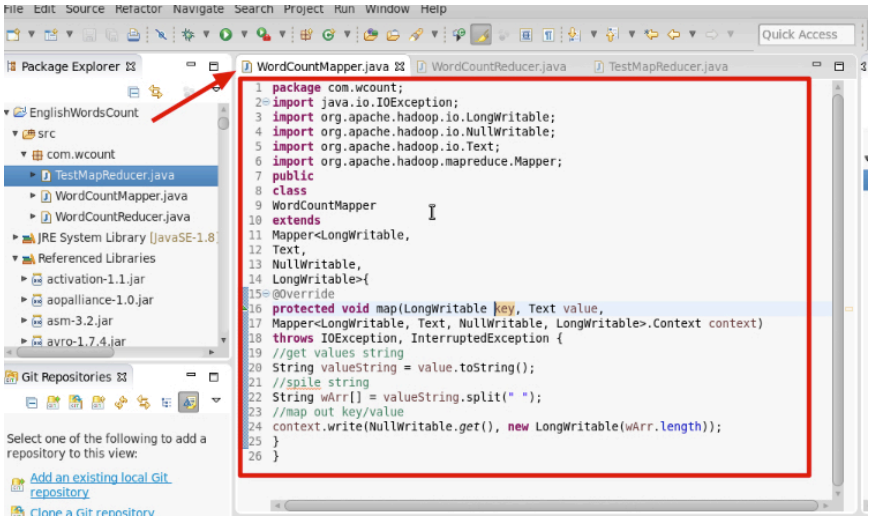
2.10 程序执行完毕之后， 可以到输出信息目录 /simple/output 下， 执

行查看命令 :cat part-r-00000 查看对数据处理后产生的结果。如图所示

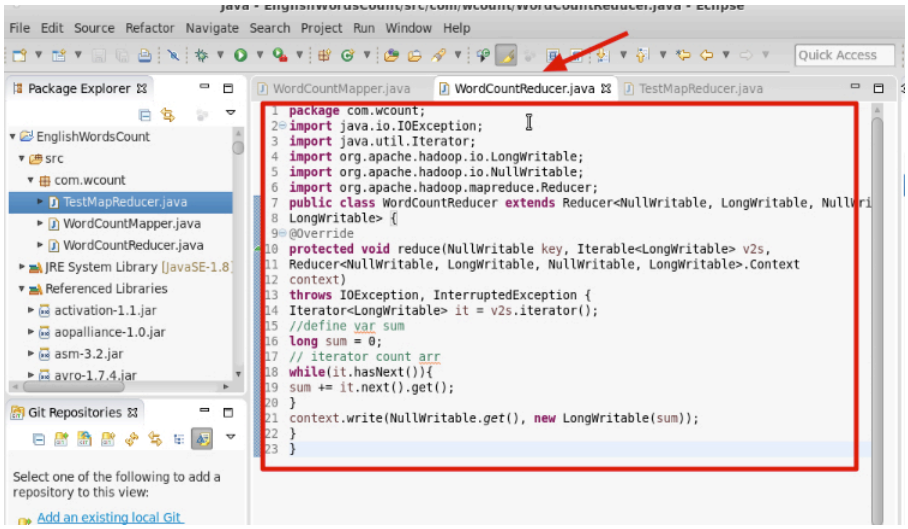


2. MR1-2 单词个数

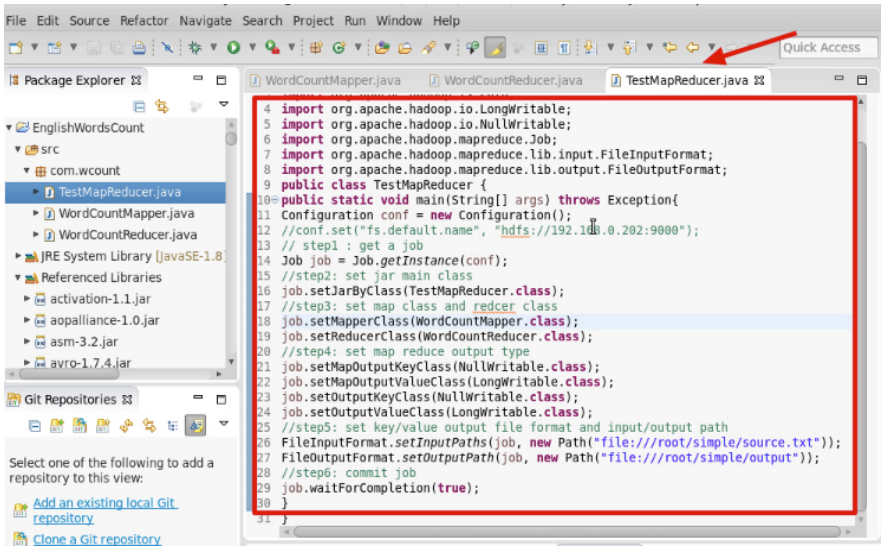
2.1 在 MR1-1 的基础上，修改 WordCountMapper 类的代码如图。



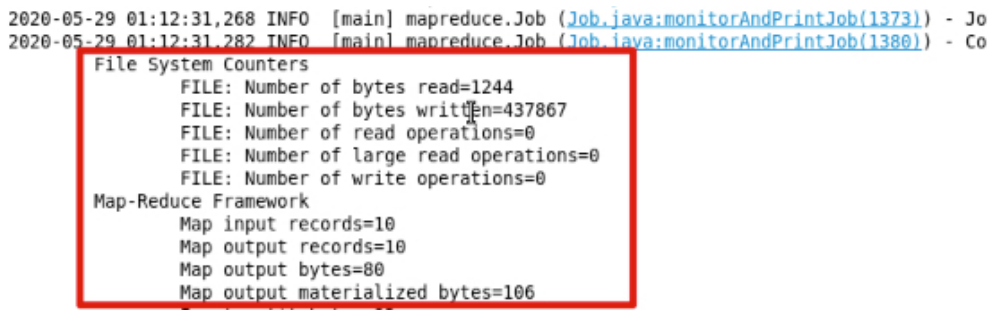
2.2 修改 WordCountReducer 类的代码如图。



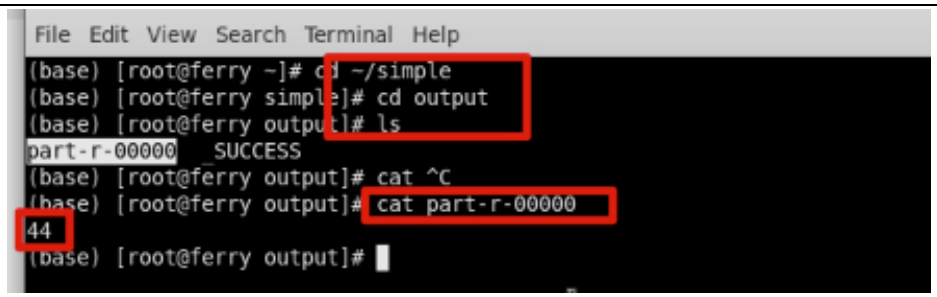
2.3 修改 TestMapReducer 类的代码如图。



2.4 选中测试类“TestMapReducer”，右键点击选择 Run as— Java Application。检查控制台内容，查看是否正确执行。



2.5 在输出信息目录/root/simple/output 下，执行 cat part-r-00000 查看数据处理结果。



3.MR2-2 成绩统计

【案例 3.1】 mapreduce 学生总成绩报表

一、项目准备阶段

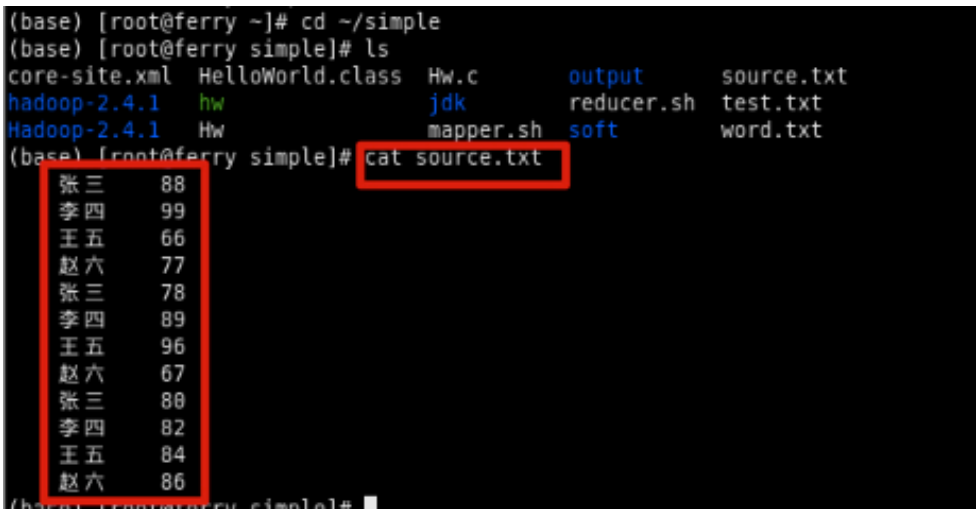
需求描述：

对输入文件中数据进行计算学生总成绩。输入文件中的每行内容均为一个学生的姓名和他相应的成绩，如果有多门学科，则每门学科为一个文件。

要求在输出中每行有两个间隔的数据，其中，第一个代表学生的姓名，第二个代表其总成绩。

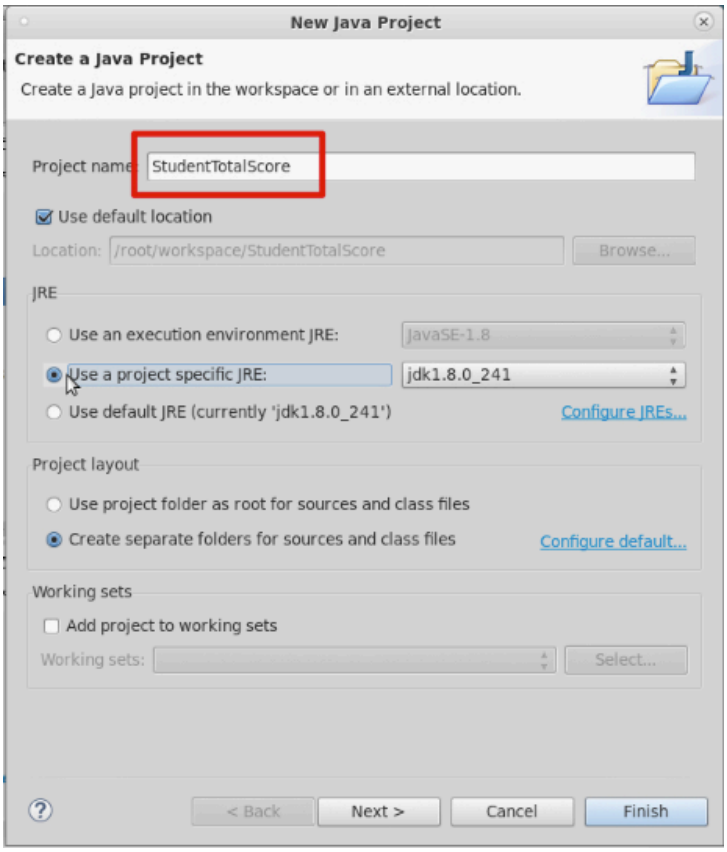
原始数据：

张三 88
李四 99
王五 66
赵六 77
张三 78
李四 89
王五 96
赵六 67
张三 80
李四 82
王五 84
赵六 86

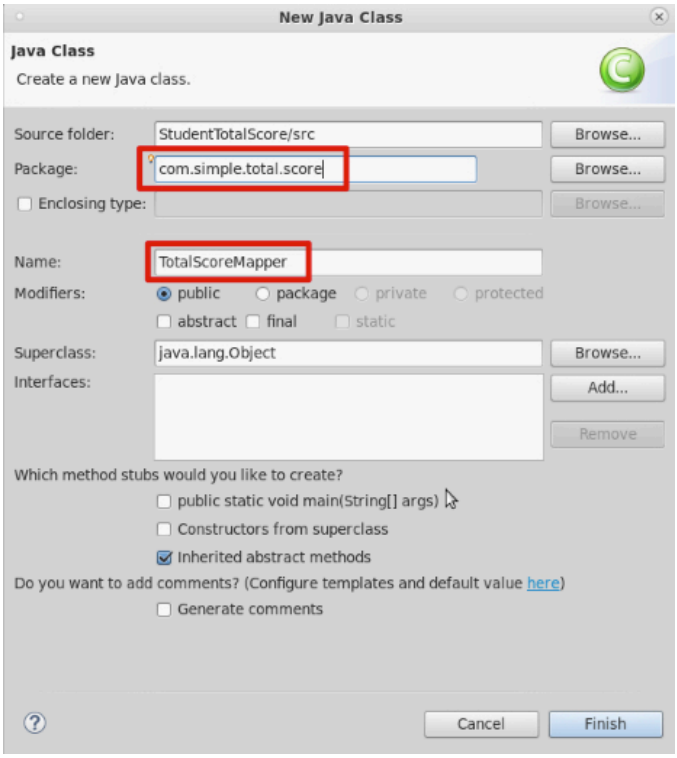


二、程序编写

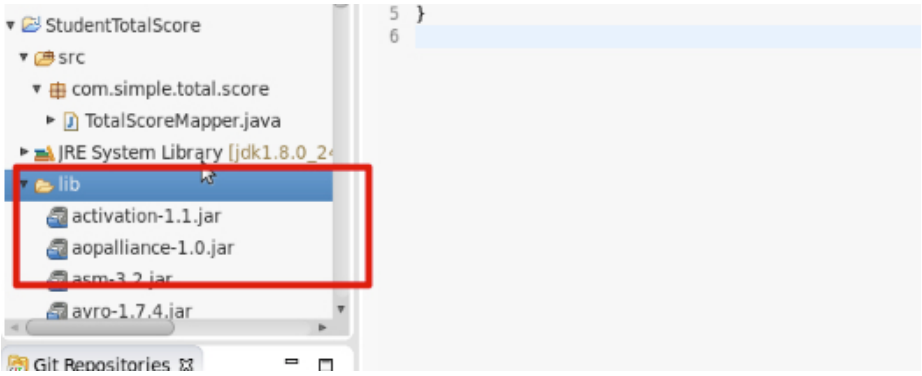
2.1 在 eclipse 中的项目列表中，右键点击，选择“new“—>”Java Project…”新建一个项目“StudentTotalScore” 。



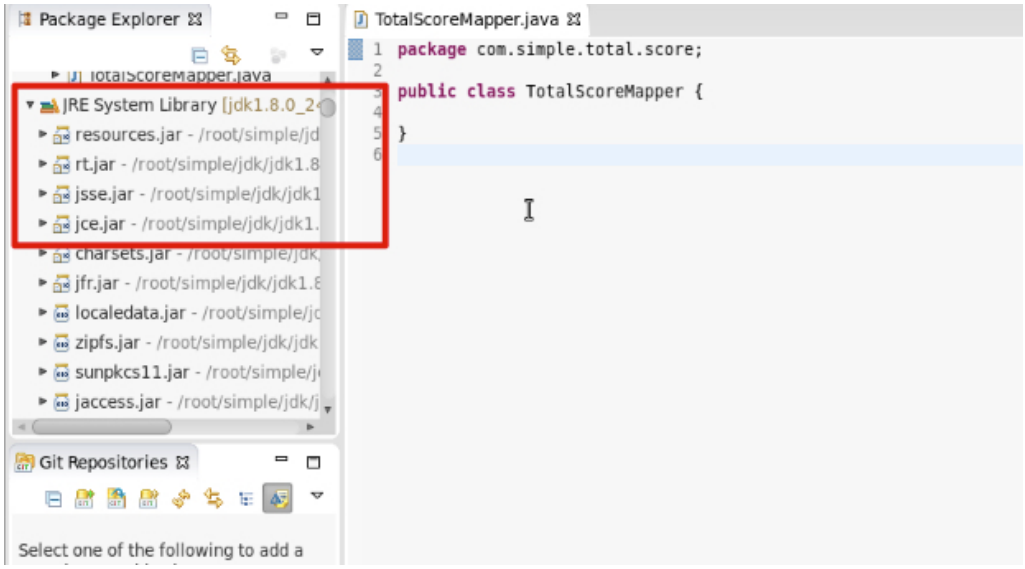
2.2 在项目 src 目录下，右键点击，选择“新建”创建一个类文件名称为“TotalScoreMapper”并指定包名” com.simple.total.score” 。



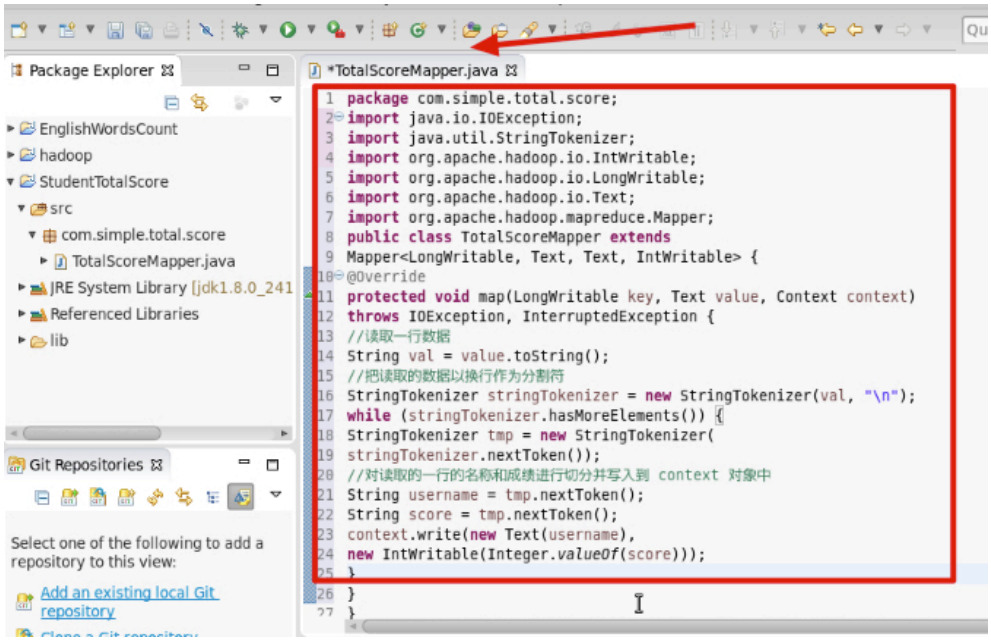
2.3 在编写“TotalScoreMapper”类之前需要把 hadoop 相关的 jar 包导入，首先右击项目选择“New”—“Folder”创建一个 lib 文件夹,并把指定位置中(同案例 MR1-1)的包放入该文件中。



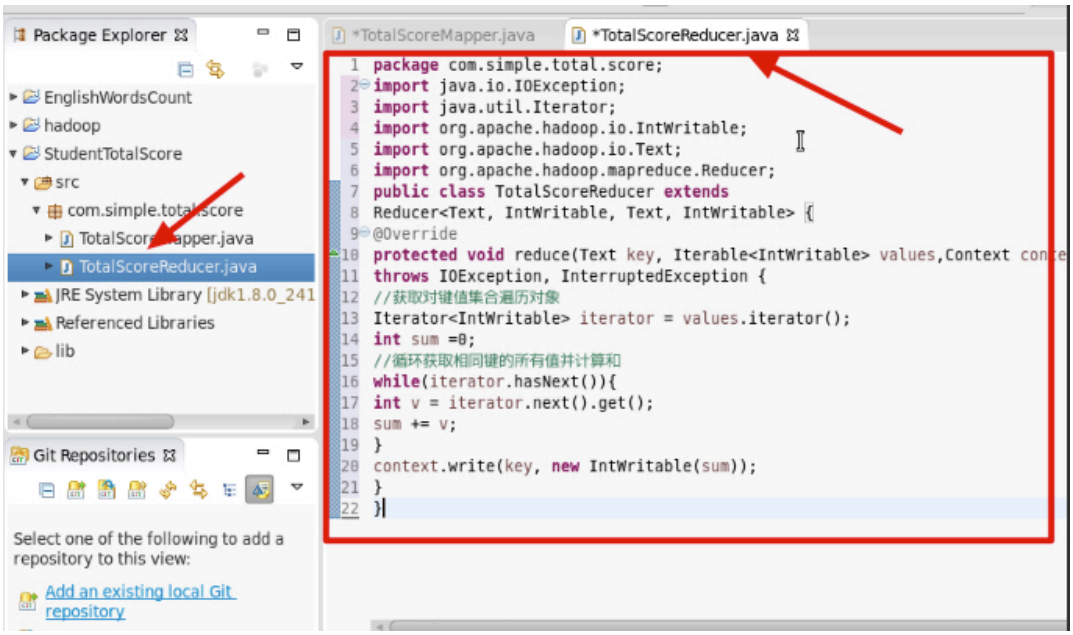
2.4 把 lib 下所有的 jar 包导入到环境变量，首先全选 lib 文件夹下的 jar 包文件，右键点击，选择“build path”-->“add to build path”,添加后，发现在项目下很多奶瓶图表的 jar 包。



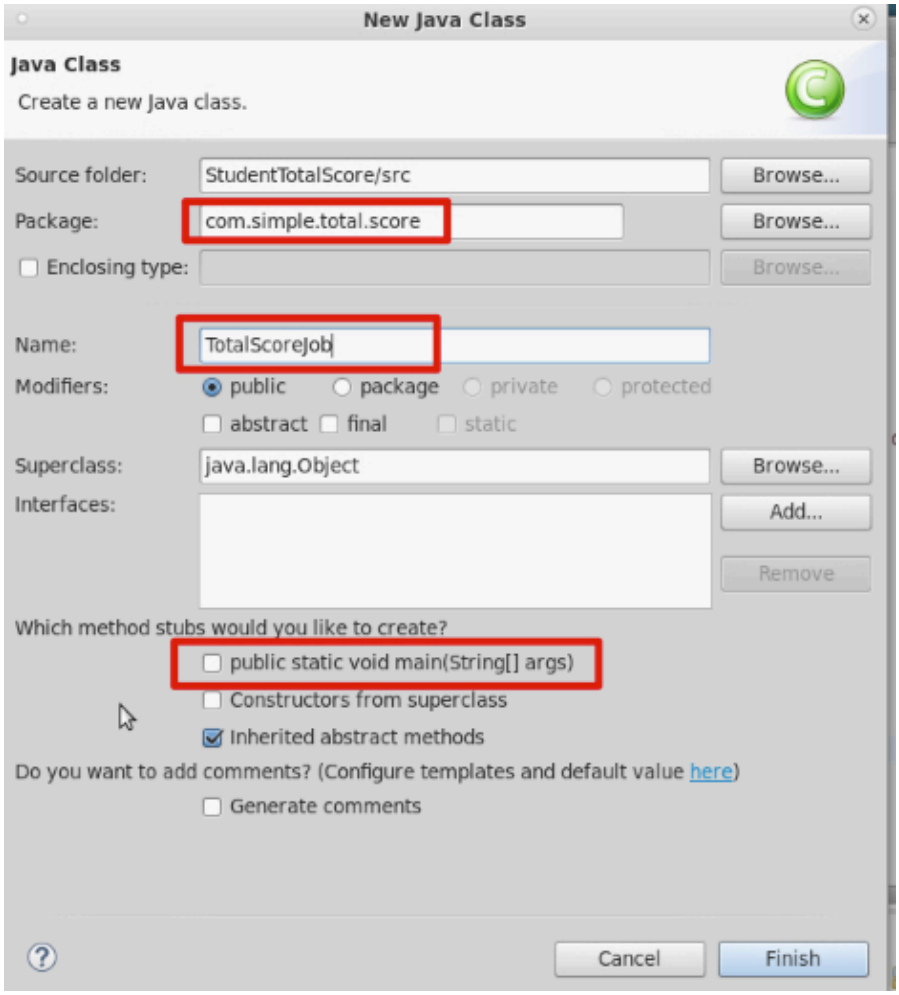
2.5 让类“TotalScoreMapper”继承类 Mapper 同时指定需要的参数类型，根据业务逻辑修改 map 类的内容如下。



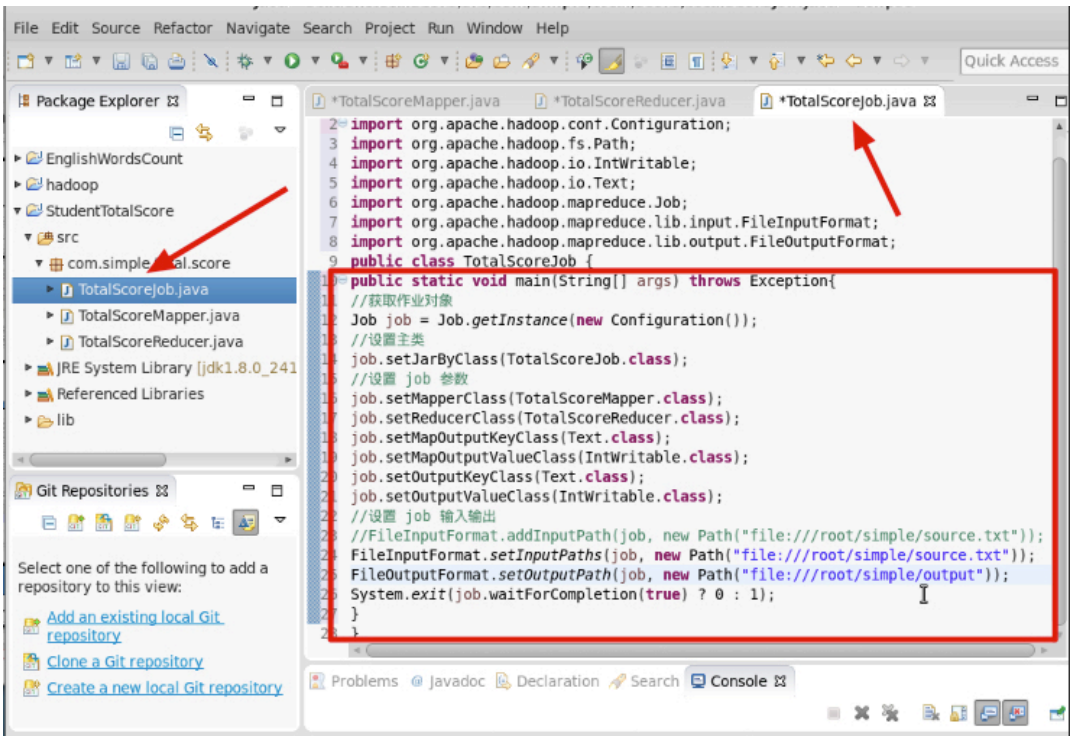
2.6 在项目 src 目录下指定的包名” com.simple.total.score”下右键点击，新建一个类名为“TotalScoreReducer “并继承 Reducer 类，然后添加该类中的代码内容如下所示。



2.7 在项目 src 目录下指定的包名” com.simple.total.score”下右键点击，新建一个测试主类名为” TotalScoreJob”并指定 main 主方法。

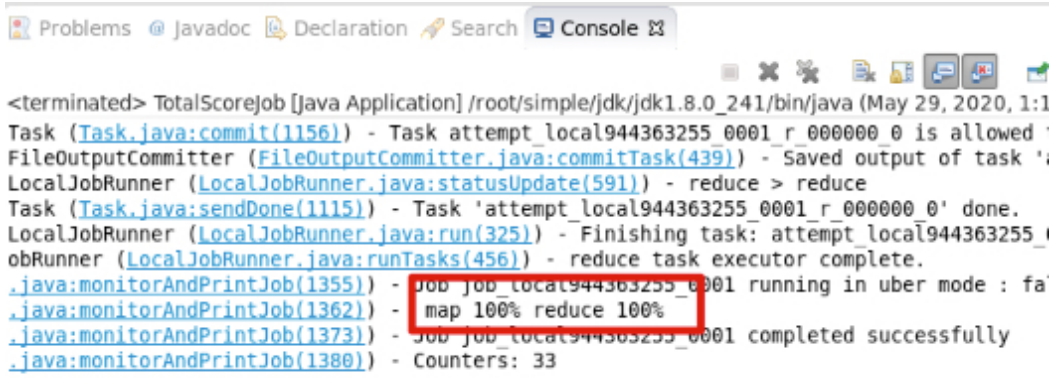


2.8 添加测试代码如下所示。

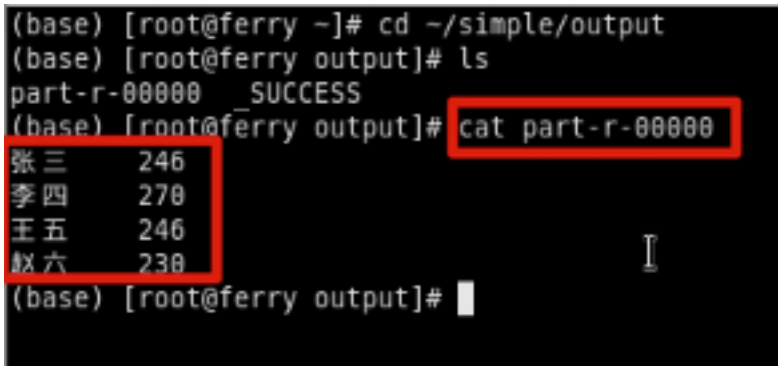


2.9 按照以上的步骤，把 mapper 和 reducer 阶段以及测试代码编写完毕之

后，选中测试类” AvgScoreJob “，右键点击选择”Run as”--->”Java Application”,查看控制台显示内容查看是否正确执行。



2.10 程序执行完毕之后，可以到输出信息目录~/simple/output 下，执行查看命令:cat part-r-00000 查看对数据处理后产生的结果。



【案例 3.2】mapreduce 学生各科平均成绩报表

一、项目准备阶段

需求描述：

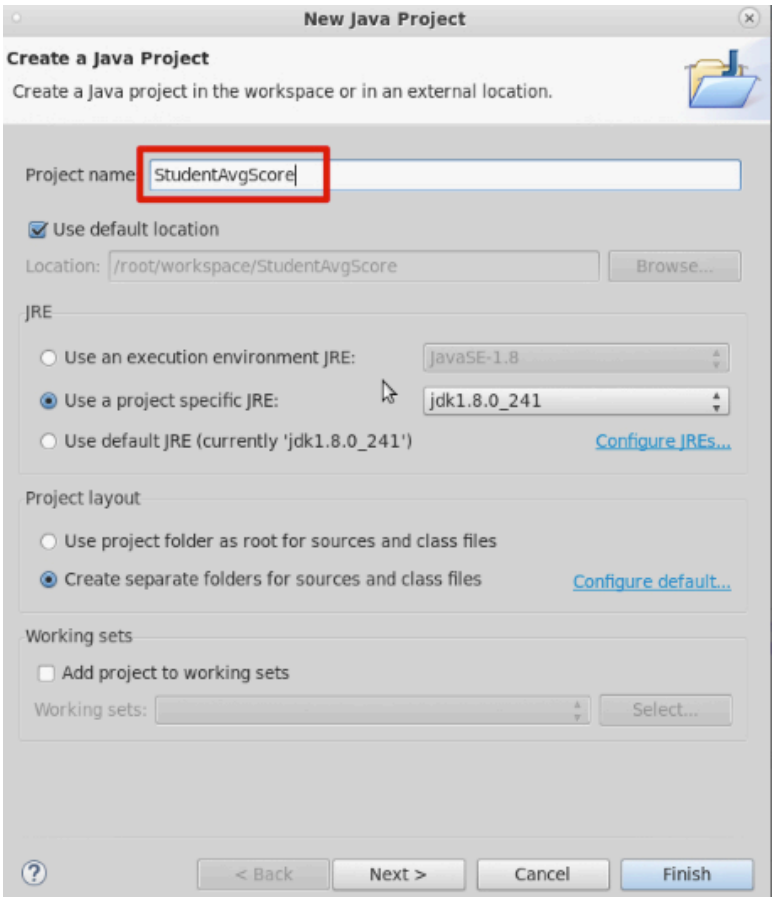
对输入文件中数据进行就算学生平均成绩。输入文件中的每行内容均为一个学生的姓名和他相应的成绩，如果有多门学科，则每门学科为一个文件。

要求在输出中每行有两个间隔的数据，其中，第一个代表学生的姓名，第二个代表其平均成绩。

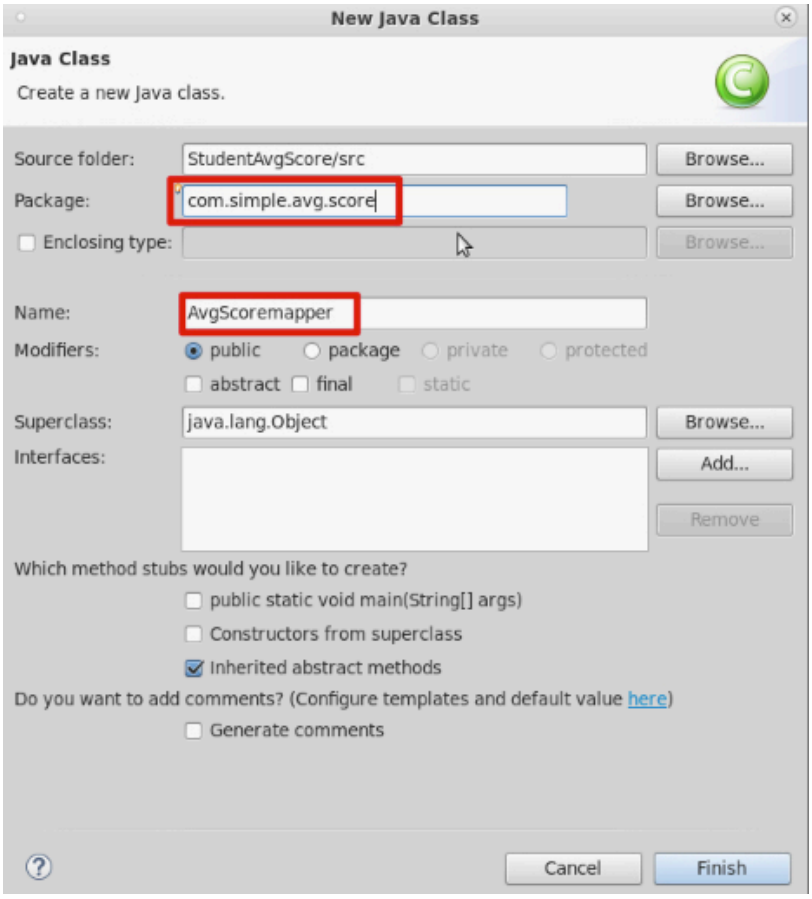
```
(base) [root@ferry ~]# cd ~/simple
(base) [root@ferry simple]# ls
core-site.xml  HelloWorld.class  Hw.c       output      source.txt
hadoop-2.4.1   hw                jdk        reducer.sh  test.txt
Hadoop-2.4.1   Hw                mapper.sh  soft        word.txt
(base) [root@ferry simple]# cat source.txt
张三    88
李四    99
王五    66
赵六    77
张三    78
李四    89
王五    96
赵六    67
张三    80
李四    82
王五    84
赵六    86
```

二、程序编写

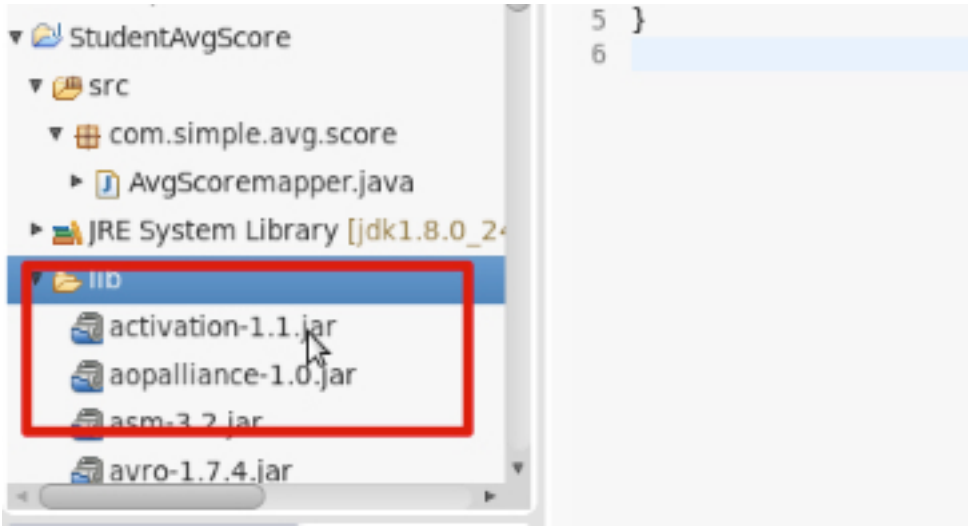
2.1 在 eclipse 中的项目列表中，右键点击，选择“new“—>”Java Project…”新建一个项目“StudentAvgScore” 。



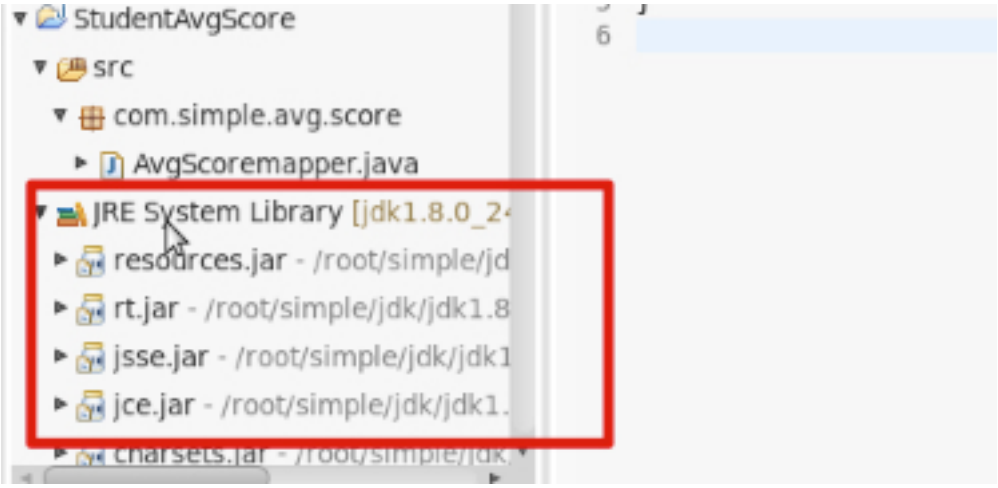
2.2 在项目 src 目录下，右键点击，选择“新建”创建一个类文件名称为“AvgScoreMapper”并指定包名” com.simple.avg.score” 。



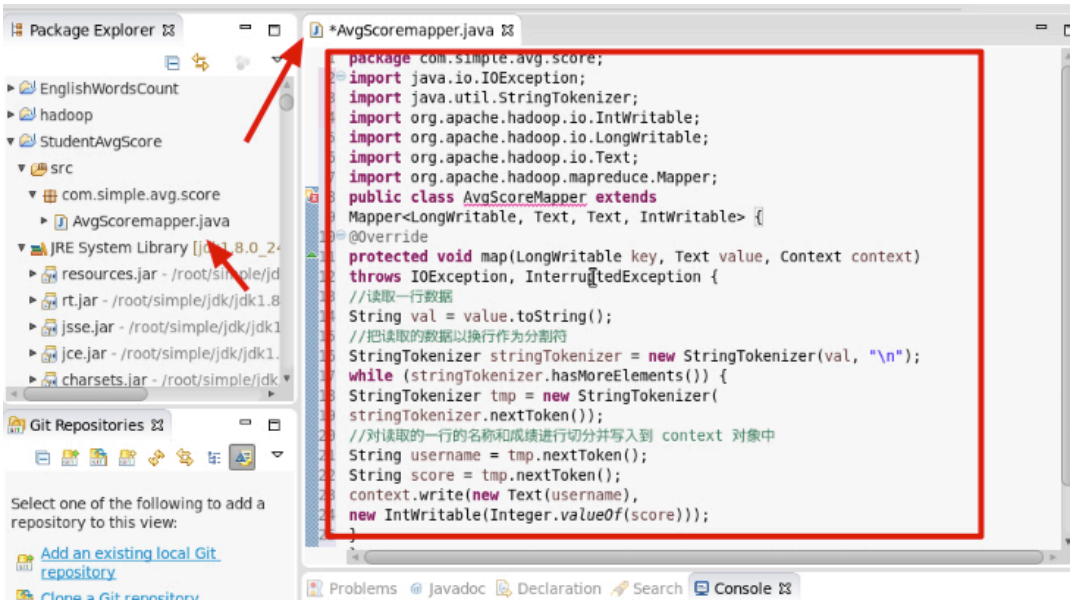
2.3 在编写“AvgScoreMapper”类之前需要把 hadoop 相关的 jar 包导入，首先右击项目选择“New”—“Folder”创建一个 lib 文件夹并把指定位置中(桌面 lib 文件夹)的包放入该文件中。



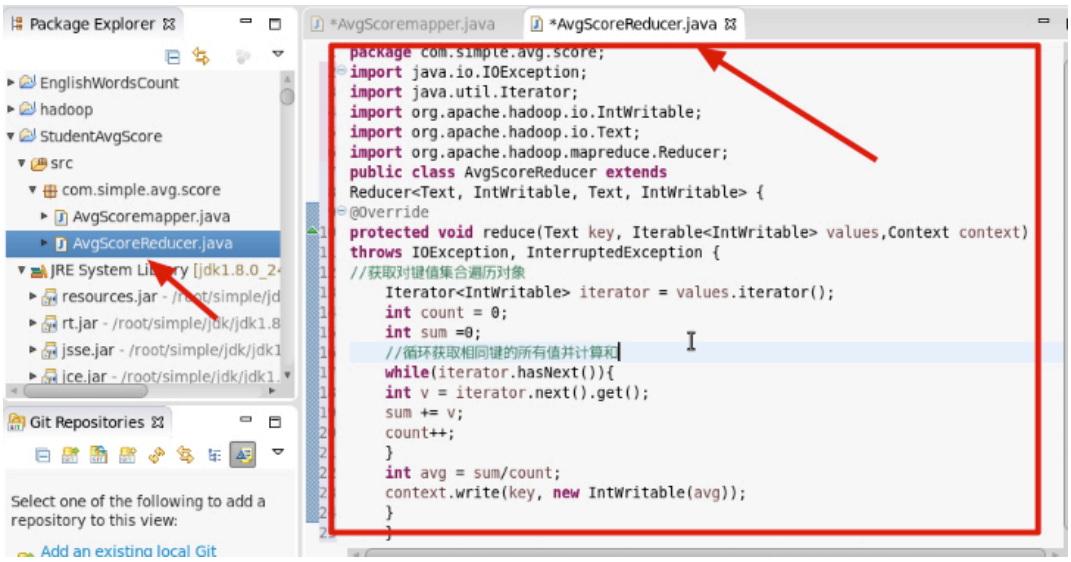
2.4 把 lib 下所有的 jar 包导入到环境变量，首先全选 lib 文件夹下的 jar 包文件，右键点击，选择“build path”-->“add to build path”,添加后，发现在项目下很多奶瓶图表的 jar 包。



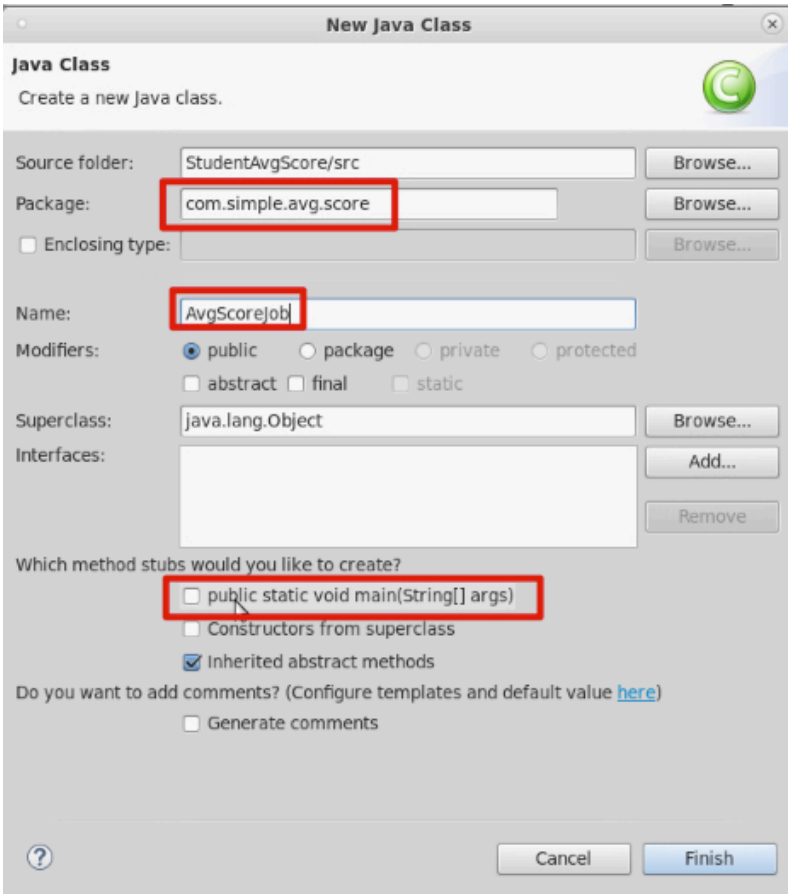
2.5 让类“AvgScoreMapper”继承类 Mapper 同时指定需要的参数类型，根据业务逻辑修改 map 类的内容如下。



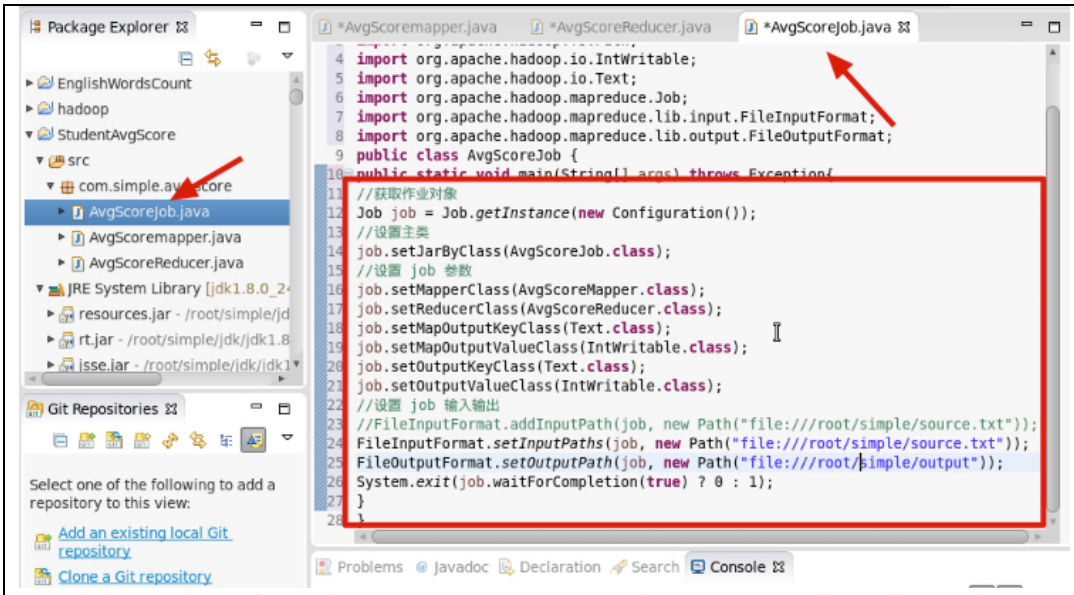
2.6 在项目 src 目录下指定的包名” com.simple.avg.score”下右键点击，新建一个类名为“AvgScoreReducer “并继承 Reducer 类，然后添加该类中的代码内容如下所示。



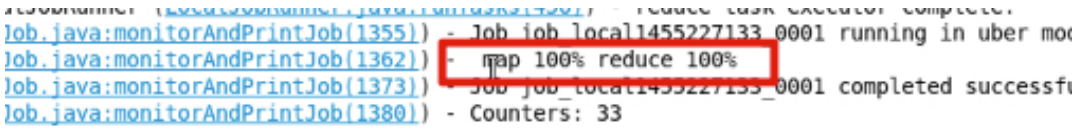
2.7 在项目 src 目录下指定的包名” com.simple.avg.score”下右键点击，新建一个测试主类名为” AvgScoreJob”并指定 main 主方法。



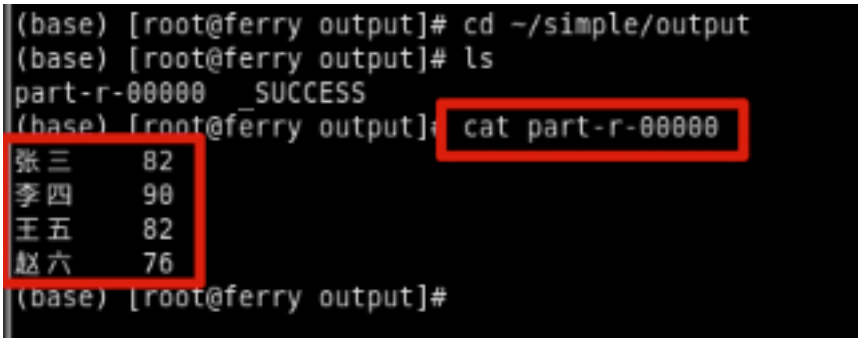
2.8 测试代码如下：



2.9 按照以上的步骤，把 mapper 和 reducer 阶段以及测试代码编写完毕之后，选中测试类” AvgScoreJob “，右键点击选择”Run as”--->”Java Application”，查看控制台显示内容查看是否正确执行。



2.10 程序执行完毕之后，可以到输出信息目录/simple/output 下，执行查看命令:cat part-r-00000 查看对数据处理后产生的结果。



4.MR3 MPShuffle

【案例 4.1】mapreduce 整数排序并指定顺序编号

一、项目准备阶段

需求描述：

对指定文件 source.txt 中的整形数据进行排序，并且排序后的数据在输出时，给每个整形数据一个编号

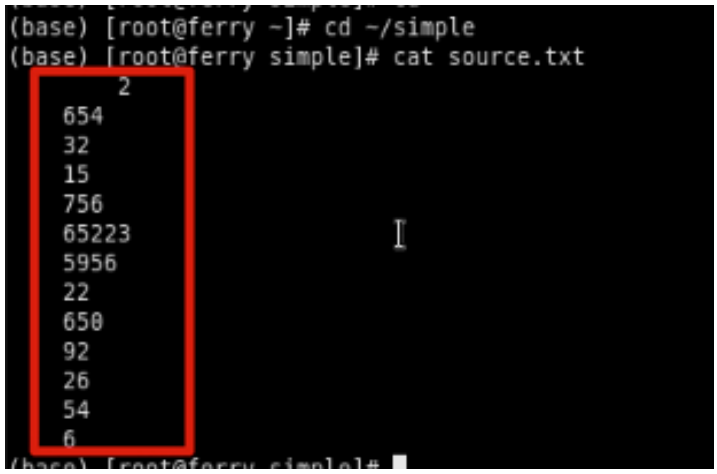
表原始数据。

2

654

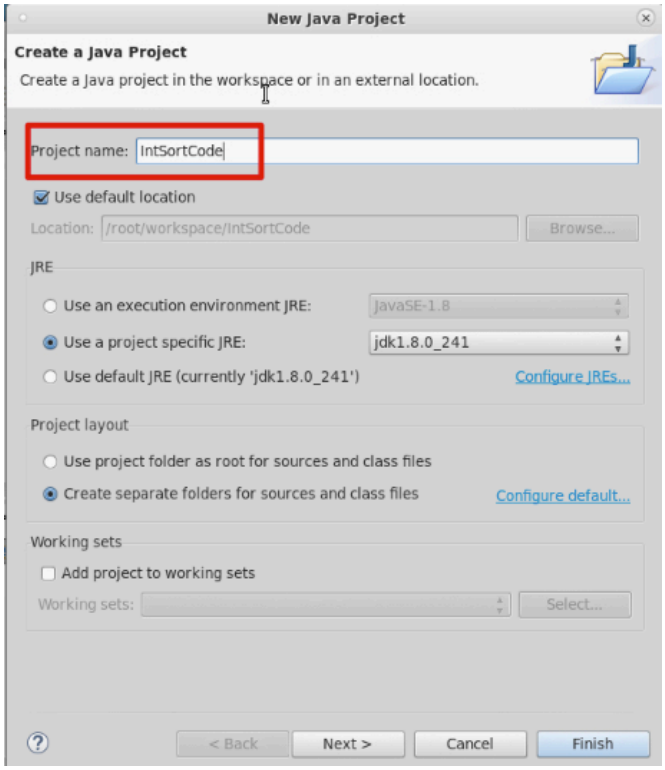
32

15
756
65223
5956
22
650
92
26
54
6

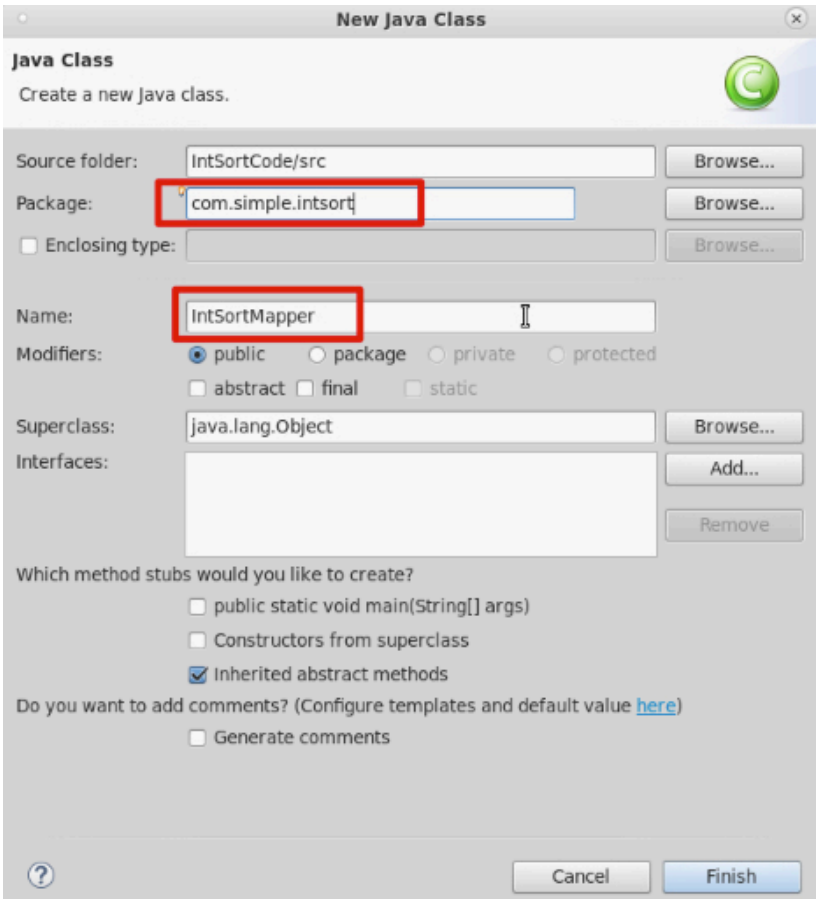


二、程序编写

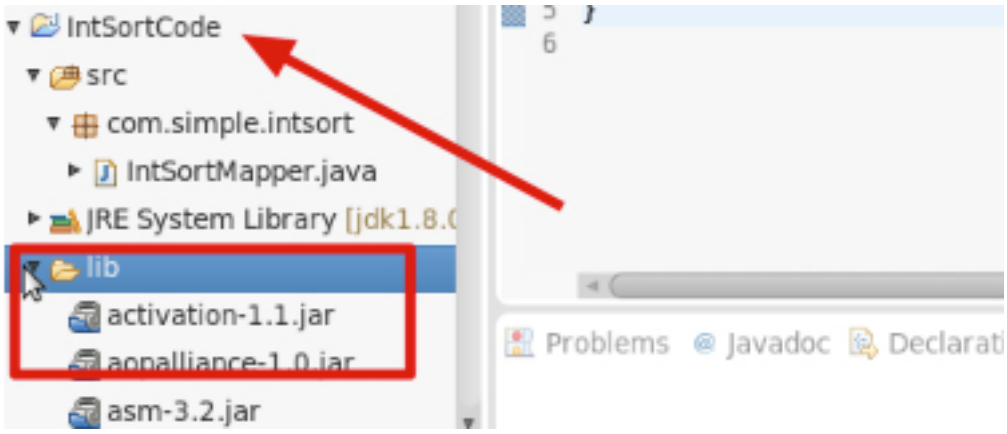
2.1 在 eclipse 中的项目列表中，右键点击，选择“new“—>”Java Project…”新建一个项目“IntSortCode” 。



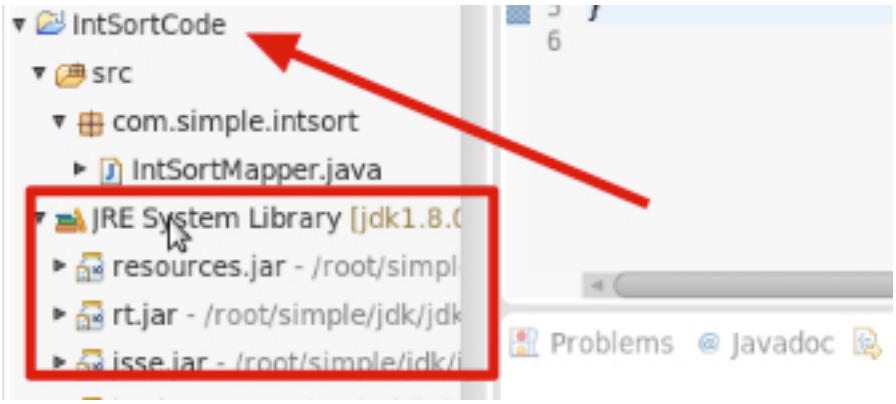
2.2 在项目 src 目录下，右键点击，选择“新建”创建一个类文件名称为“IntSortMapper”并指定包名” com.simple.intsort” 。



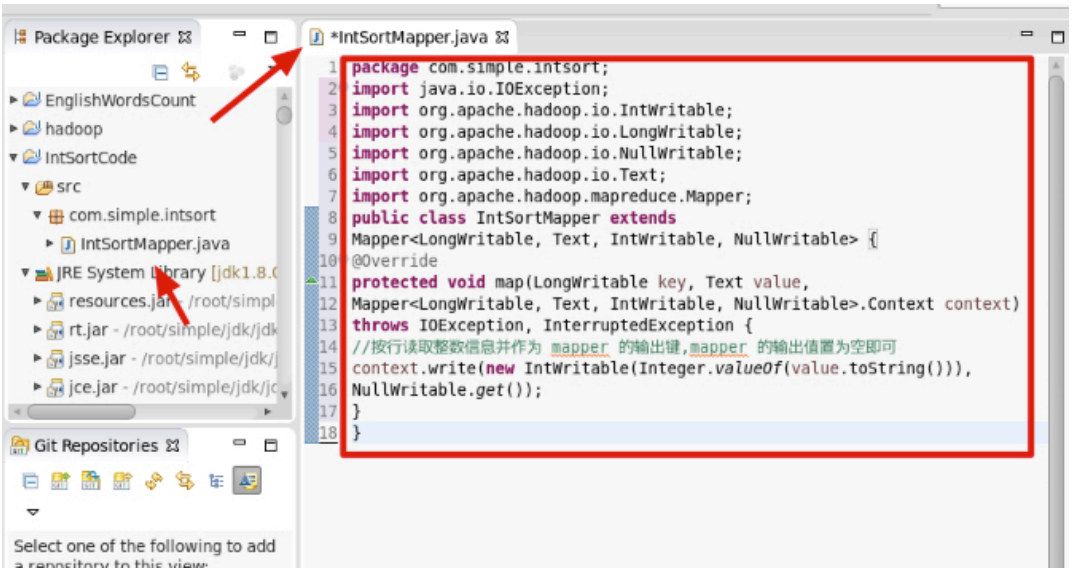
2.3 在编写“IntSortMapper”类之前需要把 hadoop 相关的 jar 包导入，首先右击项目选择“New”—“Folder”创建一个 lib 文件夹并把指定位置中(桌面 lib 文件夹)的包放入该文件中。



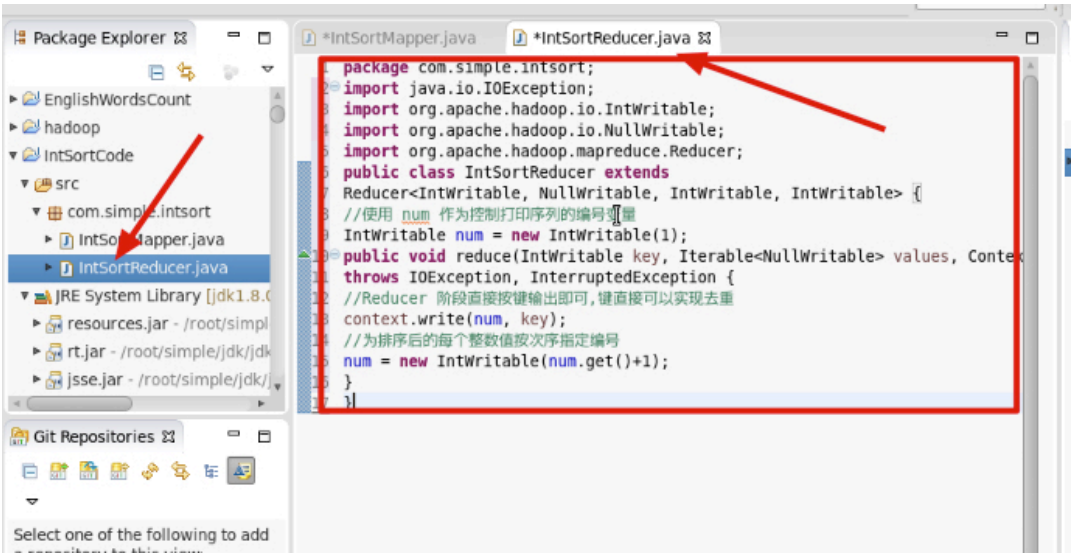
2.4 把 lib 下所有的 jar 包导入到环境变量，首先全选 lib 文件夹下的 jar 包文件，右键点击，选择“build path”-->“add to build path”,添加后，发现在项目下多一个列表项“Referenced Libraries”。如图 7 所示



2.5 让类“IntSortMapper”继承类 Mapper 同时指定需要的参数类型，根据业务逻辑修改 map 类的内容如下。

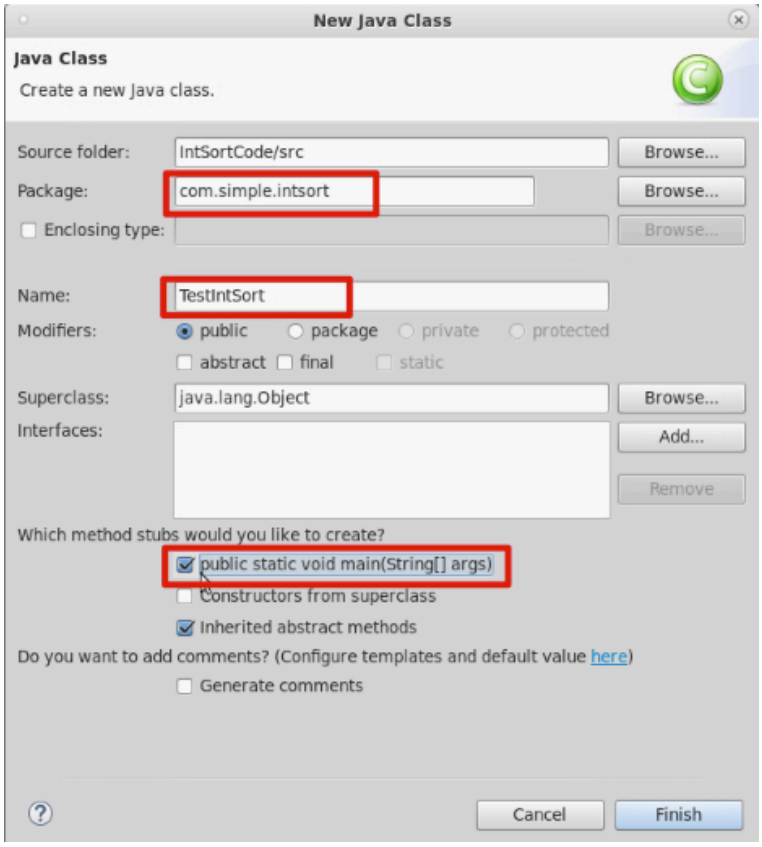


2.6 在项目 src 目录下指定的包名” com.simple.intsort”下右键点击，新建一个类名为“IntSortReducer “并继承 Reducer 类，然后添加该类中的代码内容如下所示。

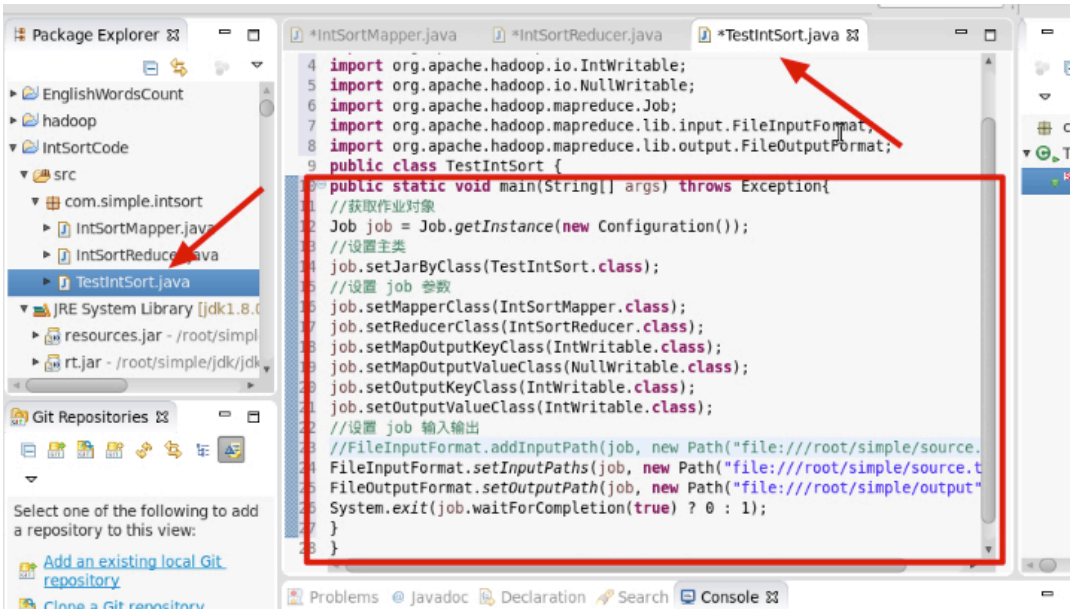


2.7 在项目 src 目录下指定的包名” com.simple.intsort”下右键点击，新

建一个测试主类名为”TestIntSort”并指定 main 主方法。



2.8 测试代码如下所示。



2.9 按照以上的步骤，把 mapper 和 reducer 阶段以及测试代码编写完毕之后，选中测试类”TestIntSort “，右键点击选择”Run as”--->”Java Application”，查看控制台显示内容查看是否正确执行。

```
e.Job (Job.java:monitorAndPrintJob(1355)) - Job job_local419050282_0001 running in uber mode
e.Job (Job.java:monitorAndPrintJob(1362)) - map 100% reduce 100%
e.Job (Job.java:monitorAndPrintJob(1373)) - Job job_local419050282_0001 completed successfull
e.Job (Job.java:monitorAndPrintJob(1380)) - Counters: 33
```

2.10 程序执行完毕之后，可以到输出信息目录/simple/output 下，执行查看命令:cat part-r-00000 查看对数据处理后产生的结果。

```
(base) [root@ferry ~]# cd ~/simple
(base) [root@ferry simple]# cd output
(base) [root@ferry output]# ls
part-r-00000  _SUCCESS
(base) [root@ferry output]# cat part-r-00000
1	2
2	6
3	15
4	22
5	26
6	32
7	54
8	92
9	650
10	654
11	756
12	5956
13	65223
(base) [root@ferry output]#
```

【案例 4.2】mapreduce 两列数据的二次排序

一、项目准备阶段

需求：

对两列数据进行排序，第一列相同的比较第二列

原始数据：

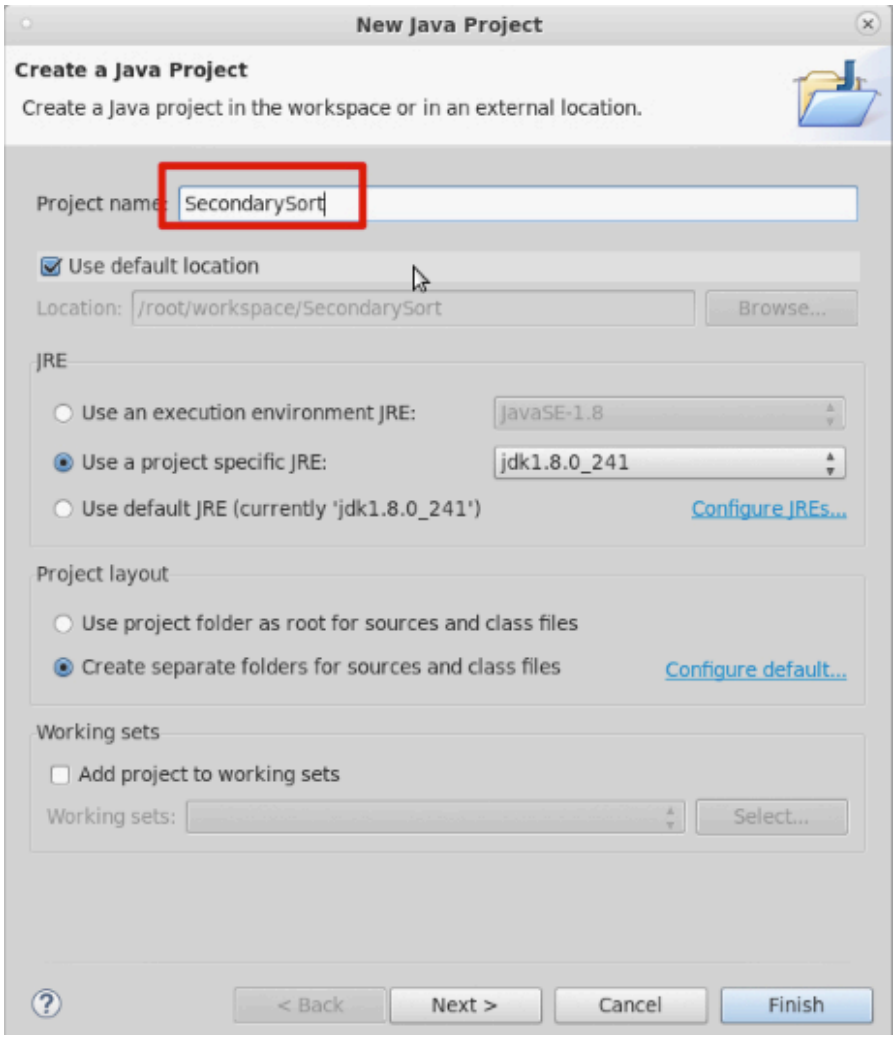
20 21
50 51
50 52
50 53
50 54
60 51
60 53
60 52
60 56
60 57
70 58
60 61
70 54
70 55
70 56
70 57

70 58
1 2
3 4
5 6
7 82
203 21
50 512
50 522
50 53
530 54
40 511
20 53
20 522
60 56
60 57
740 58
63 61
730 54
71 55
71 56
73 57
74 58
12 211
31 42
50 62
7 8

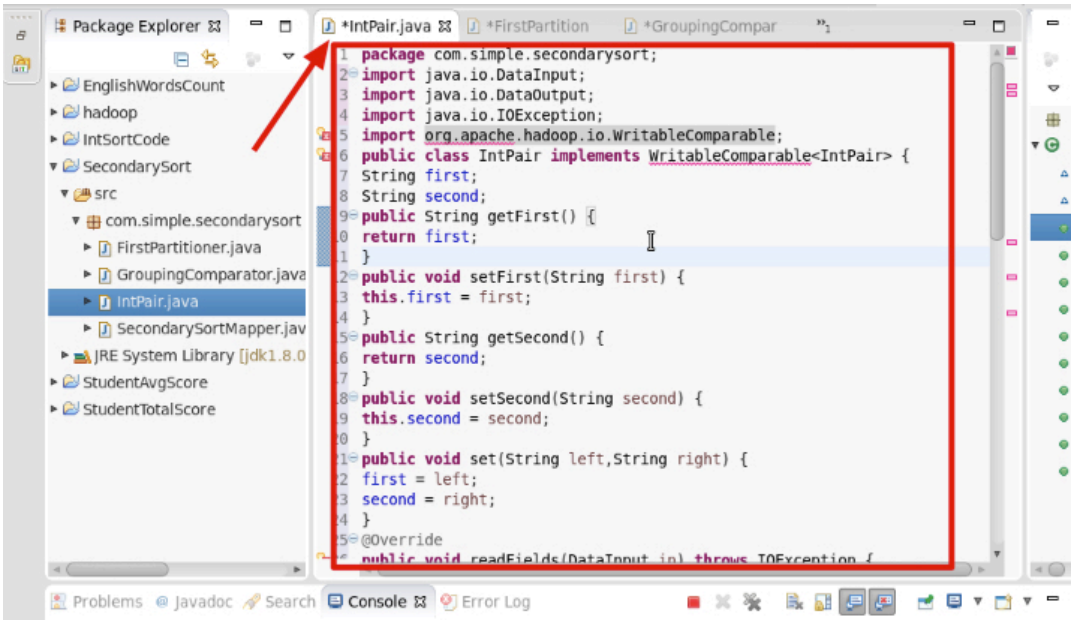
二、程序编写

2.1 在 eclipse 中的项目列表中，右键点击，选择“new“—>”Java

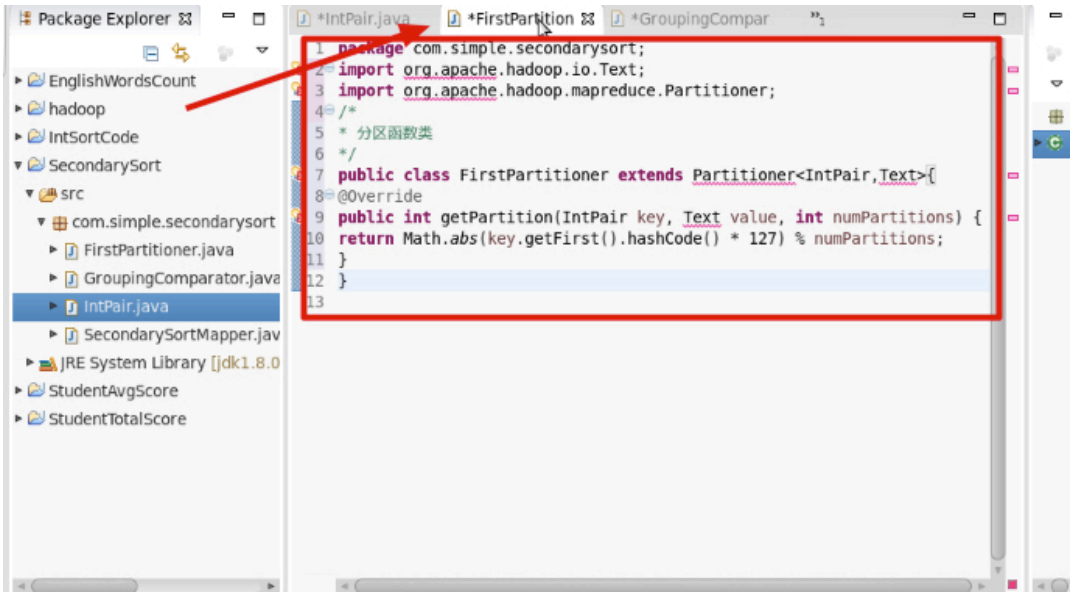
Project…”新建一个项目“SecondarySort” 。



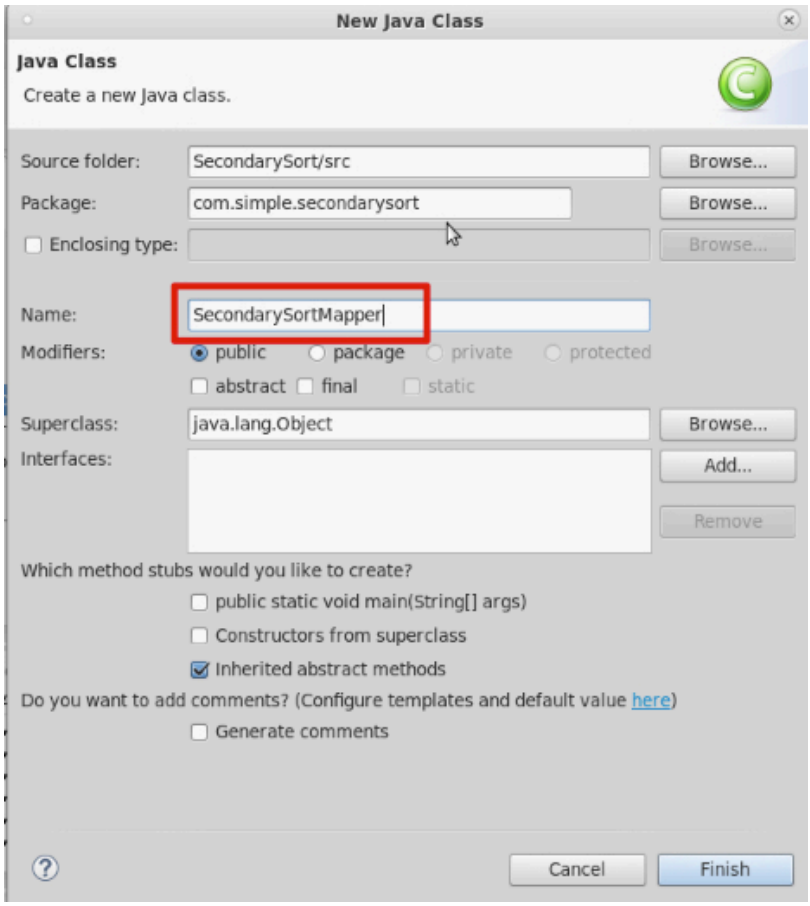
2.2 在项目 src 目录下，右键点击，选择“新建”创建一个类文件名称为“IntPair”并指定包名” com.simple.secondarysort”,该类是对给定数据的两列值的封装，并作为 mapper 的输出键对象 。实现代码如下：



2.3 在项目 src 目录下，右键点击包名” com.simple.secondarysort”选择”New”—“Class”创建一个类文件名称为“FirstPartitioner”,该类是对数据处理后的结果进行分区设置 。代码实现如下：



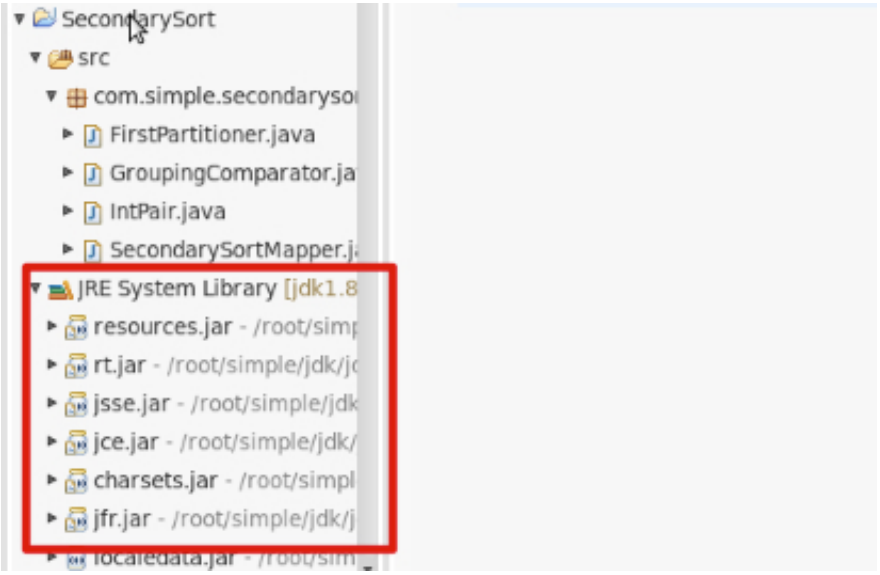
2.5 在项目 src 目录下，右键点击包名” com.simple.secondarysort”选择”New”—“Class”创建一个类文件名称为“SecondarySortMapper”。



2.6 在编写“SecondarySort Mapper”类之前需要把 hadoop 相关的 jar 包导入，首先右击项目选择“New”—“Folder”创建一个 lib 文件夹并把指定

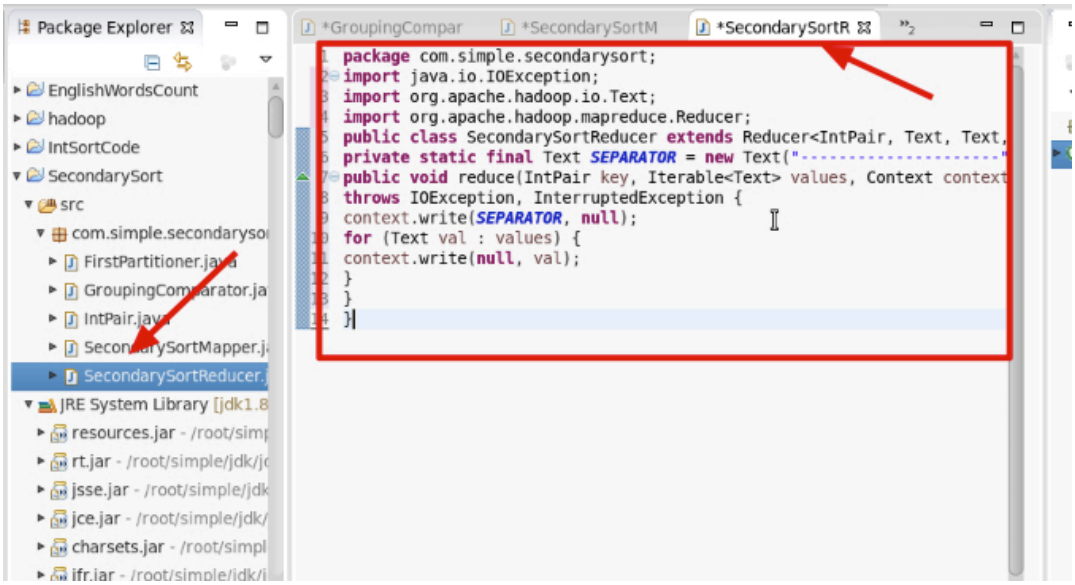
位置中(桌面 lib 文件夹)的包放入该文件中。

2.7 把 lib 下所有的 jar 包导入到环境变量，首先全选 lib 文件夹下的 jar 包文件，右键点击，选择“build path”-->“add to build path”,添加后，发现在项目下很多奶瓶图表的 jar 包。

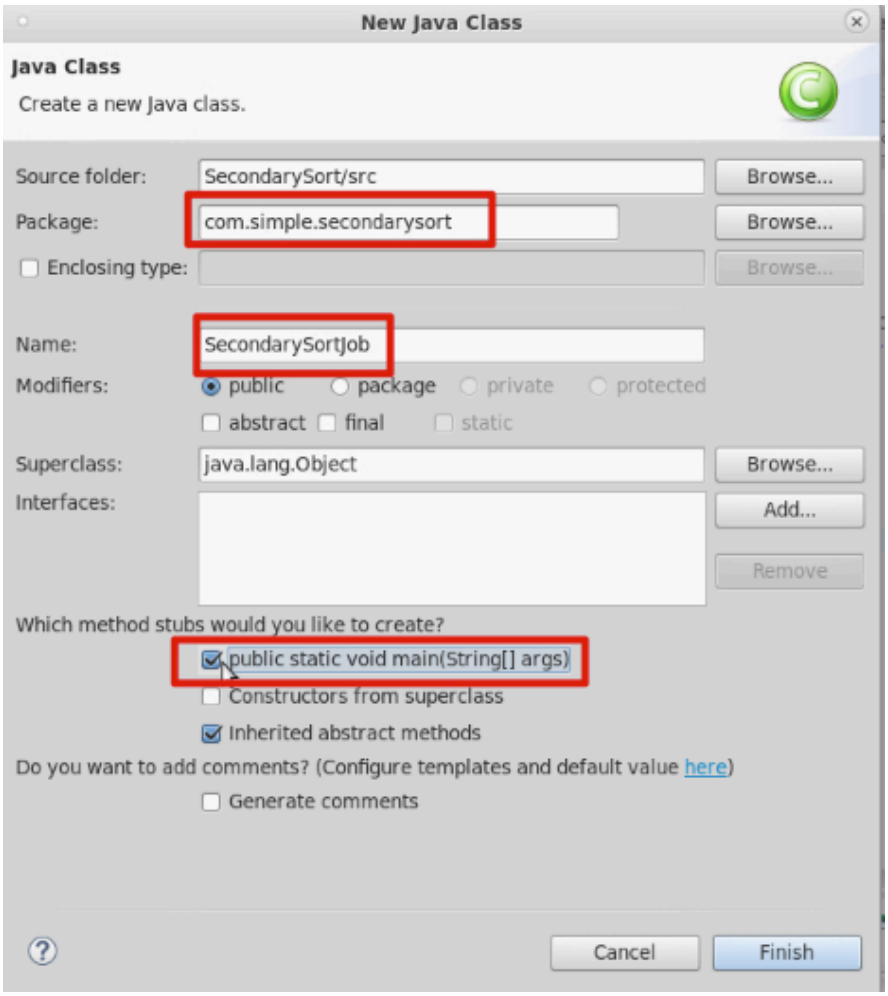


2.8 让类“SecondarySort Mapper”继承类 Mapper 同时指定需要的参数类型，根据业务逻辑修改 map 类的内容如下。

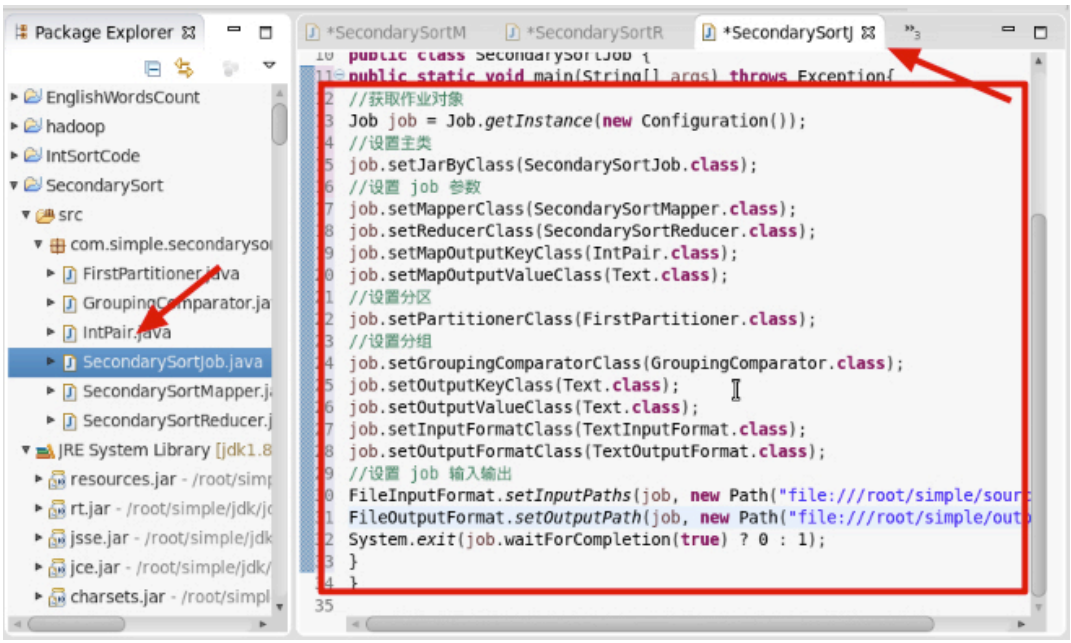
2.9 在项目 src 目录下指定的包名” com.simple.secondarysort”下右键点击，新建一个类名为“SecondarySortReducer “并继承 Reducer 类，然后添加该类中的代码内容如下所示。



2.10 在项目 src 目录下指定的包名” com.simple.secondarysort”下右键点击，新建一个测试主类名为” SecondarySort Job”并指定 main 主方法。



2.11 测试代码如下所示。

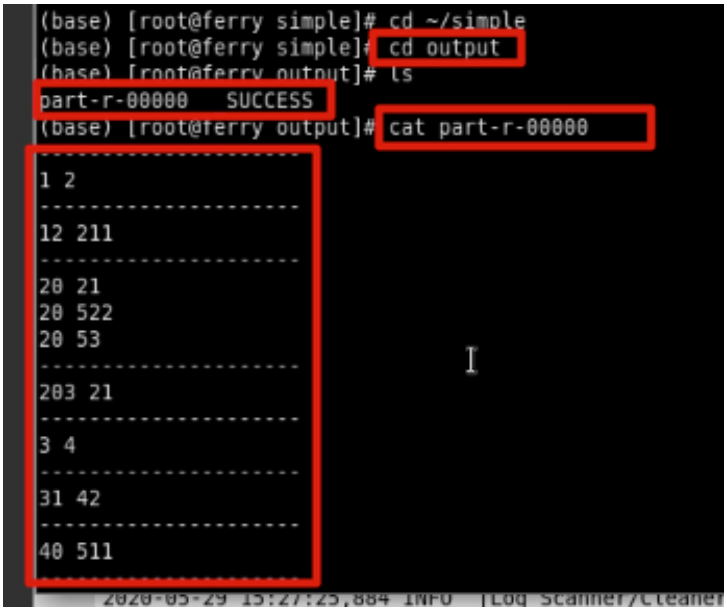


2.12 按照以上的步骤，把 mapper 和 reducer 阶段以及测试代码编写完毕之后，选中测试类” SecondarySort Job “，右键点击选择”Run as”--->”Java Application”,查看控制台显示内容查看是否正确执行。

```
.Job (Job.java:monitorAndPrintJob(1355)) - Job job_local174032281_0001 running in
.Job (Job.java:monitorAndPrintJob(1362)) - map 100% reduce 100%
.Job (Job.java:monitorAndPrintJob(1373)) - Job job_local174032281_0001 completed s
.Job (Job.java:monitorAndPrintJob(1380)) - Counters: 33

8
441440
s=0
rations=0
```

2.13 程序执行完毕之后，可以到输出信息目录/simple/output 下，执行查看命令:cat part-r-00000 查看对数据处理后产生的结果。



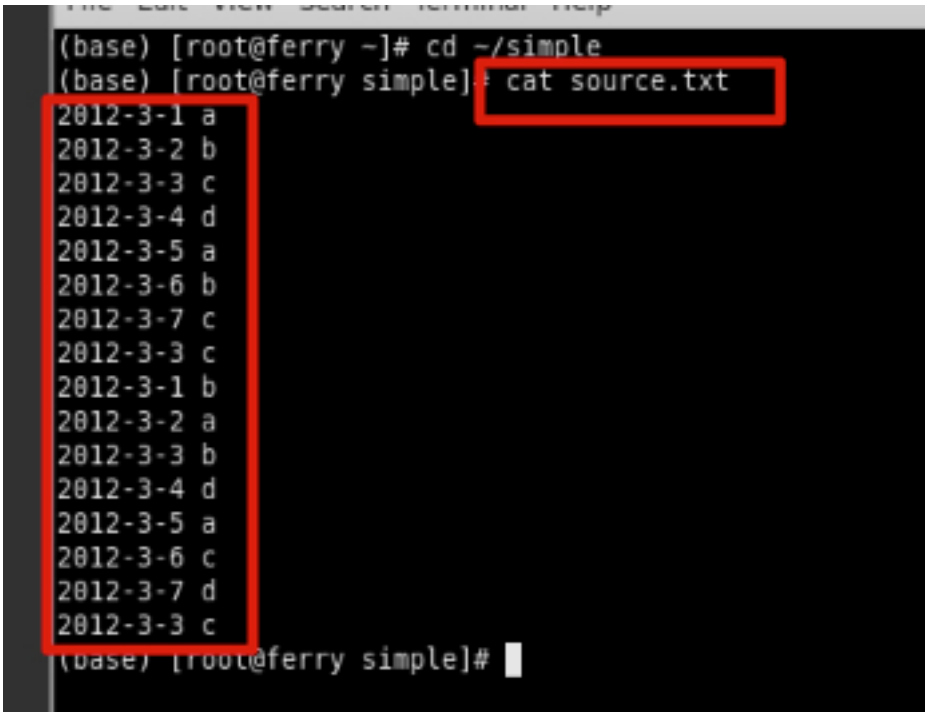
【案例 4.3】mapreduce 多条数据去重处理

一、项目准备阶段

把指定的数据信息以单条记录的方式保存在文本文件 source.txt 中并存放到指定的位置，该位置可以是本地，也可以是 hdfs 分布式系统。

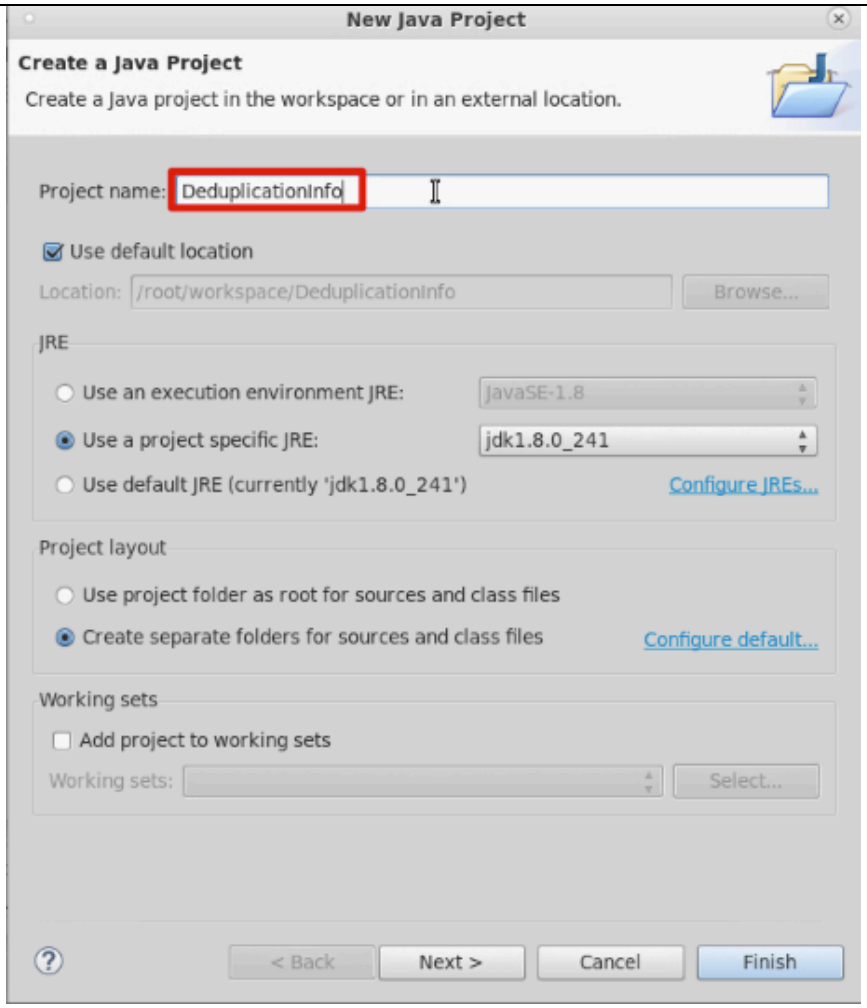
原始数据

2012-3-1 a
2012-3-2 b
2012-3-3 c
2012-3-4 d
2012-3-5 a
2012-3-6 b
2012-3-7 c
2012-3-3 c
2012-3-1 b
2012-3-2 a
2012-3-3 b
2012-3-4 d
2012-3-5 a
2012-3-6 c
2012-3-7 d
2012-3-3 c

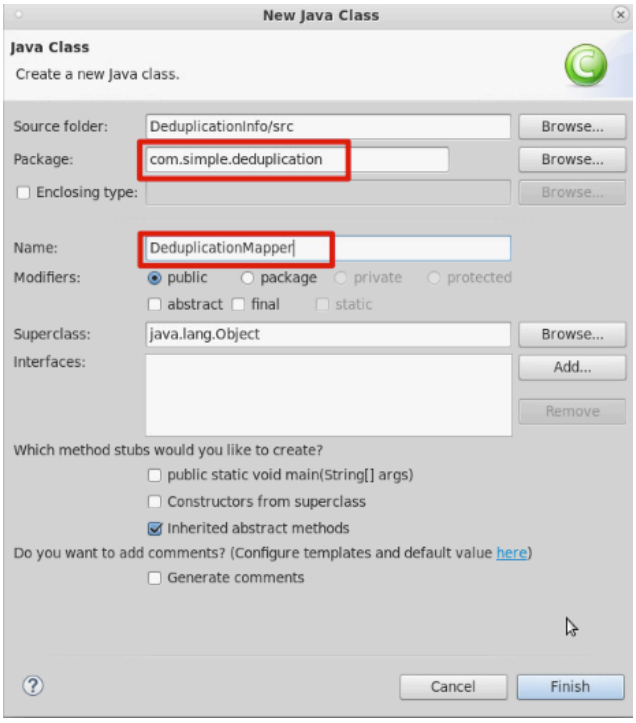


二、程序编写

2.1 在 eclipse 中的项目列表中，右键点击，选择“new“—>”Java Project…”新建一个项目“DeduplicationInfo” 。



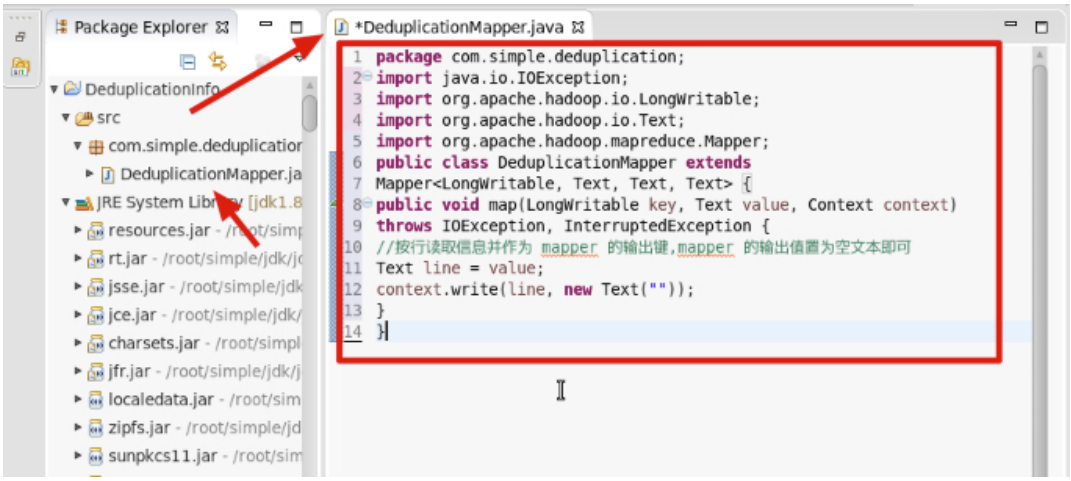
2.2 在项目 src 目录下，右键点击，选择“新建”创建一个类文件名称为“DeduplicationMapper”并指定包名” com.simple.deduplication” 。



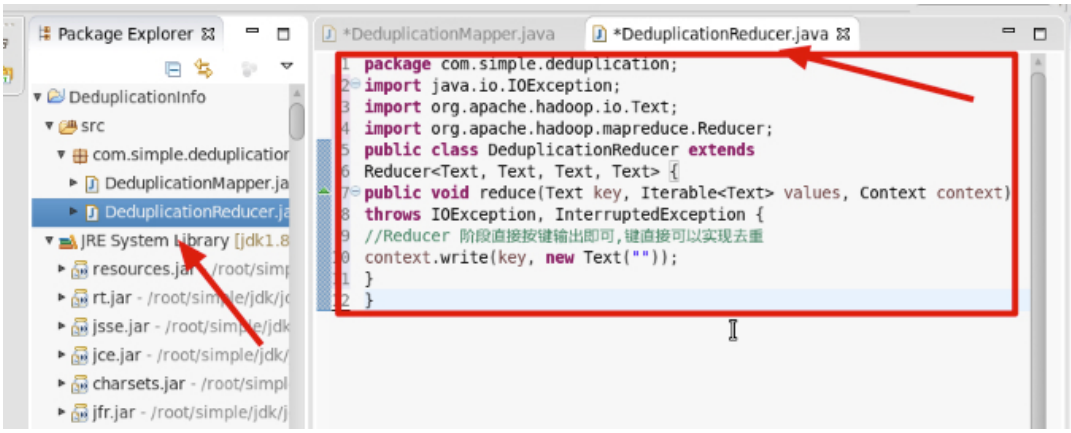
2.3 在编写“DeduplicationMapper”类之前需要把 hadoop 相关的 jar 包导入，首先右击项目选择“New”—“Folder”创建一个 lib 文件夹并把指定位置中(桌面 lib 文件夹)的包放入该文件中。

2.4 把 lib 下所有的 jar 包导入到环境变量，首先全选 lib 文件夹下的 jar 包文件，右键点击，选择“build path”-->“add to build path”,添加后，发现在项目下多一个列表项“Referenced Libraries”。

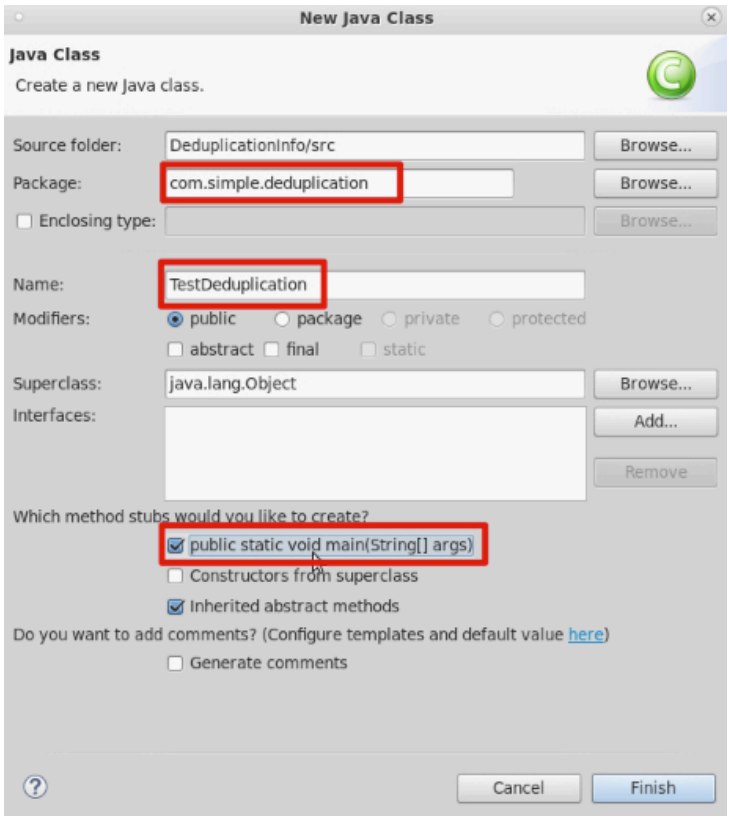
2.5 让类“DeduplicationMapper”继承类 Mapper 同时指定需要的参数类型，根据业务逻辑修改 map 类的内容如下。



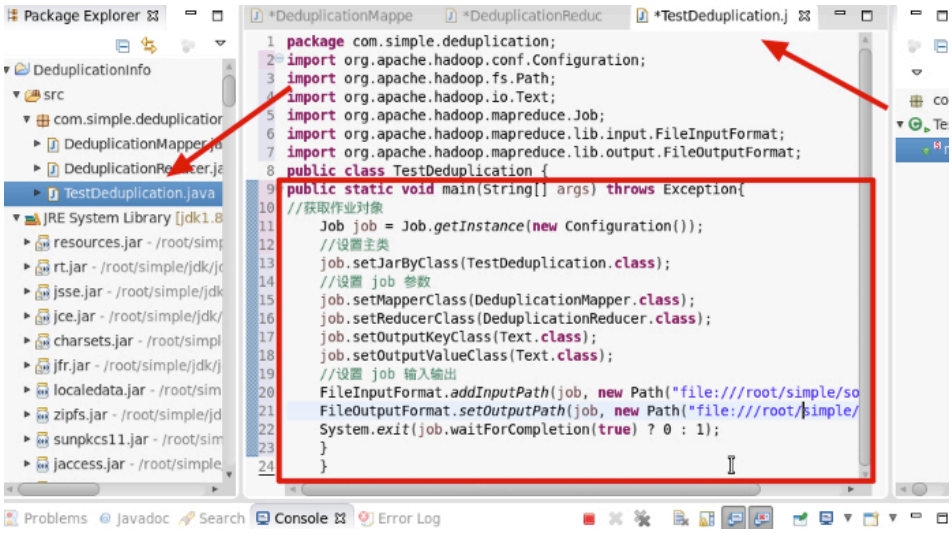
2.6 在项目 src 目录下指定的包名” com.simple.deduplication”下右键点击，新建一个类名为“DeduplicationReducer “并继承 Reducer 类，然后添加该类中的代码内容如下所示。



2.7 在项目 src 目录下指定的包名” com.simple.deduplication”下右键点击，新建一个测试主类名为“TestDeduplication “并指定 main 主方法。



2.8 测试代码如下所示。



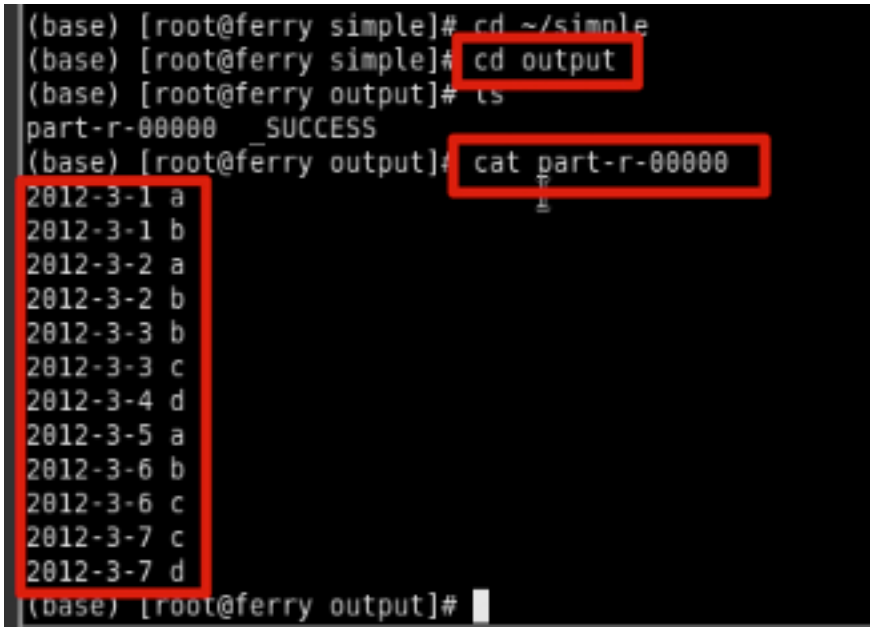
2.9 按照以上的步骤，把 mapper 和 reducer 阶段以及测试代码编写完毕之后，选中测试类“TestDeduplication “，右键点击选择”Run as”--->”Java Application”，查看控制台显示内容查看是否正确执行。

```
[main] mapreduce.Job (Job.java:monitorAndPrintJob(1355)) - Job job_local460714404_0001 running in ub
[main] mapreduce.Job (Job.java:monitorAndPrintJob(1362)) - map 100% reduce 100%
[main] mapreduce.Job (Job.java:monitorAndPrintJob(1373)) - Job job_local460714404_0001 completed suc
[main] mapreduce.Job (Job.java:monitorAndPrintJob(1380)) - Counters: 33

of bytes read=1140
of bytes written=434486
of read operations=0
of large read operations=0
of write operations=0

ords=16
```

2.10 程序执行完毕之后，可以到输出信息目录下，执行查看命令 cat part-r-00000 查看对数据处理后产生的结果。



【案例 4.4】mapreduce 非结构化日志文件处理

一、项目准备阶段

需求：根据 tomcat 日志计算 url 访问了情况，具体的 url 如下，

要求：区别统计 GET 和 POST URL 访问量

结果为：访问方式、URL、访问量

原始数据：

127.0.0.1 - - [03/Jul/2014:23:53:32 +0800] "POST /service/addViewTimes_544.htm HTTP/1.0" 200 2 0.004

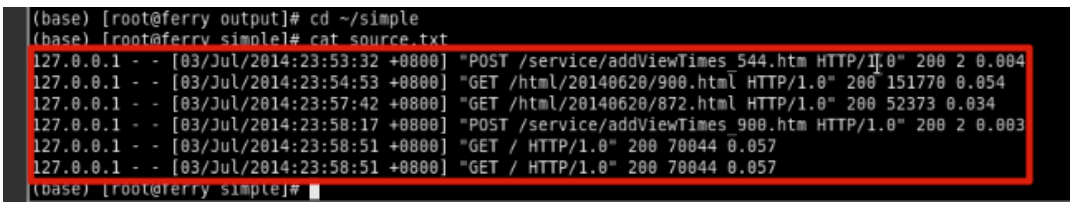
127.0.0.1 - - [03/Jul/2014:23:54:53 +0800] "GET /html/20140620/900.html HTTP/1.0" 200 151770 0.054

127.0.0.1 - - [03/Jul/2014:23:57:42 +0800] "GET /html/20140620/872.html HTTP/1.0" 200 52373 0.034

127.0.0.1 - - [03/Jul/2014:23:58:17 +0800] "POST /service/addViewTimes_900.htm HTTP/1.0" 200 2 0.003

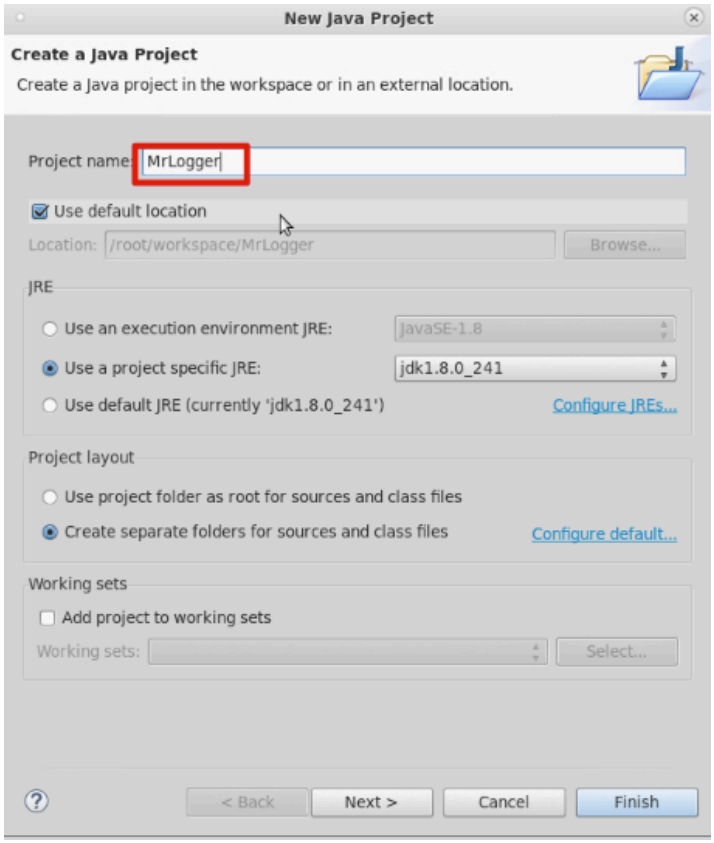
127.0.0.1 - - [03/Jul/2014:23:58:51 +0800] "GET / HTTP/1.0" 200 70044 0.057

127.0.0.1 - - [03/Jul/2014:23:58:51 +0800] "GET / HTTP/1.0" 200 70044 0.057

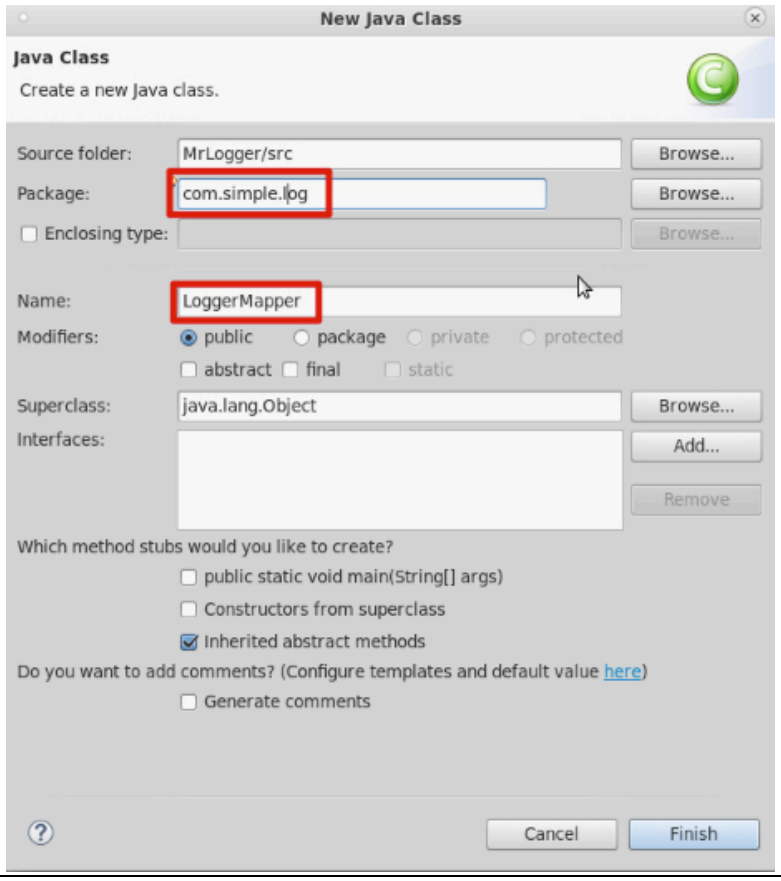


二、程序编写

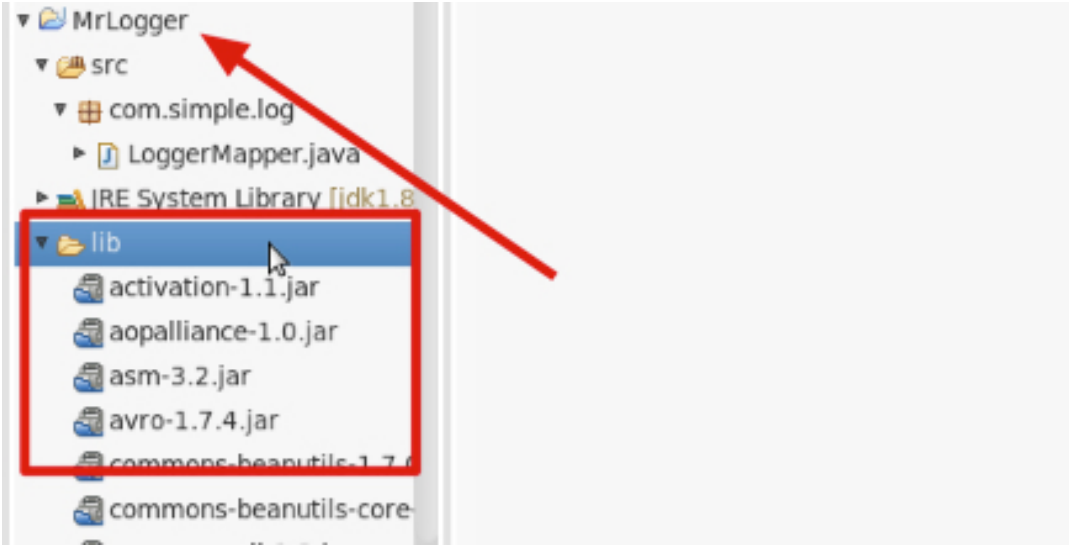
2.1 在 eclipse 中的项目列表中，右键点击，选择“new“—>”Java Project…”新建一个项目“MrLogger” 。



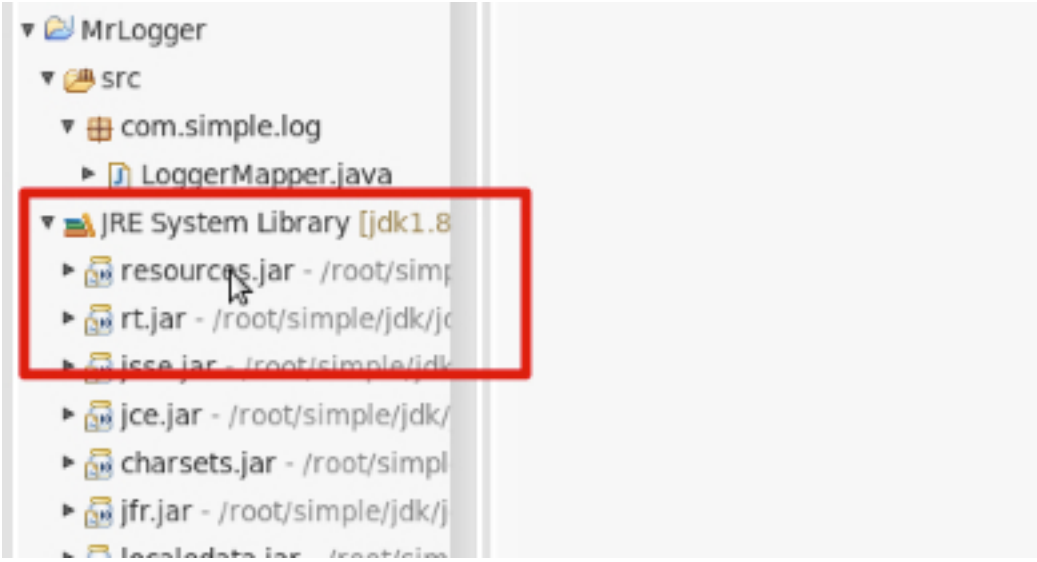
2.2 在项目 src 目录下，右键点击，选择“新建”创建一个类文件名称为“LoggerMapper”并指定包名” com.simple.log” 。



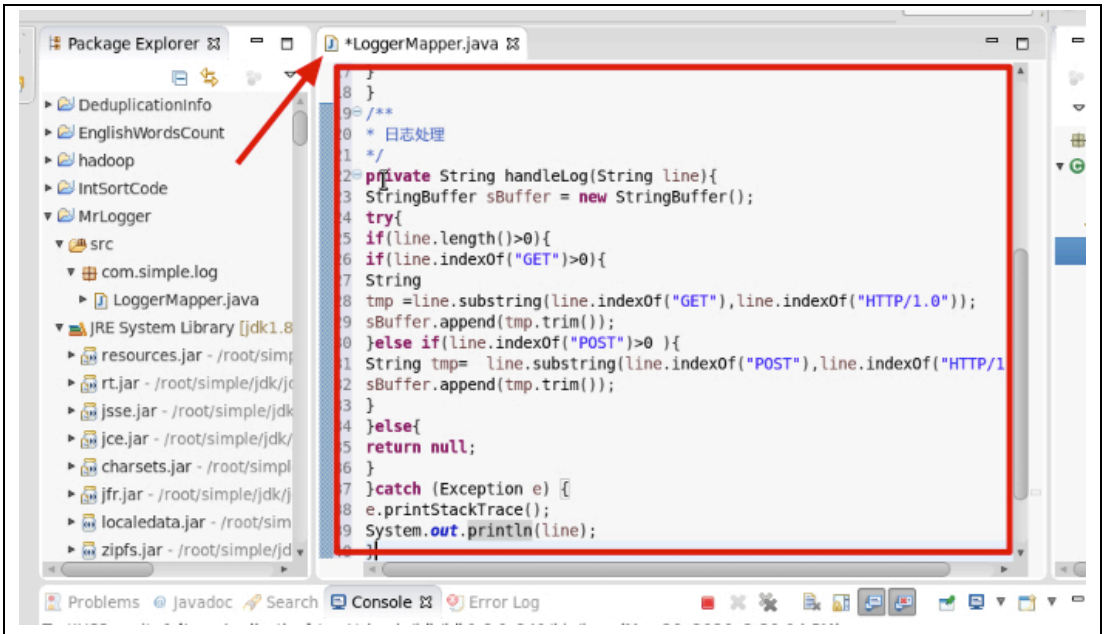
2.3 在编写“LoggerMapper”类之前需要把 hadoop 相关的 jar 包导入，首先右击项目选择“New”—“Folder”创建一个 lib 文件夹并把指定位置中(桌面 lib 文件夹)的包放入该文件中。



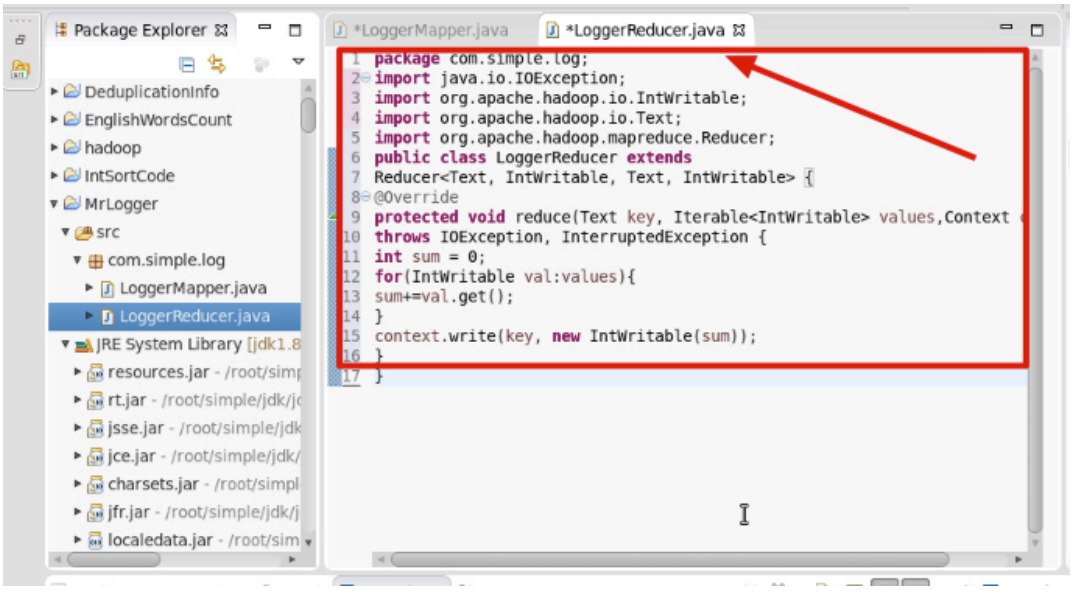
2.4 把 lib 下所有的 jar 包导入到环境变量，首先全选 lib 文件夹下的 jar 包文件，右键点击，选择“build path”-->“add to build path”,添加后，发现在项目下很多奶瓶图表的 jar 包。



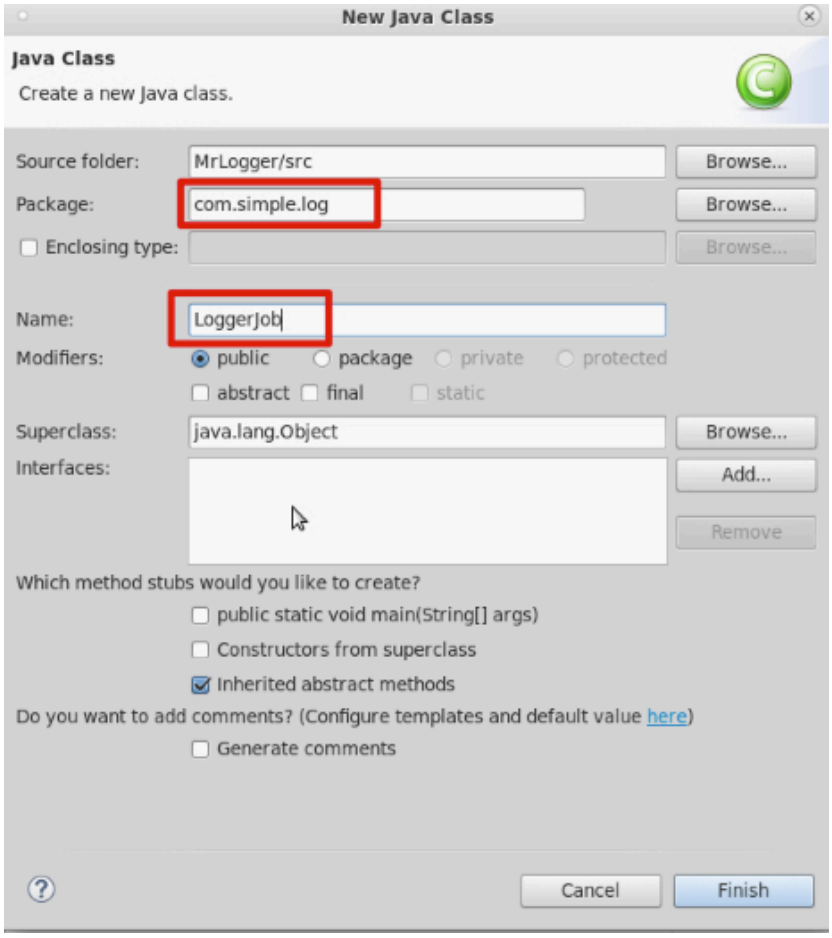
2.5 让类“LoggerMapper”继承类 Mapper 同时指定需要的参数类型，根据业务逻辑修改 map 类的内容如下。



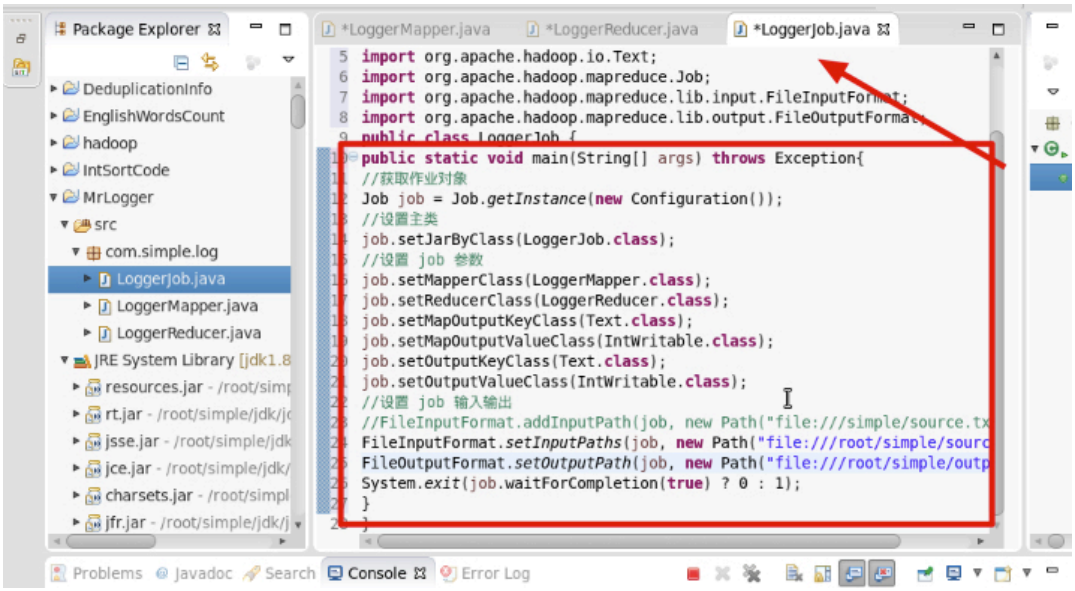
2.6 在项目 src 目录下指定的包名” com.simple.log”下右键点击，新建一个类名为“LoggerReducer “并继承 Reducer 类，然后添加该类中的代码内容如下所示。



2.7 在项目 src 目录下指定的包名” com.simple.log”下右键点击，新建一个测试主类名为” LoggerJob”并指定 main 主方法。



2.8 测试代码如下所示。



2.9 按照以上的步骤，把 mapper 和 reducer 阶段以及测试代码编写完毕之后，选中测试类” LoggerJob “，右键点击选择”Run as”--->”Java Application”，查看控制台显示内容查看是否正确执行。

```
pool-3-thread-1] mapred.Task (Task.java:sendDone(1115)) - Task 'attempt_local1312657434_0001_r_0000
pool-3-thread-1] mapred.LocalJobRunner (LocalJobRunner.java:run(325)) - Finishing task: attempt_loc
Thread-11] mapred.LocalJobRunner (LocalJobRunner.java:runTasks(456)) - reduce task executor complet
main] mapreduce.Job (Job.java:monitorAndPrintJob(1355)) - Job job_local1312657434_0001 running in u
main] mapreduce.Job (Job.java:monitorAndPrintJob(1362)) -  map 100% reduce 100%
main] mapreduce.Job (Job.java:monitorAndPrintJob(1373)) - Job job_local1312657434_0001 completed su
main] mapreduce.Job (Job.java:monitorAndPrintJob(1380)) - Counters: 33

 bytes read=1792
 bytes written=438128
 read operations=0
 large read operations=0
 write operations=0
```

2.10 程序执行完毕之后，可以到输出信息目录/simple/output 下，执行查看命令:cat part-r-00000 查看对数据处理后产生的结果。

```
(base) [root@ferry simple]# cd output
(base) [root@ferry output]# ls
part-r-00000  _SUCCESS
(base) [root@ferry output]# cat part-r-00000
GET /   2
GET /html/20140620/872.html     1
GET /html/20140620/900.html     1
POST /service/addViewTimes_544.htm      1
POST /service/addViewTimes_900.htm      1
(base) [root@ferry output]#
```

5.MR4 MR 调优

【案例 5.1】mapreducetopN 排名

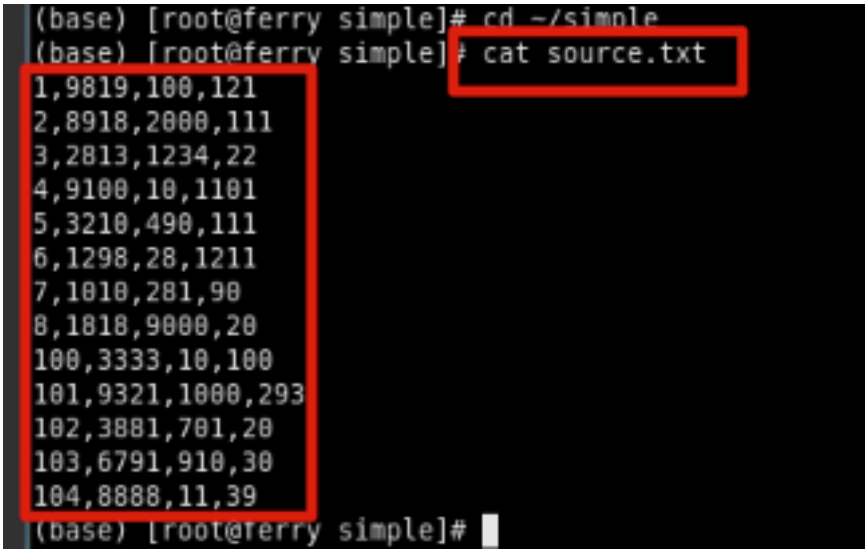
一、项目准备阶段

需求：

第一个字段：orderid 订单 ID　　第二个字段：userid 用户 ID　　第三个字段：payment 付款金额　　第四个字段：productid 产品 ID

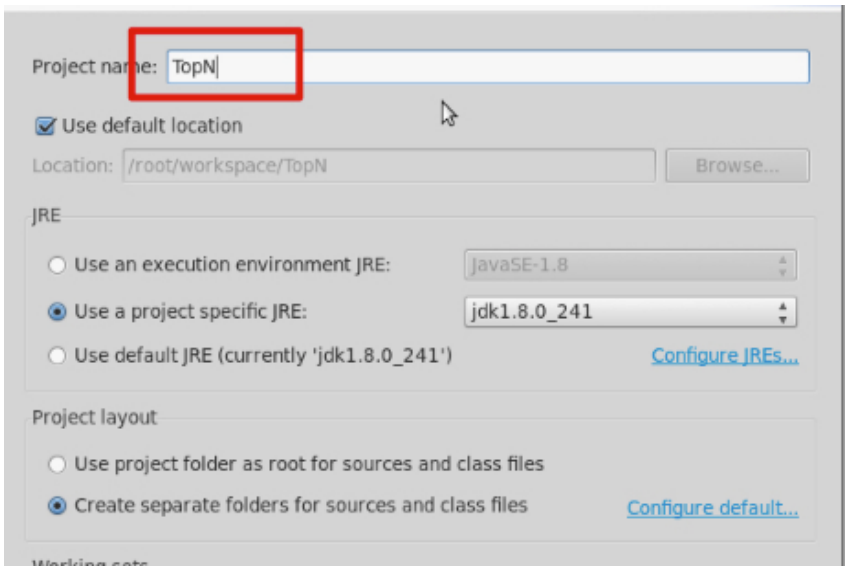
1）付款金额最大的几笔订单

1,9819,100,121
2,8918,2000,111
3,2813,1234,22
4,9100,10,1101
5,3210,490,111
6,1298,28,1211
7,1010,281,90
8,1818,9000,20
100,3333,10,100
101,9321,1000,293
102,3881,701,20
103,6791,910,30
104,8888,11,39

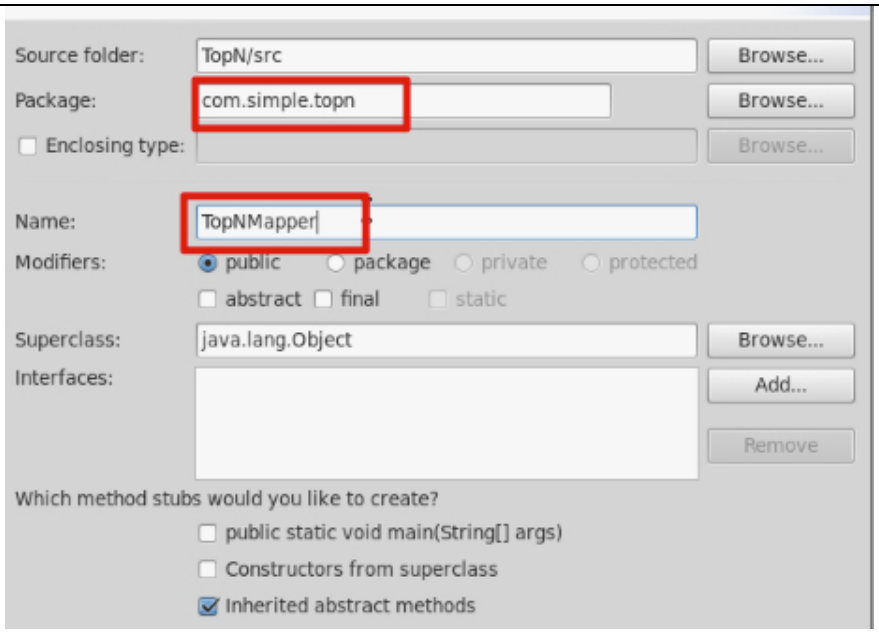


二、程序编写

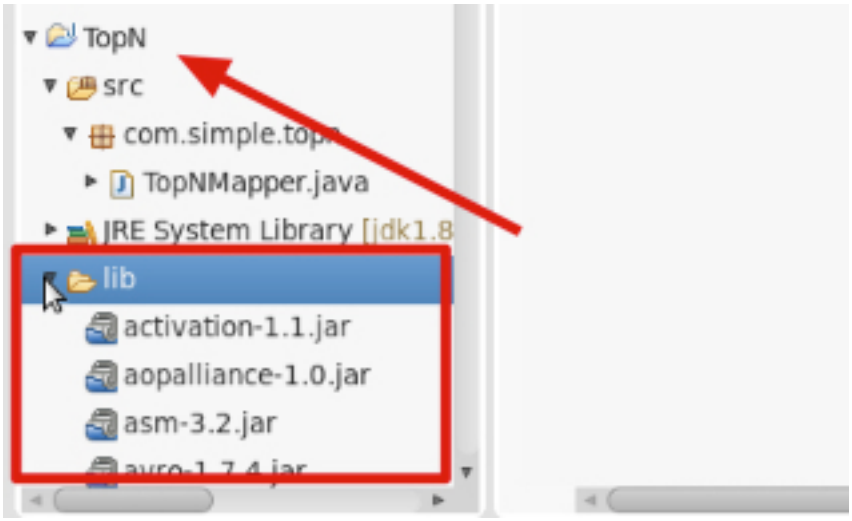
2.1 在 eclipse 中的项目列表中，右键点击，选择“new“—>”Java Project…”新建一个项目“TopN” 。



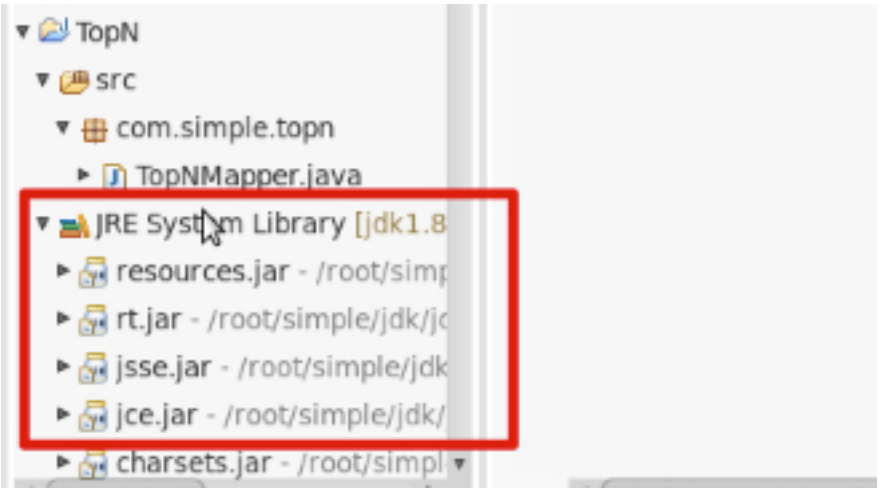
2.2 在项目 src 目录下，右键点击，选择”New”—“Class”创建一个类文件名称为“TopNMapper”并指定包名” com.simple.topn” 。



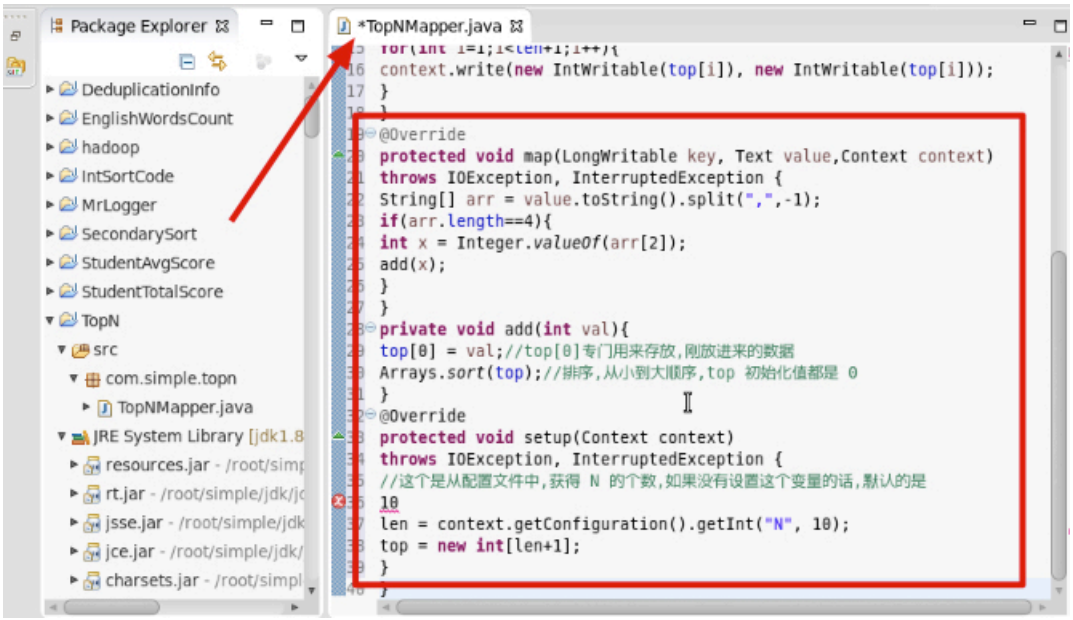
2.3 在编写“TopNMapper”类之前需要把 hadoop 相关的 jar 包导入，首先右击项目选择“New”—“Folder”创建一个 lib 文件夹并把指定位置中(桌面 lib 文件夹)的包放入该文件中。



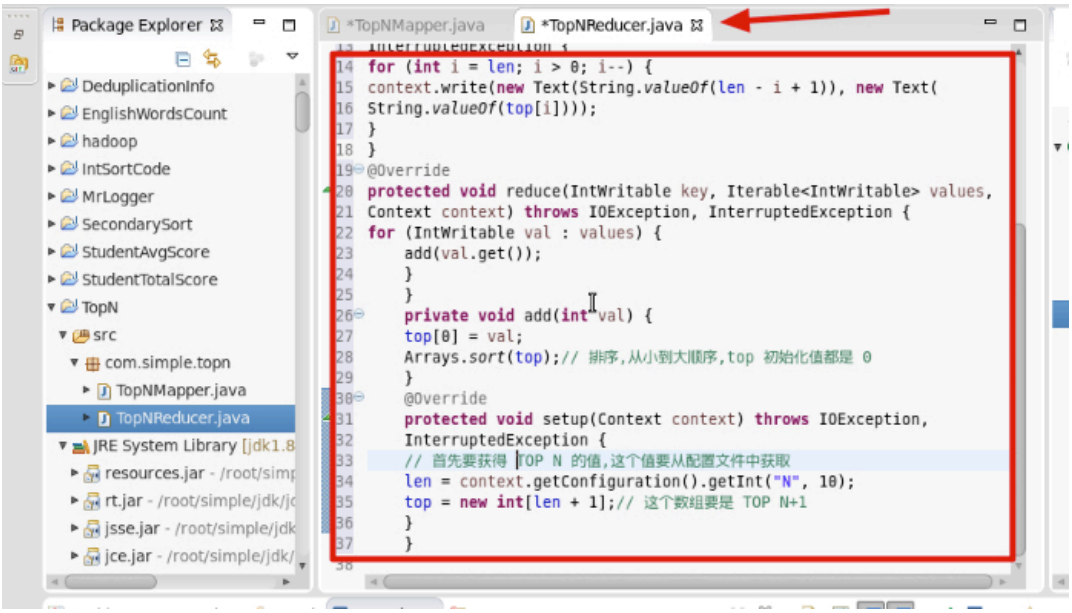
2.4 把 lib 下所有的 jar 包导入到环境变量，首先全选 lib 文件夹下的 jar 包文件，右键点击，选择“build path”-->“add to build path”,添加后，发现在项目下很多奶瓶图表的 jar 包。



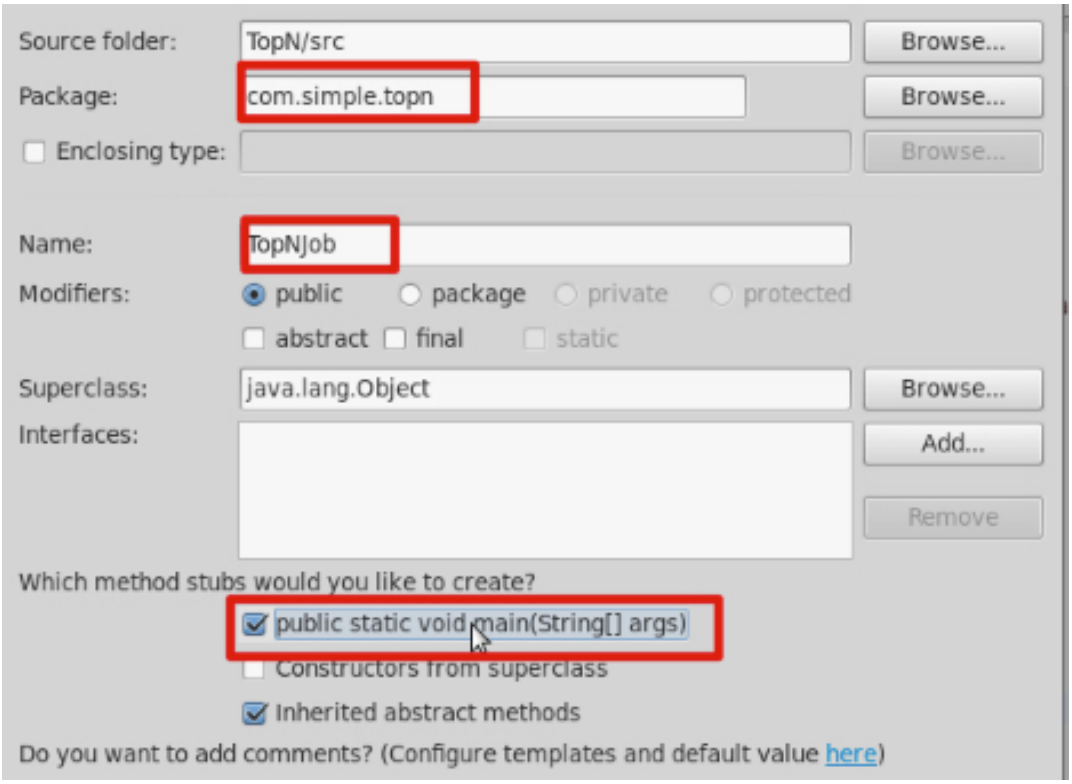
2.5 让类“TopNMapper”继承类 Mapper 同时指定需要的参数类型，根据业务逻辑修改 map 类的内容如下。



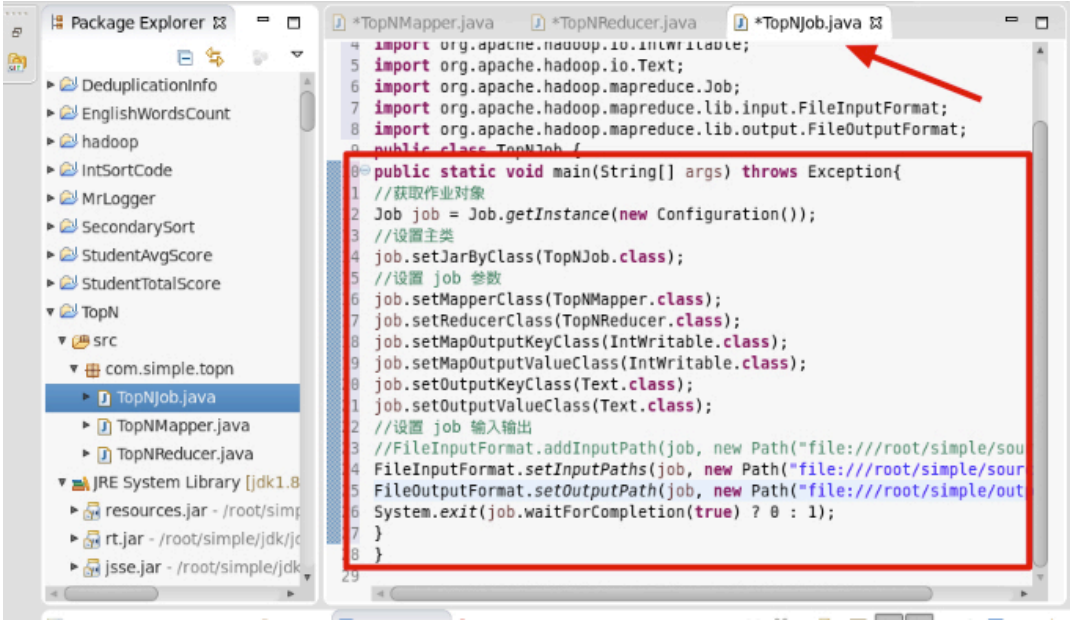
2.6 在项目 src 目录下指定的包名” com.simple.topn”下右键点击，新建一个类名为“TopNReducer “并继承 Reducer 类，然后添加该类中的代码内容如下所示。



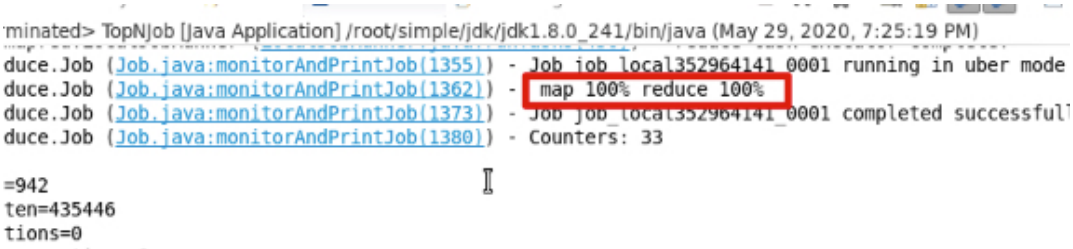
2.7 在项目 src 目录下指定的包名” com.simple.topn”下右键点击，新建一个测试主类名为” TopNJob”并指定 main 主方法。



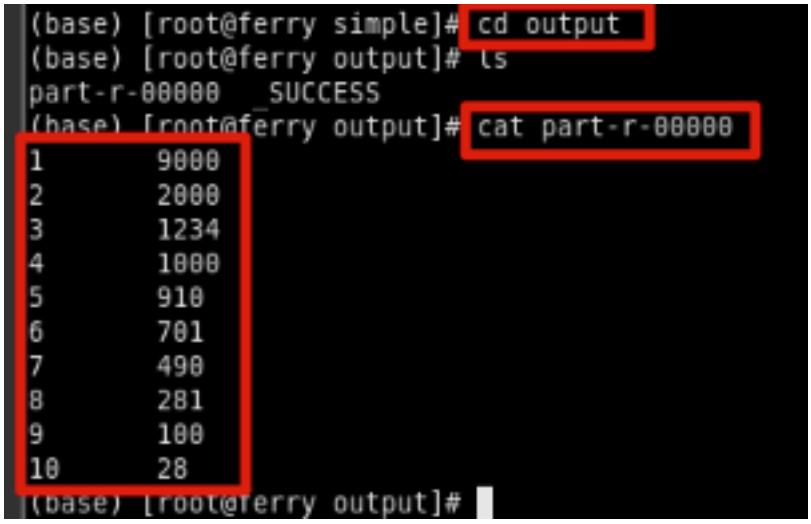
2.8 测试代码如下所示。



2.9 按照以上的步骤，把 mapper 和 reducer 阶段以及测试代码编写完毕之后，选中测试类” TopNJob “，右键点击选择”Run as”--->”Java Application”，查看控制台显示内容查看是否正确执行。



2.10 程序执行完毕之后，可以到输出信息目录/simple/output 下，执行查看命令:cat part-r-00000 查看对数据处理后产生的结果。



【案例 5.2】mapreduce 统计拨打公共服务号码的电话信息

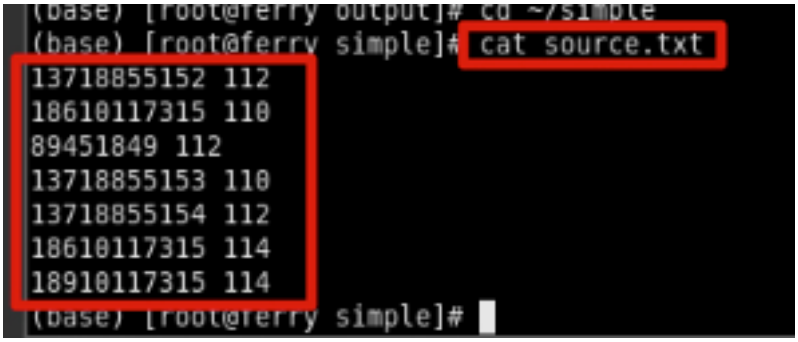
一、项目准备阶段

需求：

对下面原始输出进行处理，把所有拨打同一个公共服务电话的电话号码统计

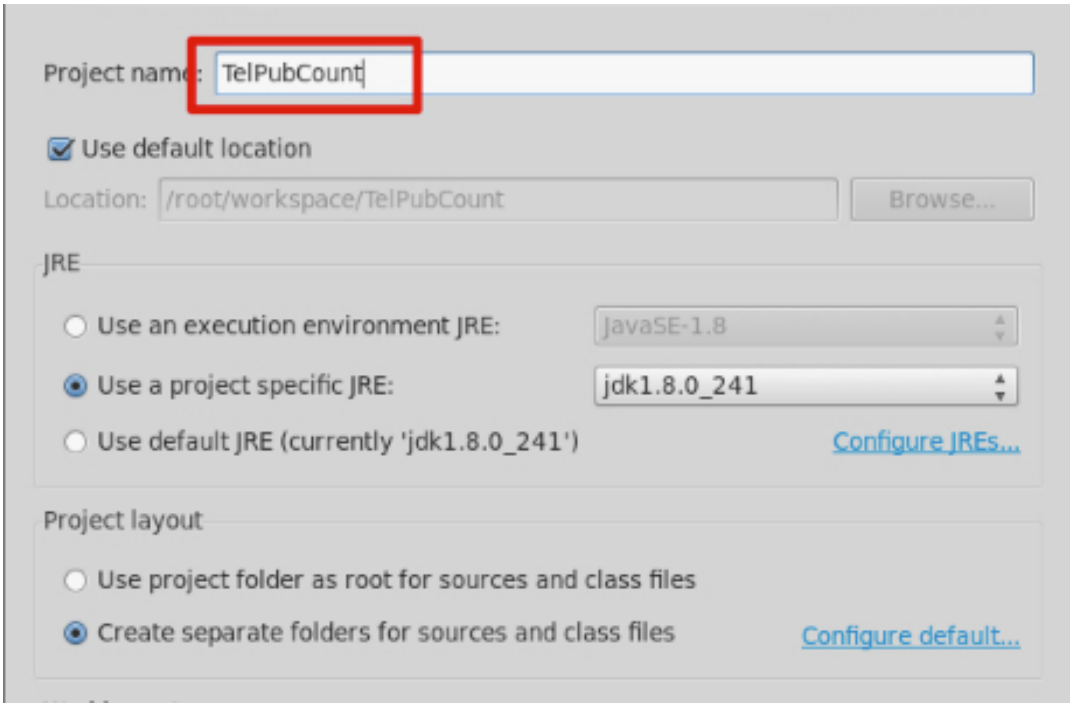
起来，展示为每个公共服务号码对应多个用户号码。

原始数据：
13718855152 112
18610117315 110
89451849 112
13718855153 110
13718855154 112
18610117315 114
18910117315 114

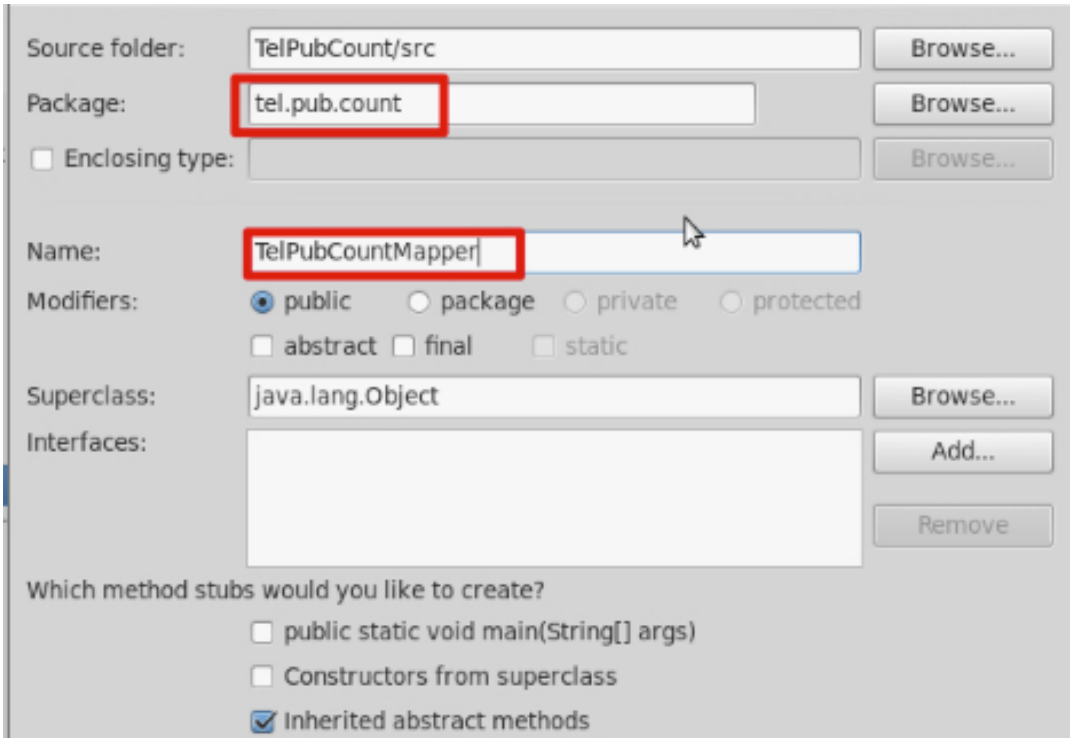


二、程序编写

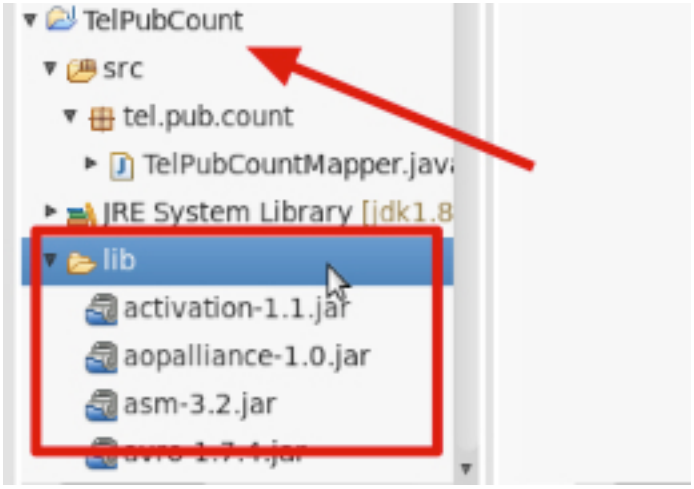
2.1 在 eclipse 中的项目列表中，右键点击，选择“new“—>”Java Project…”新建一个项目“TelPubCount” 。



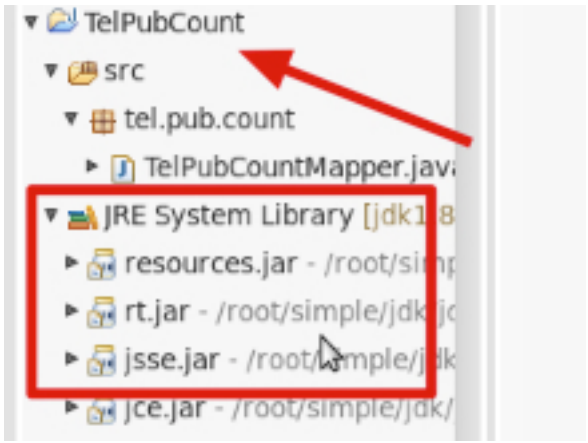
2.2 在项目 src 目录下，右键点击，选择”New”—“Class”创建一个类文件名称为“TelPubCountMapper”并指定包名” tel.pub.count” 。



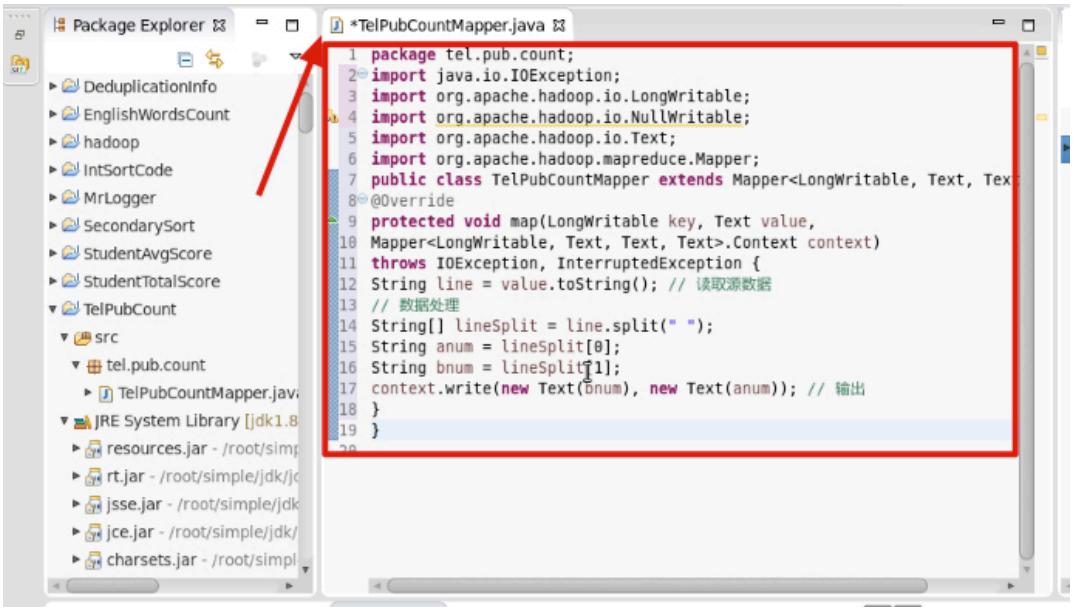
2.3 在编写“TelPubCountMapper”类之前需要把 hadoop 相关的 jar 包导入，首先右击项目选择“New”—“Folder”创建一个 lib 文件夹并把指定位置中(桌面 lib 文件夹)的包放入该文件中。



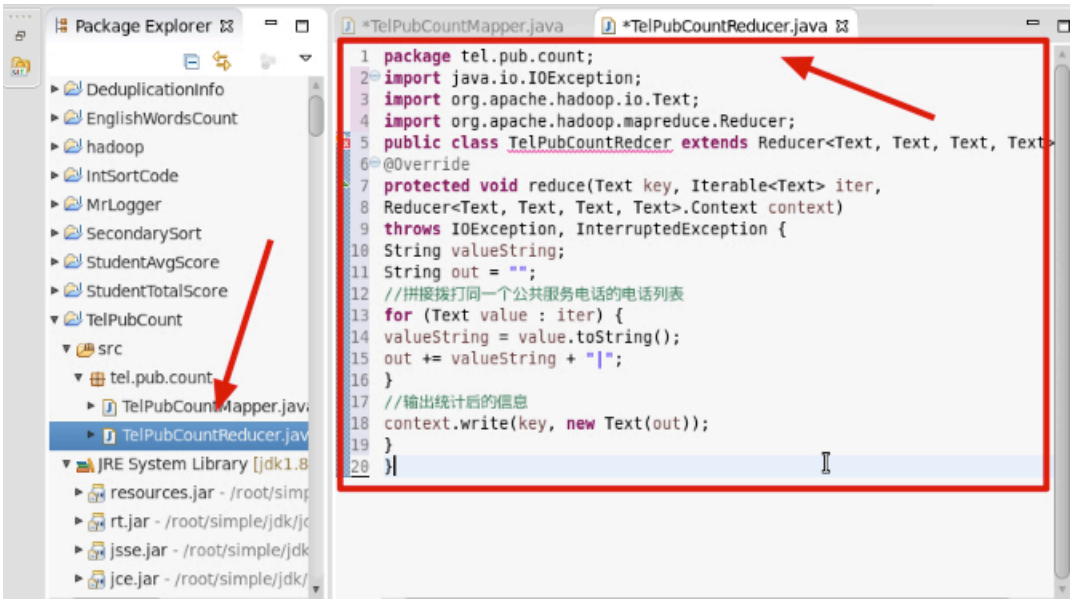
2.4 把 lib 下所有的 jar 包导入到环境变量，首先全选 lib 文件夹下的 jar 包文件，右键点击，选择“build path”-->“add to build path”，添加后，发现在项目下很多奶瓶图表的 jar 包。



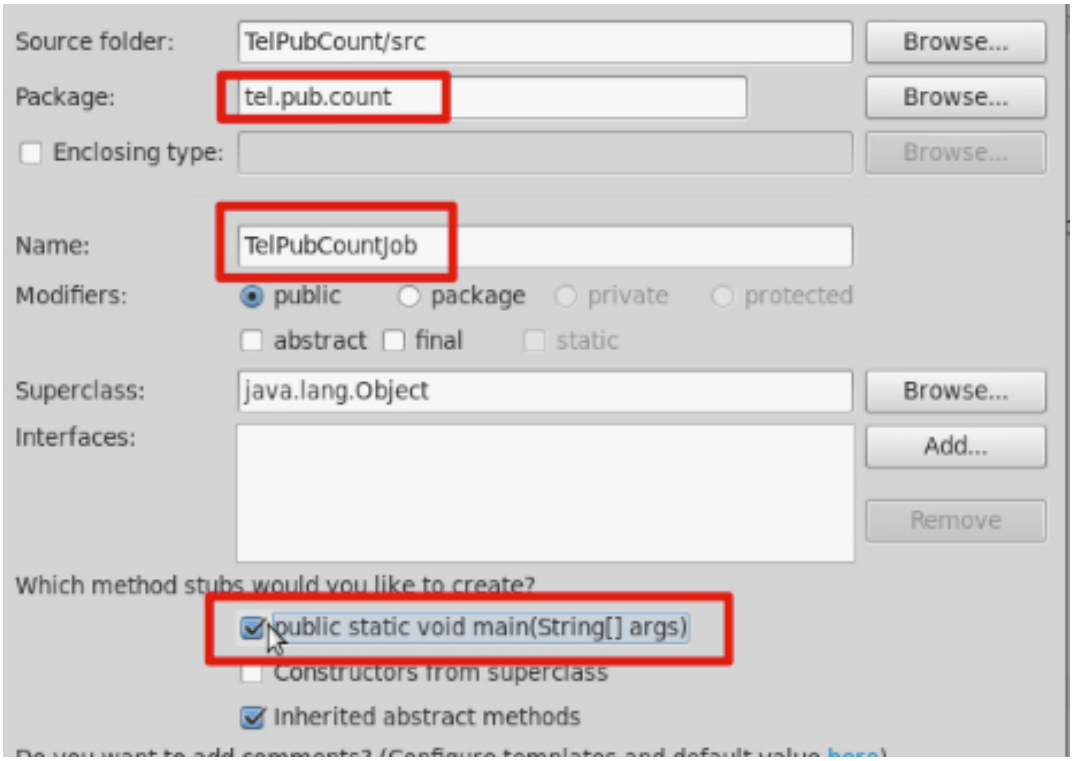
2.5 让类“TelPubCountMapper”继承类 Mapper 同时指定需要的参数类型，根据业务逻辑修改 map 类的内容如下。



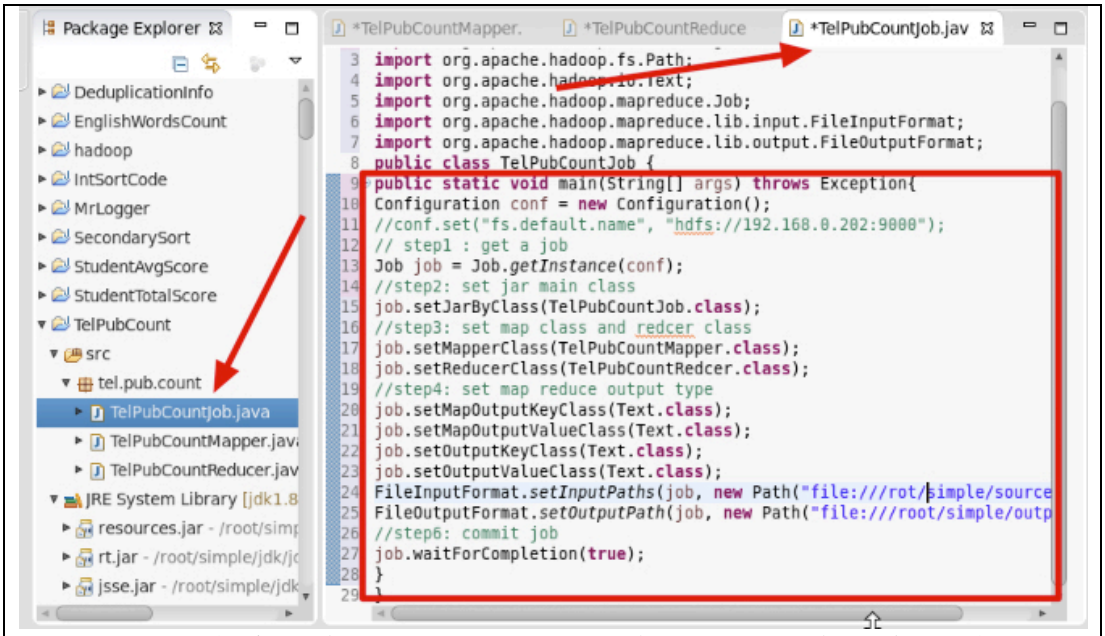
2.6 在项目 src 目录下指定的包名” tel.pub.count”下右键点击，新建一个类名为“TelPubCountRedcer “并继承 Reducer 类，然后添加该类中的代码内容如下所示。



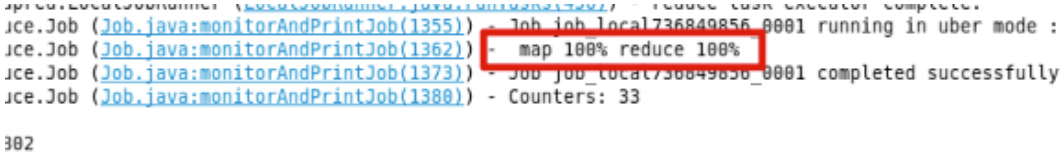
2.7 在项目 src 目录下指定的包名” tel.pub.count”下右键点击，新建一个测试主类名为” TelPubCountJob”并指定 main 主方法。



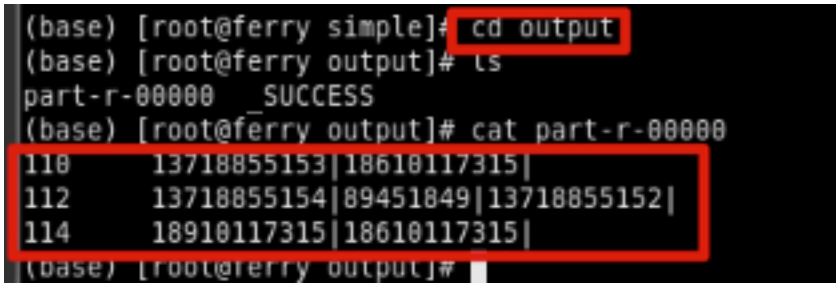
2.8 测试代码如下所示。



2.9 按照以上的步骤，把 mapper 和 reducer 阶段以及测试代码编写完毕之后，选中测试类” TelPubCountJob “，右键点击选择”Run as”--->”Java Application”，查看控制台显示内容查看是否正确执行。如图 9 所示



2.10 程序执行完毕之后，可以到输出信息目录/simple/output 下，执行查看命令:cat part-r-00000 查看对数据处理后产生的结果。



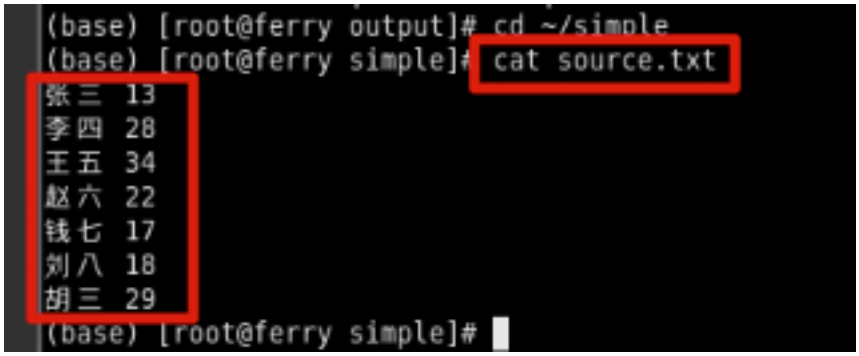
【案例 5.3】mapreduce 学生信息按年龄分区

一、项目准备阶段

需求：对所给的所给的学生信息，包括用户名和年龄，通过 map-reduce 模拟对海量学生信息进行处理，即实现把所有的学生信息按年龄界限为 18 进行分区。

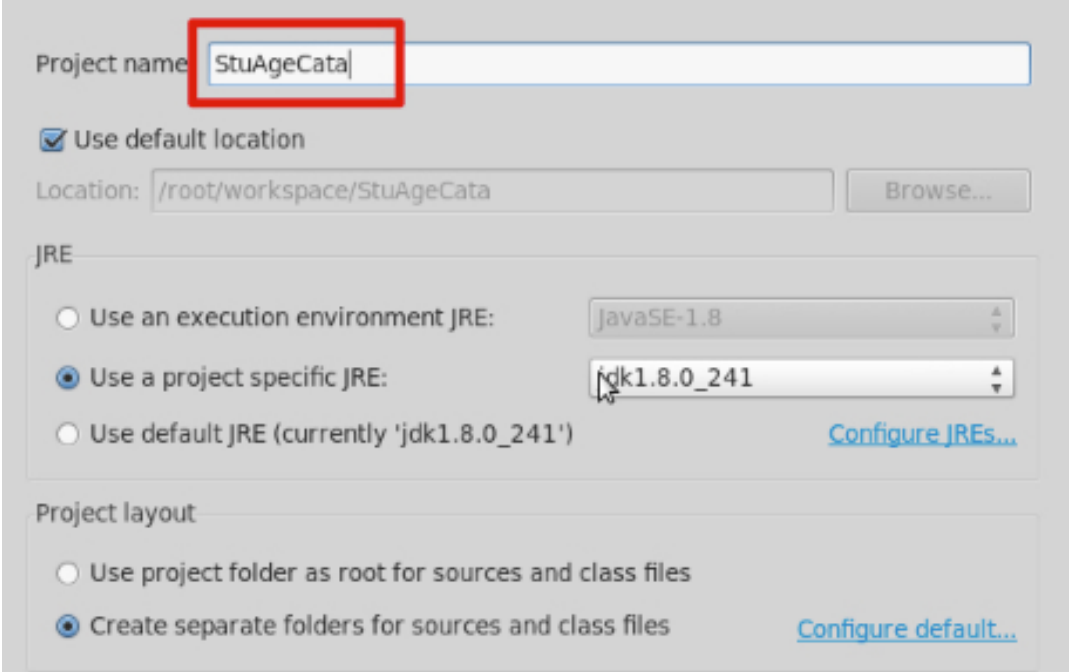
原始数据：
张三 13
李四 28
王五 34
赵六 22

钱七 17
刘八 18
胡三 29

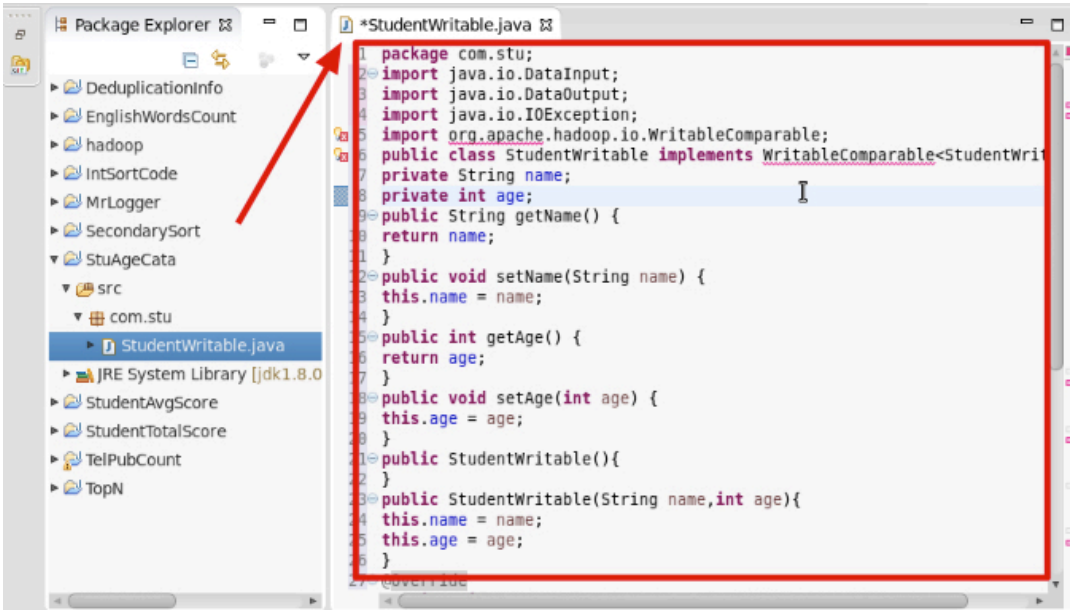


二、程序编写

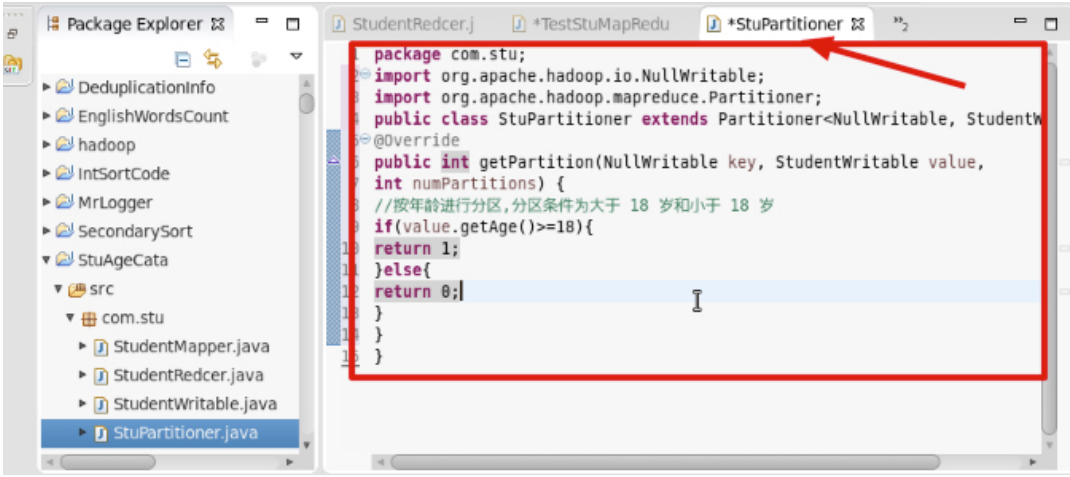
2.1 在 eclipse 中的项目列表中，右键点击，选择“new“—>”Java Project…”新建一个项目“StuAgeCata” 。



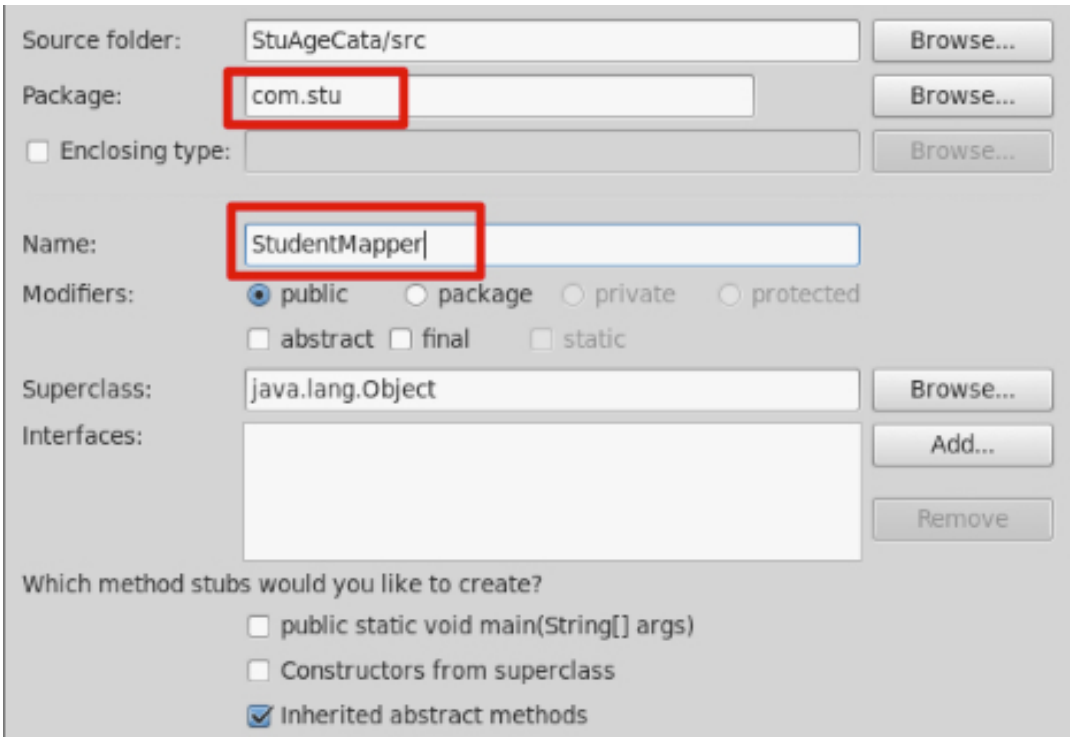
2.2 在项目 src 目录下，右键点击，选择“新建”创建一个类文件名称为“StudentWritable”并指定包名” com.stu”,该类是对给定数据的三列值的封装，并作为 mapper 的输出键值对象 。



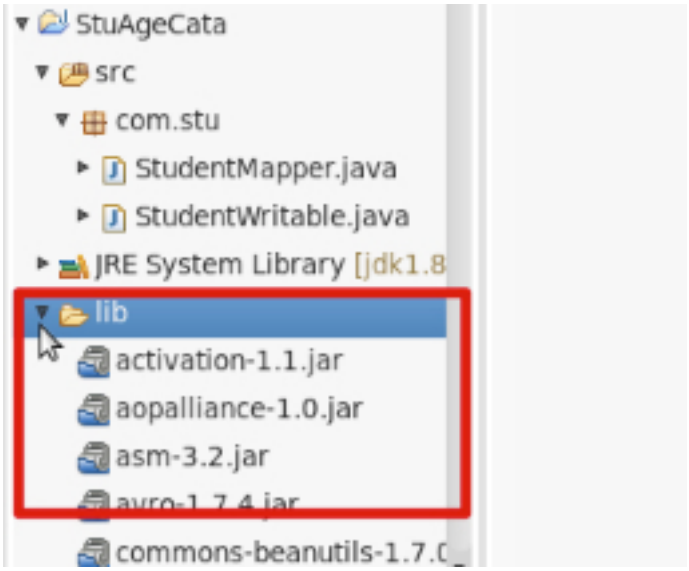
2.3 在项目 src 目录下，右键点击包名” com.stu”选择”New”—“Class”创建一个类文件名称为“StuPartitioner”，该类是对数据处理后的结果进行分区设置 。



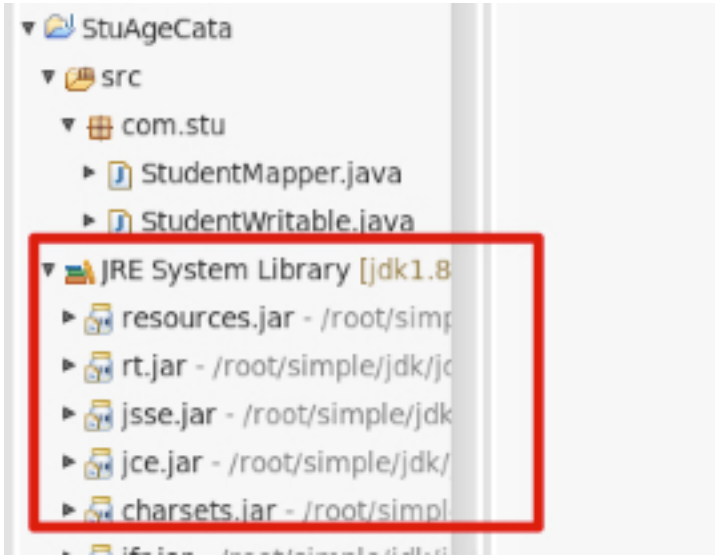
2.4 在项目 src 目录下，右键点击包名” com.stu”选择”New”—“Class”创建一个类文件名称为“StudentMapper”。



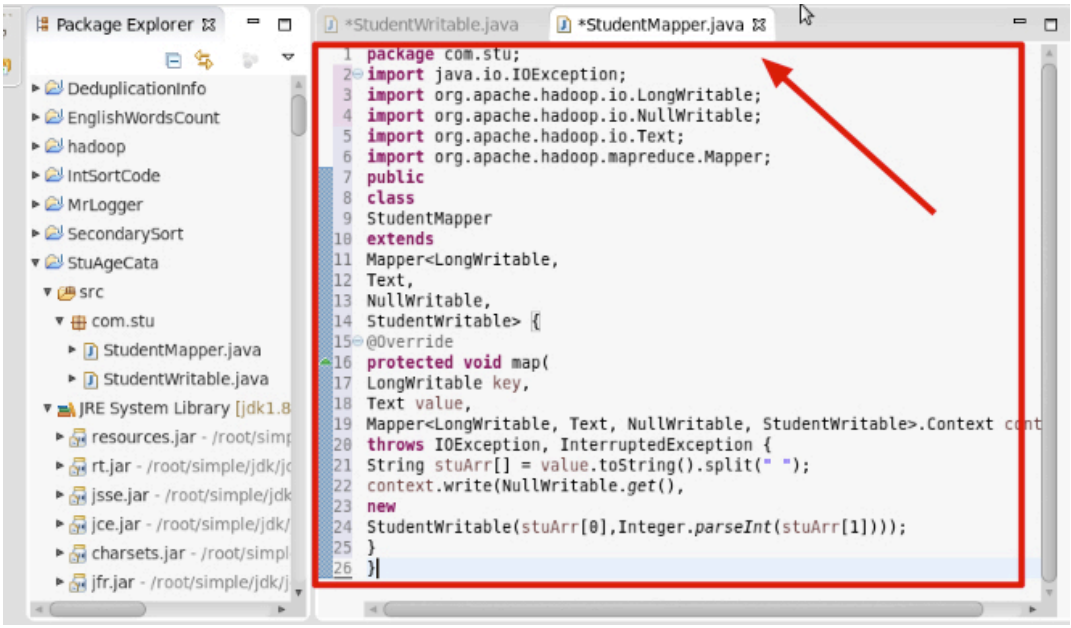
2.5 在编写“StudentMapper”类之前需要把 hadoop 相关的 jar 包导入，首先右击项目选择“New”—“Folder”创建一个 lib 文件夹并把指定位置中(桌面 lib 文件夹)的包放入该文件中。



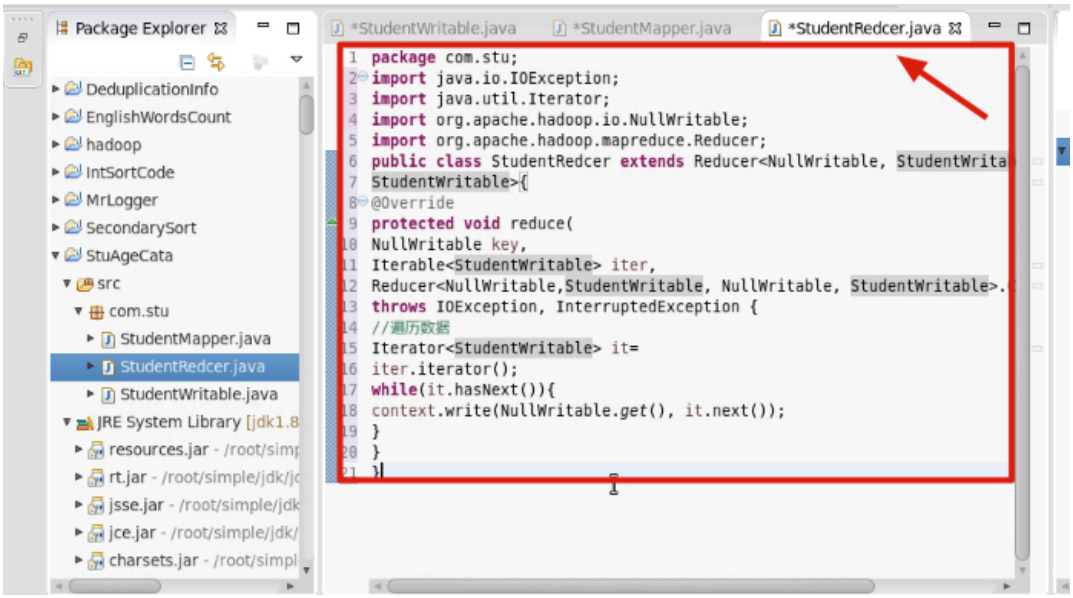
2.6 把 lib 下所有的 jar 包导入到环境变量，首先全选 lib 文件夹下的 jar 包文件，右键点击，选择“build path”-->“add to build path”,添加后，发现在项目下很多奶瓶图表的 jar 包。



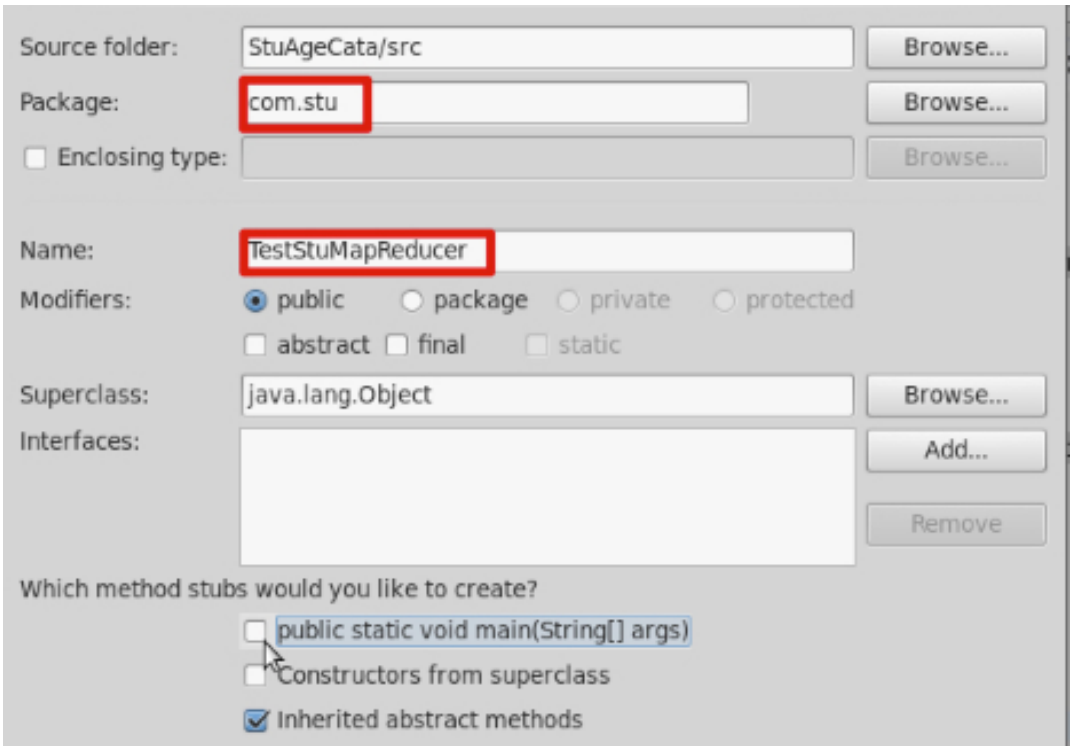
2.7 让类“StudentMapper”继承类 Mapper 同时指定需要的参数类型，根据业务逻辑修改 map 类的内容如下。



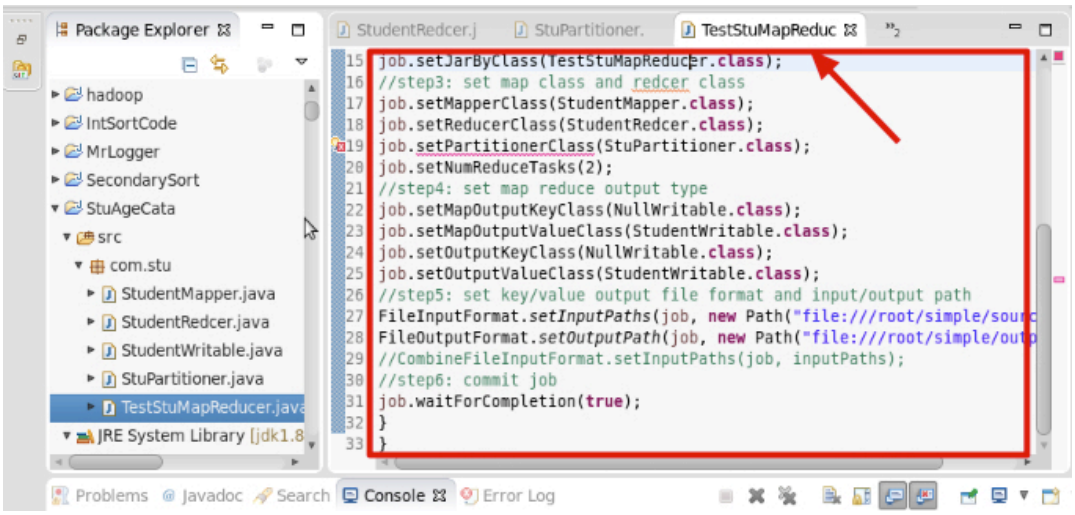
2.8 在项目 src 目录下指定的包名” com.stu”下右键点击，新建一个类名为“StudentRedcer“并继承 Reducer 类，然后添加该类中的代码内容如下所示。



2.9 在项目 src 目录下指定的包名” com.stu”下右键点击，新建一个测试主类名为” TestStuMapReducer”并指定 main 主方法。



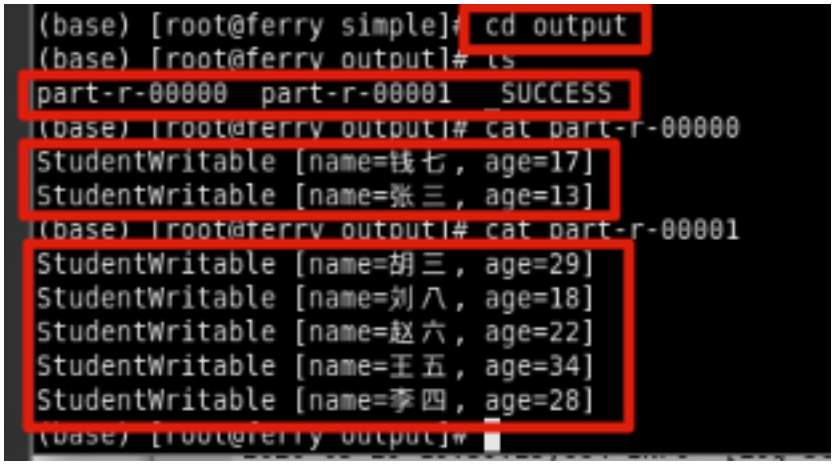
2.10 最后在项目 src 目录下指定的包名” com.stu”下右键点击，新建一个测试主类名为” TestStuMapReducer “添加测试代码如下所示。



2.11 按照以上的步骤，把 mapper 和 reducer 阶段以及测试代码编写完毕之后，选中测试类” TestFlow “，右键点击选择”Run as”--->”Java Application”,查看控制台显示内容查看是否正确执行。

```
.Job (Job.java:monitorAndPrintJob(1355)) - Job job_local1724031442_0001 running in u
.Job (Job.java:monitorAndPrintJob(1362)) - map 100% reduce 100%
.Job (Job.java:monitorAndPrintJob(1373)) - Job job_local1724031442_0001 completed su
.Job (Job.java:monitorAndPrintJob(1380)) - Counters: 33
```

2.12 程序执行完毕之后，可以到输出信息目录/simple/output 下，执行查看命令:cat part-r-00000，cat part-r-00001 查看对数据处理后产生的结果。



【案例 5.4】mapreduce 电话号码流量封装类作为键并排序

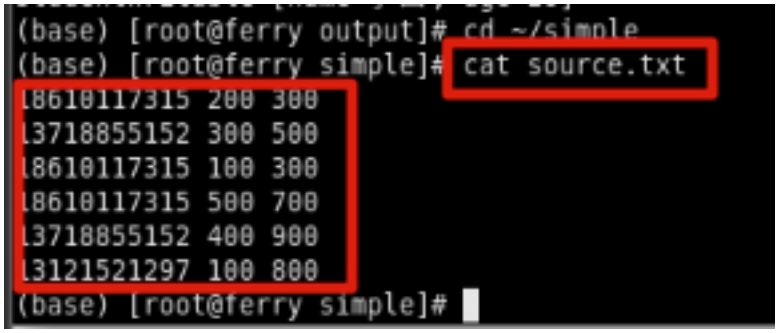
一、项目准备阶段

需求：对所给的所有电话号码产生的流量记录进行封装并把封装类作为 map 输出键进行指定电话号码前 3 为的大小进行排序升序后输出。

原始数据：

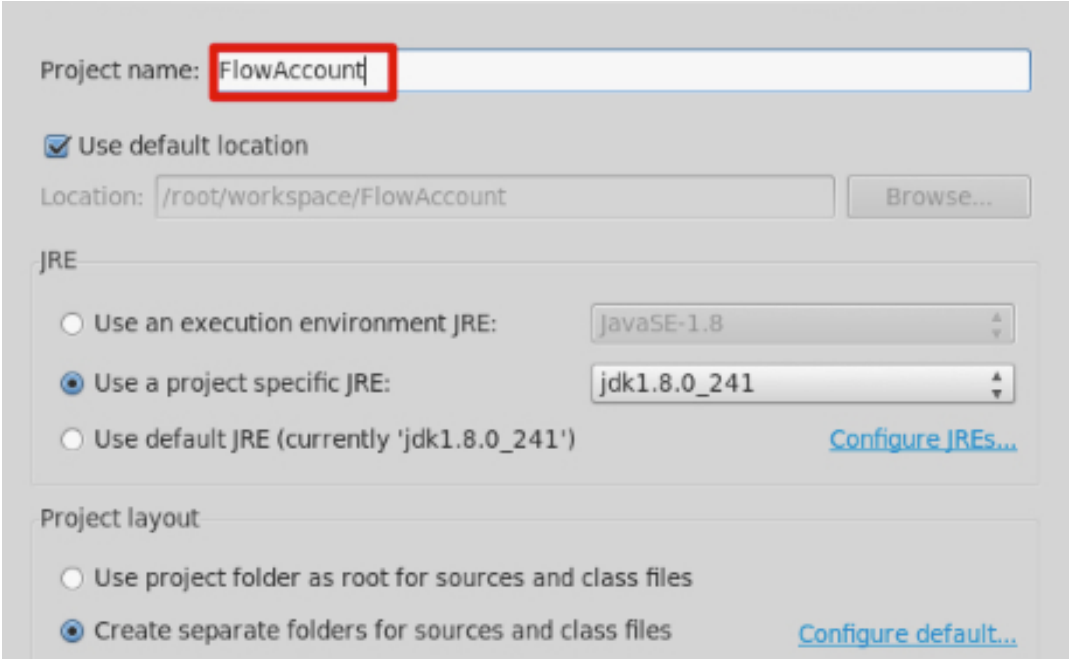
18610117315 200 300
13718855152 300 500
18610117315 100 300

18610117315 500 700
13718855152 400 900
13121521297 100 800

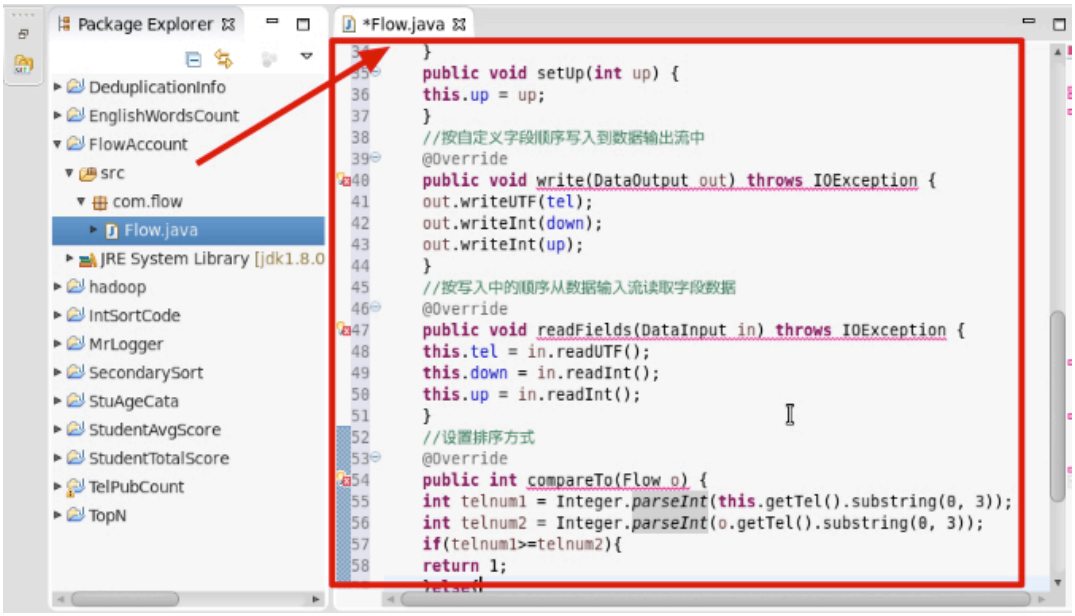


二、程序编写

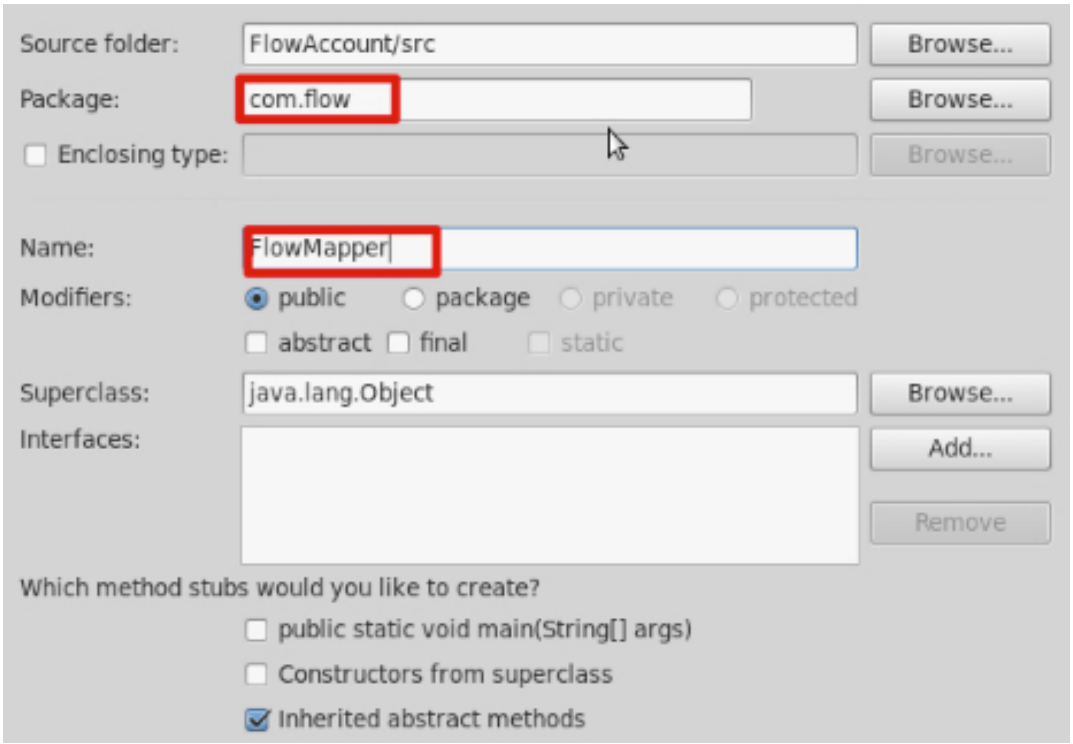
2.1 在 eclipse 中的项目列表中，右键点击，选择“new“—>”Java Project…”新建一个项目“FlowAccount” 。



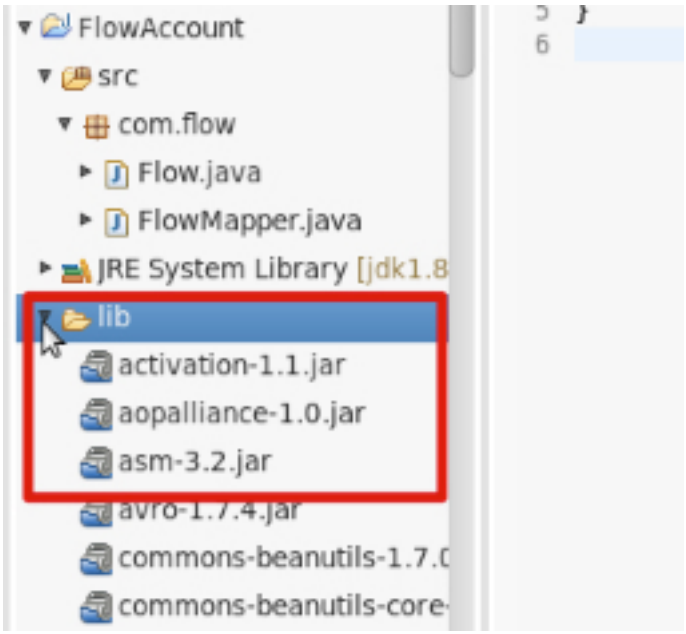
2.2 在项目 src 目录下，右键点击，选择”New”—“Class”创建一个类文件名称为“Flow”并指定包名” com.flow”,该类是对给定数据的三列值的封装，并作为 mapper 的输出键值对象 。



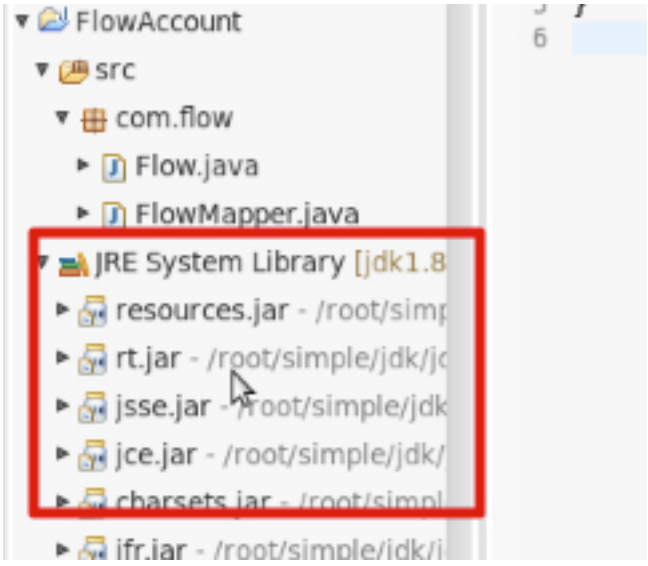
2.3 在项目 src 目录下，右键点击，选择”New”—“Class”创建一个类文件名称为“FlowMapper”并指定包名” com.flow” 。



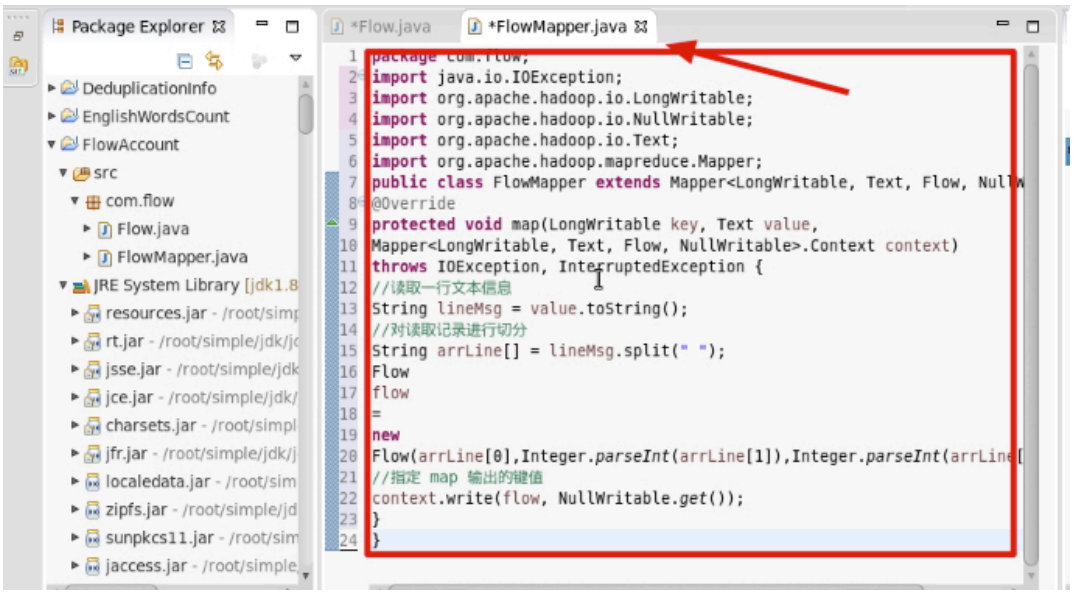
2.4 在编写“FlowMapper”类之前需要把 hadoop 相关的 jar 包导入，首先右击项目选择“New”—“Folder”创建一个 lib 文件夹并把指定位置中(桌面 lib 文件夹)的包放入该文件中。



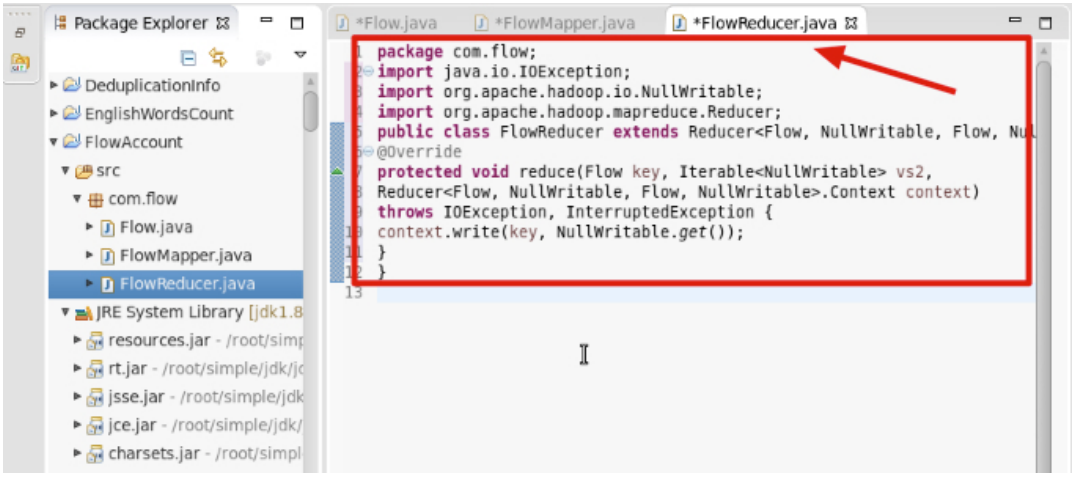
2.5 把 lib 下所有的 jar 包导入到环境变量，首先全选 lib 文件夹下的 jar 包文件，右键点击，选择“build path”-->“add to build path”,添加后，发现在项目下很多奶瓶图表的 jar 包。



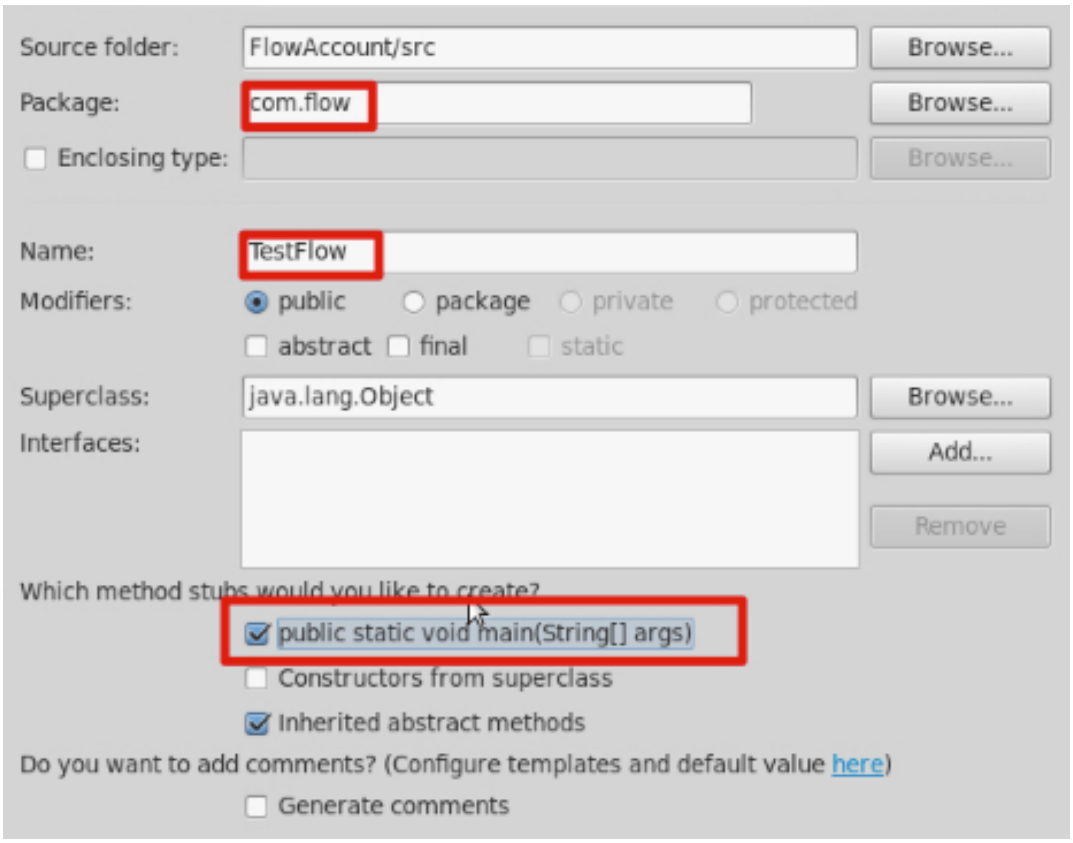
2.6 让类“FlowMapper”继承类 Mapper 同时指定需要的参数类型，根据业务逻辑修改 map 类的内容如下。



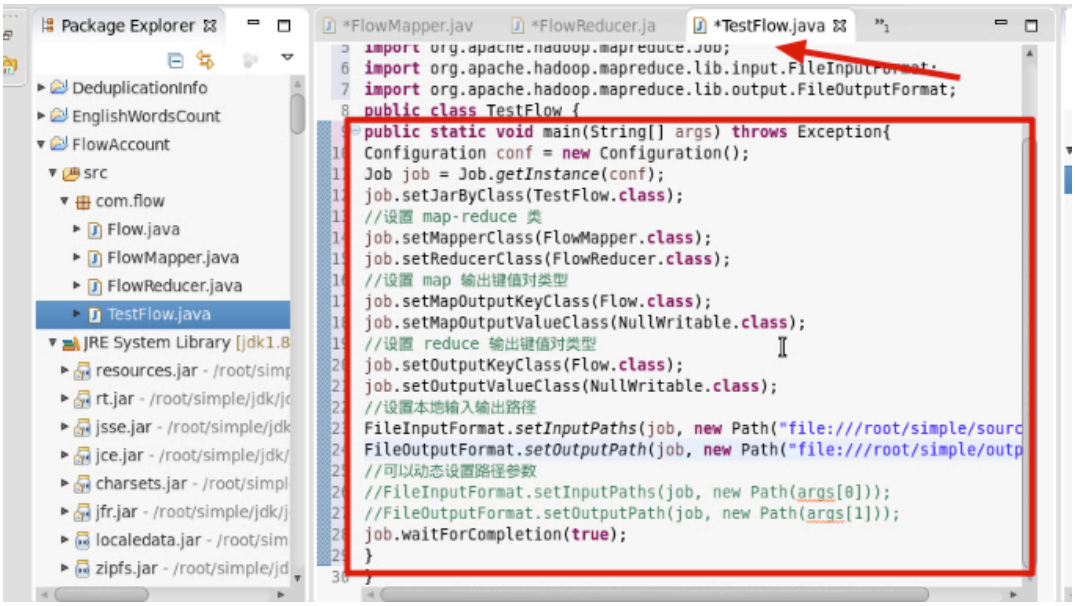
2.7 在项目 src 目录下指定的包名” com.flow”下右键点击，新建一个类名为“FlowReducer “并继承 Reducer 类，然后添加该类中的代码内容如下所示。



2.8 在项目 src 目录下指定的包名” com.flow”下右键点击，新建一个测试主类名为” TestFlow ”并指定 main 主方法。



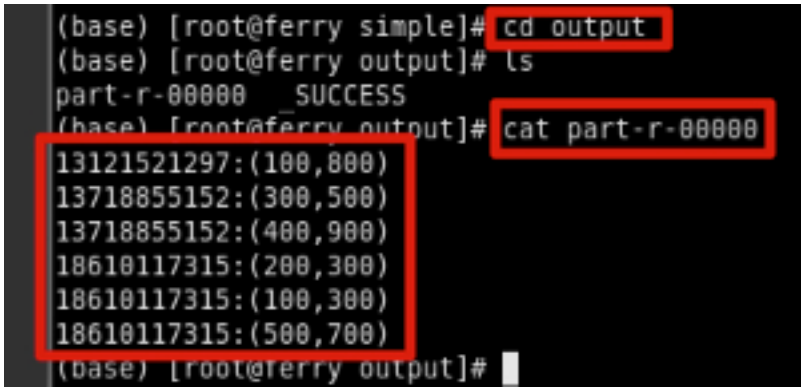
2.9 测试代码如下所示。



2.10 按照以上的步骤，把 mapper 和 reducer 阶段以及测试代码编写完毕之后，选中测试类” TestFlow “，右键点击选择”Run as”--->”Java Application”,查看控制台显示内容查看是否正确执行

```
uce.Job (Job.java:monitorAndPrintJob(1355)) - Job job_local285986574_0001 running in uber mode
uce.Job (Job.java:monitorAndPrintJob(1362)) - map 100% reduce 100%
uce.Job (Job.java:monitorAndPrintJob(1373)) - Job job_local285986574_0001 completed successful
uce.Job (Job.java:monitorAndPrintJob(1380)) - Counters: 33
```

2.11 程序执行完毕之后，可以到输出信息目录/simple/output 下，执行查看命令:cat part-r-00000，查看对数据处理后产生的结果。

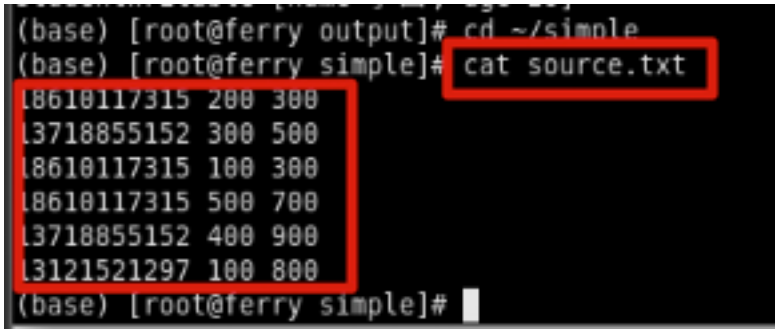


【案例 5.5】mapreduce 电话号码流量统计并分区

一、项目准备阶段

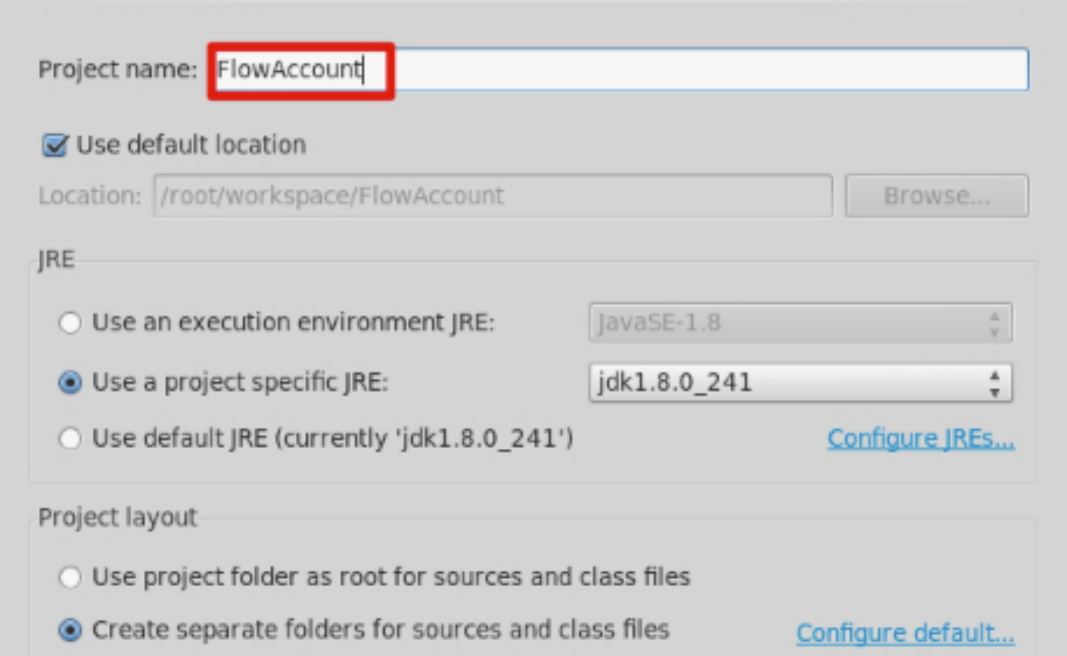
需求：对所给的所有电话号码产生的流量记录进行按电话号码进行汇总，求出所有相同电话号码产生的流量和。

原始数据：
18610117315 200 300
13718855152 300 500
18610117315 100 300
18610117315 500 700
13718855152 400 900
13121521297 100 800

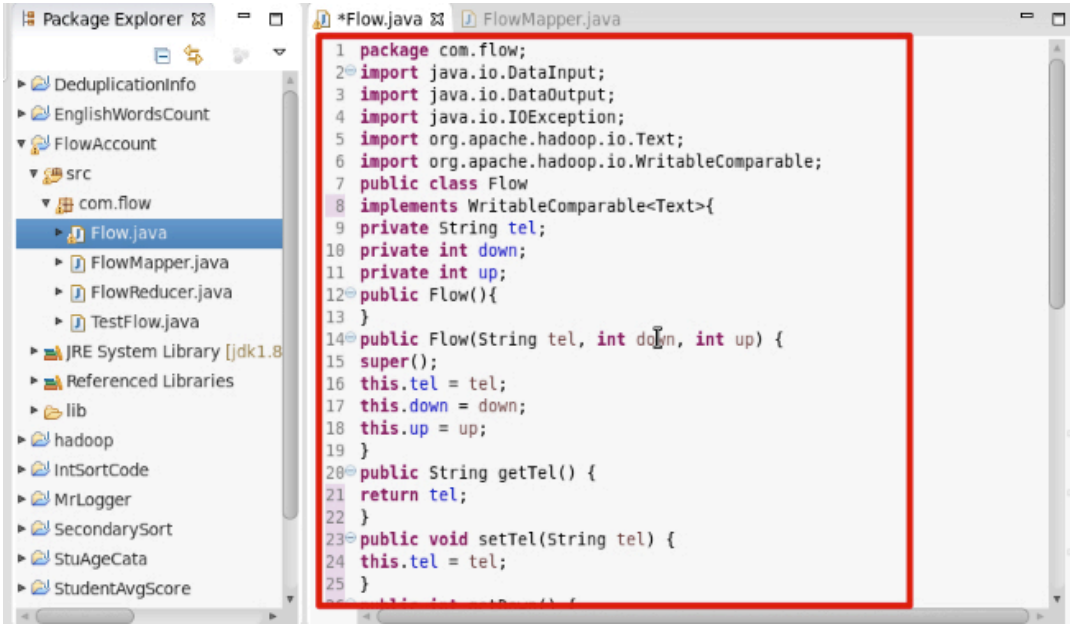


二、程序编写

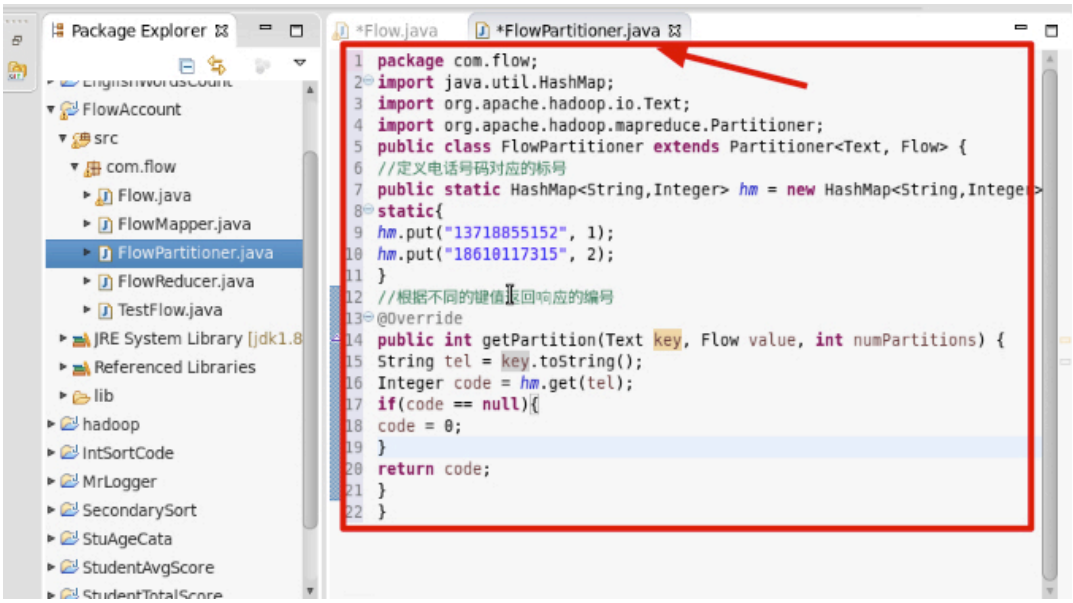
2.1 在 eclipse 中的项目列表中，右键点击，选择“new“—>”Java Project…”新建一个项目“FlowAccount” 。



2.2 在项目 src 目录下，右键点击，选择”New”—“Class”创建一个类文件名称为“Flow”并指定包名” com.flow”,该类是对给定数据的三列值的封装，并作为 mapper 的输出键值对象 。



2.3 在项目 src 目录下，右键点击，选择”New”—“Class”创建一个类文件名称为“FlowPartitioner”并指定包名”com.flow”,该类是对数据处理后的结果进行分区设置 。代码实现如下：

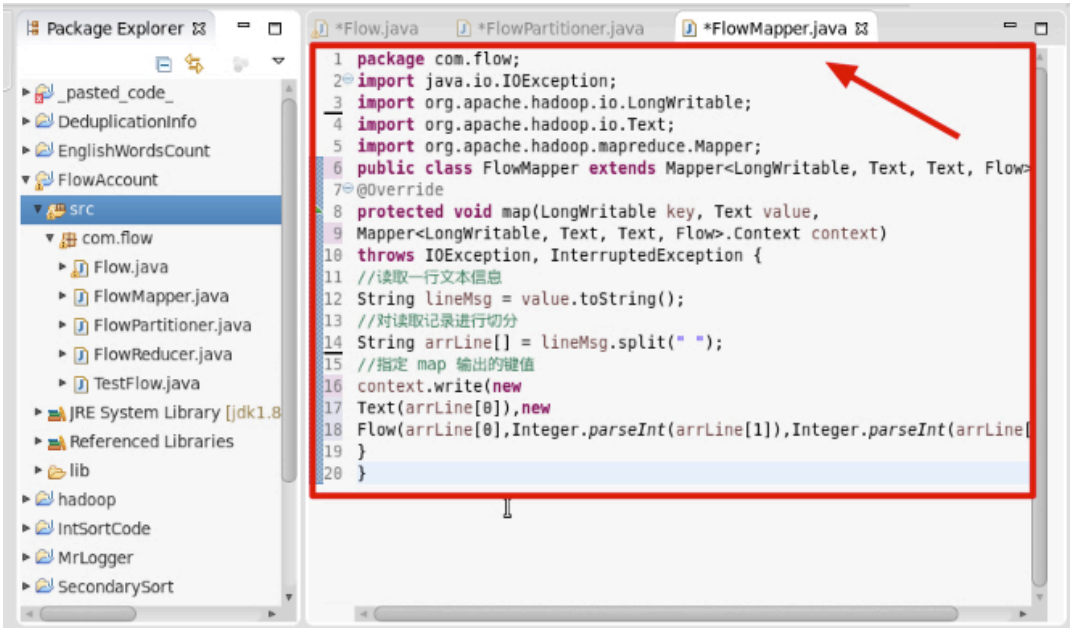


2.4 在项目 src 目录下，右键点击，选择”New”—“Class”创建一个类文件名称为“FlowMapper”并指定包名” com.flow” 。（同上案例）

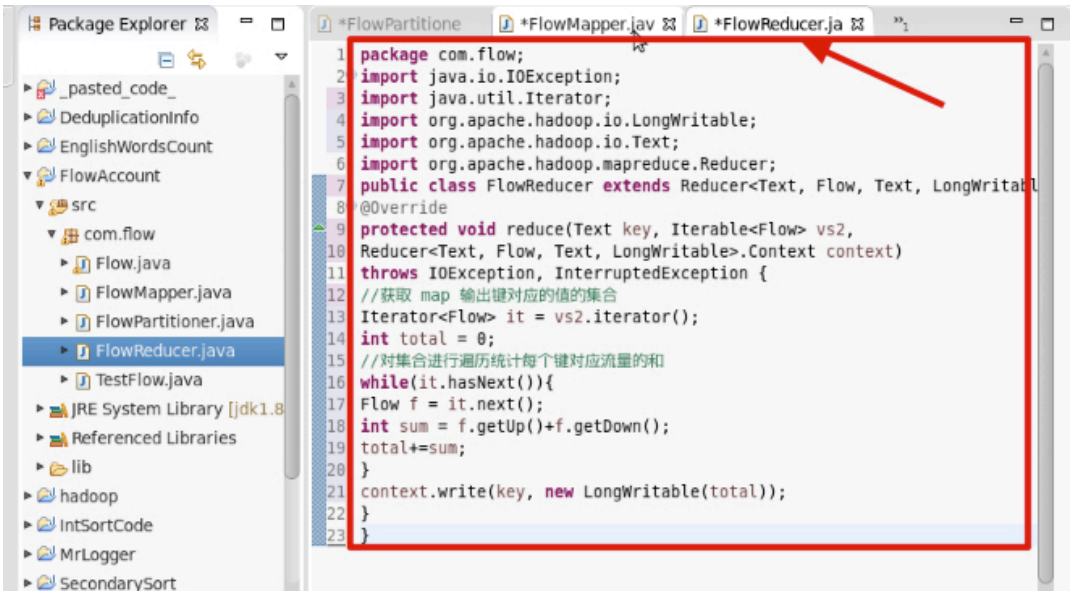
2.5 在编写“FlowMapper”类之前需要把 hadoop 相关的 jar 包导入，首先右击项目选择“New”—“Folder”创建一个 lib 文件夹并把指定位置中(桌面 lib 文件夹)的包放入该文件中。（同上案例）

2.6 把 lib 下所有的 jar 包导入到环境变量，首先全选 lib 文件夹下的 jar 包文件，右键点击，选择“build path”-->“add to build path”,添加后，发现在项目下很多奶瓶图表的 jar 包。（同上案例）

2.7 让类“FlowMapper”继承类 Mapper 同时指定需要的参数类型，根据业务逻辑修改 map 类的内容如下。

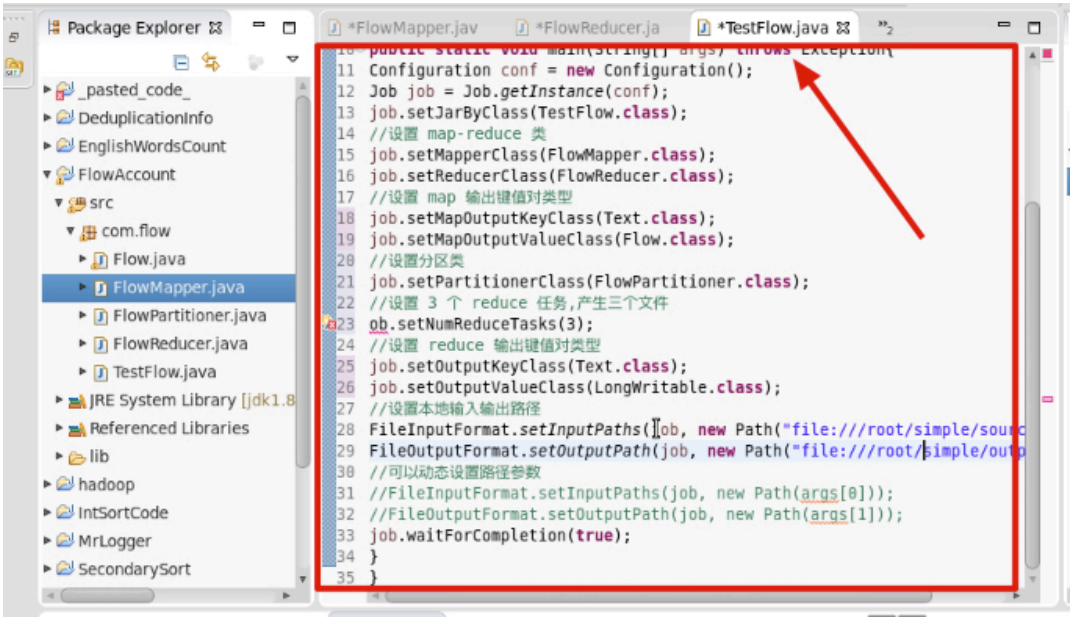


2.8 在项目 src 目录下指定的包名” com.flow”下右键点击，新建一个类名为”FlowReducer”并继承 Reducer 类,然后添加该类中的代码内容如下所示。



2.9 在项目 src 目录下指定的包名” com.flow”下右键点击，新建一个测试主类名为” TestFlow ”并指定 main 主方法。（同上案例）

2.10 测试代码如下所示。

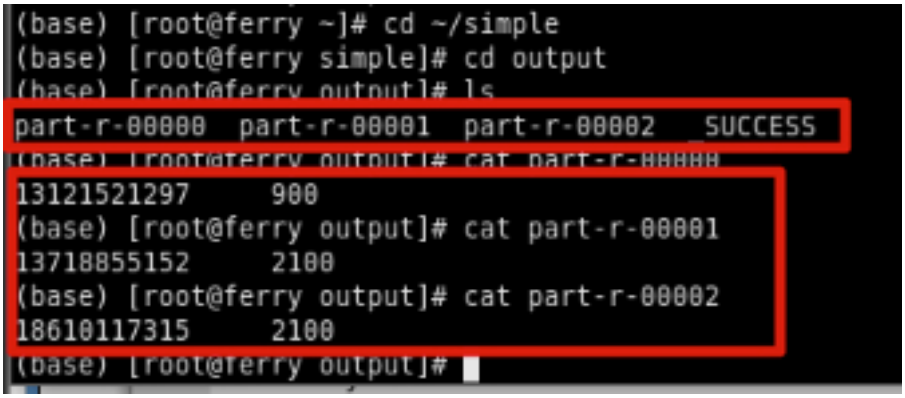


2.11 按照以上的步骤，把 mapper 和 reducer 阶段以及测试代码编写完毕之后，选中测试类” TestFlow “，右键点击选择”Run as”--->”Java Application”,查看控制台显示内容查看是否正确执行。

```
educe.Job (Job.java:monitorAndPrintJob(1355)) - Job job_local1018001584_0001 running in
educe.Job (Job.java:monitorAndPrintJob(1362)) -  map 100% reduce 100%
educe.Job (Job.java:monitorAndPrintJob(1373)) - Job job_local1018001584_0001 completed s
educe.Job (Job.java:monitorAndPrintJob(1380)) - Counters: 33

=3103
ten=877785
tions=0
 operations=0
```

2.12 程序执行完毕之后，可以到输出信息目录/simple/output 下，执行查看命令:cat part-r-00000，cat part-r-00001，cat part-r-00002 查看对数据处理后产生的结果。

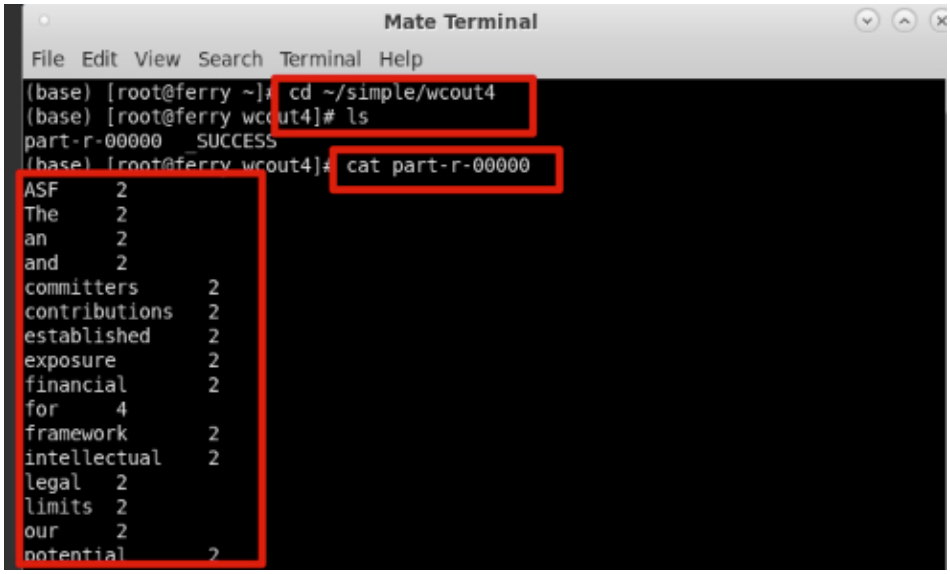


实验结论与分析：

1. 粘贴最终结果图，并对结果做分析说明,.,,.....

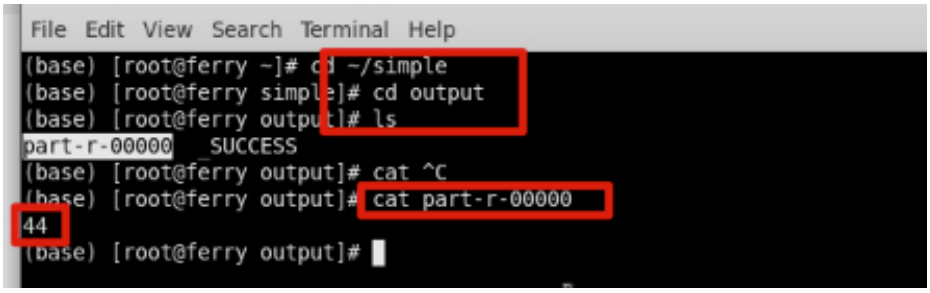
1.MR1-1 词频统计

首先对待处理的信息进行拆分，拆分之后在 map 阶段，把拆分的每个单词作为 map 方法的输出键，而 map 的方法输出的值设置为 1，最后在 reduce 阶段对每个键的值集合进行遍历并把遍历的值进行相加，输出结果即可。



2.MR1-2 单词个数

首先对待处理的信息进行拆分，拆分之后在 map 阶段，拆分后计算出单词个数并作为 map 方法的输出值，而 map 的方法输出键作为 NullWritable 即可，最后在 reduce 阶段对每个键的值集合进行遍历并把遍历的值进行相加，输出结果即可。

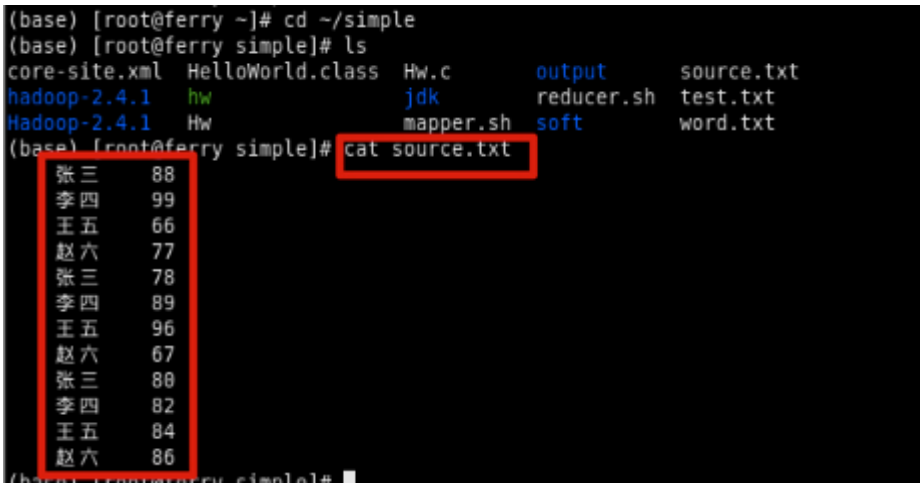


3.MR2-2 成绩统计

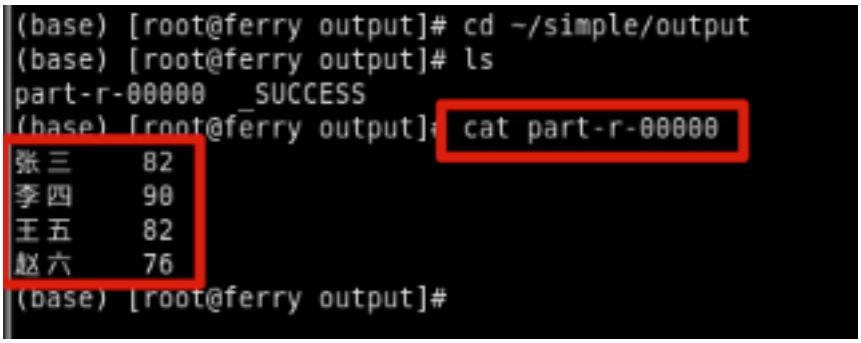
Map 处理的是一个纯文本文件，文件中存放的数据时每一行表示一个学生的姓名和他相应一科成绩。Mapper 处理的数据是由 InputFormat 分解过的数据集，其中 InputFormat 的作用是将数据集切割成小数据集 InputSplit，每一个 InputSplit 将由一个 Mapper 负责处理。此外，InputFormat 中还提供了一个 RecordReader 的实现，并将一个 InputSplit 解析成<key,value>对提供给了 map 函数。InputFormat 的默认值是 TextInputFormat，它针对文本文件，按行将文本切割成 InputSlit，并用 LineRecordReader 将 InputSplit 解析成<key,value>对，key 是行在文本中的位置，value 是文件中的一行。Map 的结果会通过 partion 分发到 Reducer，Reducer 做完 Reduce 操作后，将通

过以格式 OutputFormat 输出。Mapper 最终处理的结果对<key,value>，会送到 Reducer 中进行合并，合并的时候，有相同 key 的键/值对则送到同一个 Reducer 上。Reducer 是所有用户定制 Reducer 类地基础，它的输入是 key 和这个 key 对应的所有 value 的一个迭代器，同时还有 Reducer 的上下文。Reduce 的结果由 Reducer.Context 的 write 方法输出到文件中。

【案例 3.1】 mapreduce 学生总成绩报表



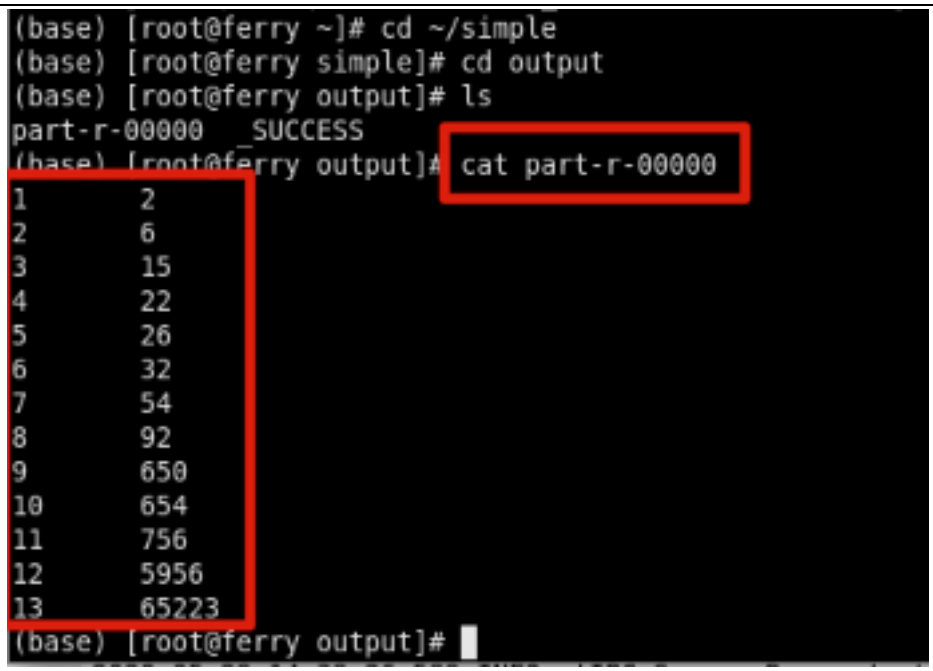
【案例 3.2】mapreduce 学生各科平均成绩报表



4.MR3 MRShuffle

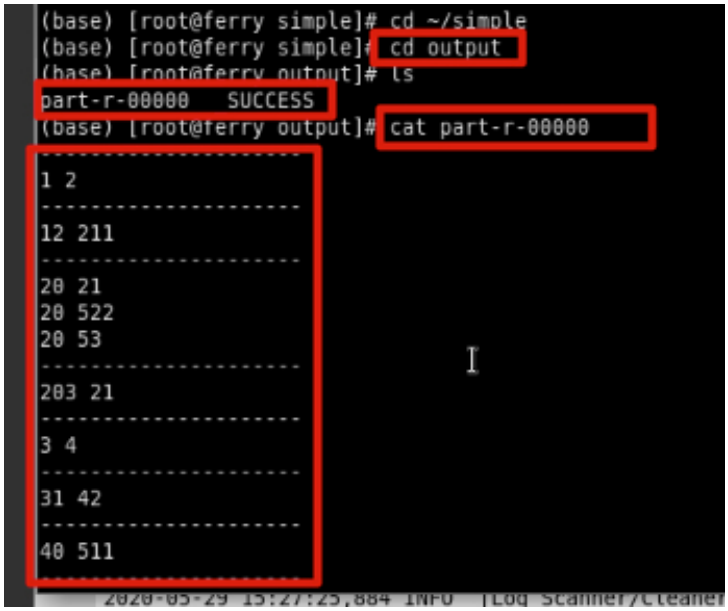
【案例 4.1】mapreduce 整数排序并指定顺序编号

本试验要求对一系列数据进行排序你，自然就会想到只要把这些数据作为 map 阶段的输出键即可，排序后的数据在输出时，在数值前面添加一个数字的自然顺序编号，此时只要考虑在 reduce 阶段定义一个全局变量，每次执行 reduce 方法时，就是对 map 阶段的 key 进行处理，此时全局变量就可以作为编号，每执行一次 reduce，全局变量就会增加 1。



【案例 4.2】mapreduce 两列数据的二次排序

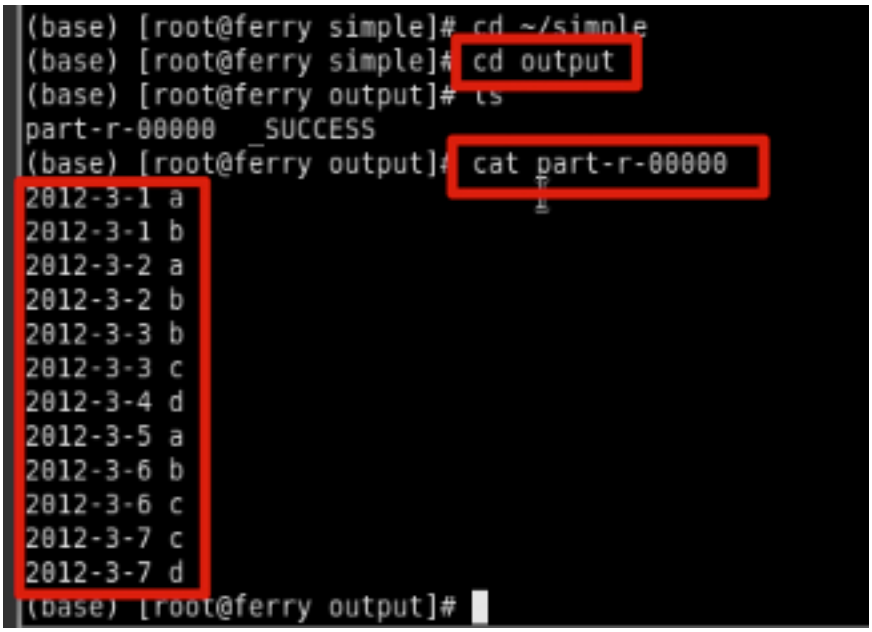
在 MapReduce 操作时，我们知道传递的<key,value>会按照 key 的大小进行排序，最后输出的结果是按照 key 排过序的。有的时候我们在 key 排序的基础上，对 value 也进行排序。这种需求就是二次排序。二次排序是在框架在对 key2 排序后再对 reduce 输出结果的结果 value3 进行二次排序的需求。在 map 阶段，使用 job.setInputFormatClass 定义的 InputFormat 将输入的数据集分割成小数据块 splites，同时 InputFormat 提供一个 RecordReder 的实现。本例子中使用的是 TextInputFormat，他提供的 RecordReader 会将文本的字节偏移量作为 key，这一行的文本作为 value。



【案例 4.3】mapreduce 多条数据去重处理

数据去重的最终目标是让原始数据中出现次数超过一次的数据在输出文件中

只出现一次。具体就是 reduce 的输入应该以数据作为 key，而对 value-list 则没有要求。当 reduce 接收到一个<key，value-list>时就直接将 key 复制到输出的 key 中，并将 value 设置成空值。

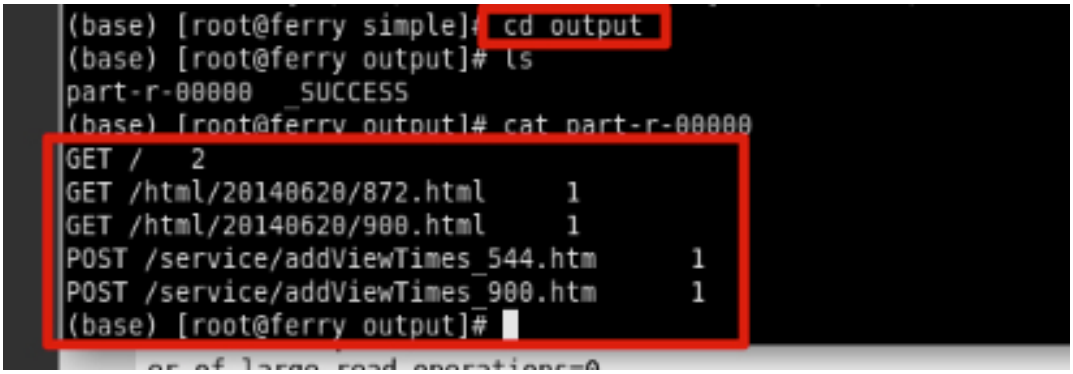


【案例 4.4】mapreduce 非结构化日志文件处理

需求：根据 tomcat 日志计算 url 访问了情况，具体的 url 如下，

要求：区别统计 GET 和 POST URL 访问量

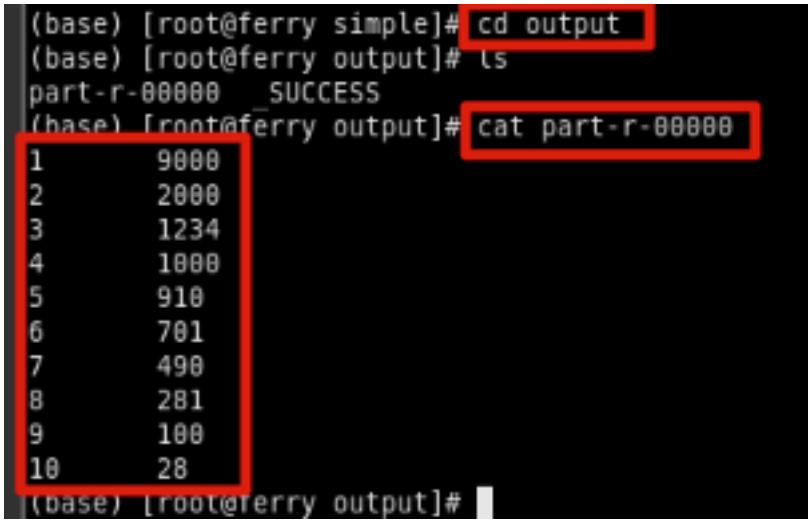
结果为：访问方式、URL、访问量



5.MR4 MR 调优

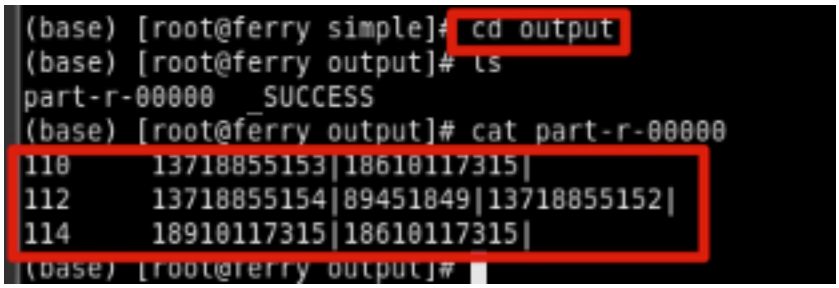
【案例 5.1】mapreducetopN 排名

在每一个 map 端，就求出开始 TopN，这样可以减少在 reduce 端的压力，即在每一个 map 中先求出 Top10，然后将这 top10,传给 reduce，让后 reduce 把所有全部的传来的数据中找出 top10



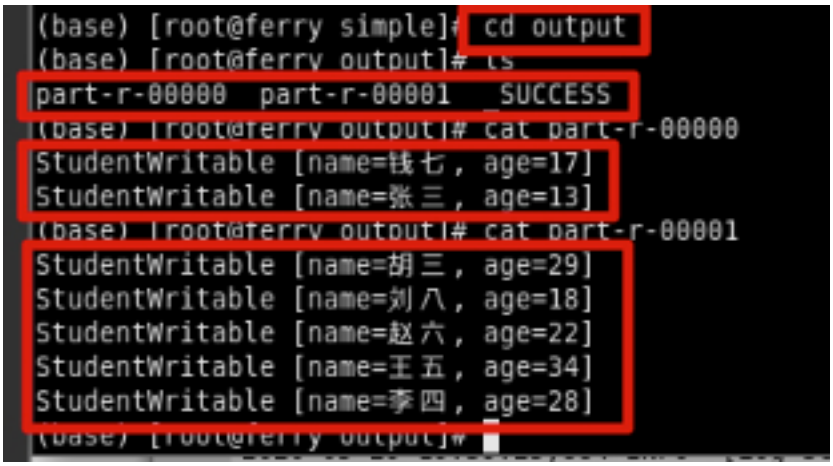
【案例 5.2】mapreduce 统计拨打公共服务号码的电话信息

原理：首先对待处理的信息进行拆分，拆分之后在 map 阶段，把公共服务电话作为 map 方法的输出键，用户电话作为 map 方法的输出值，最后在 reduce 阶段对每个键的值集合进行遍历并按指定规则拼接起来，输出结果即可。



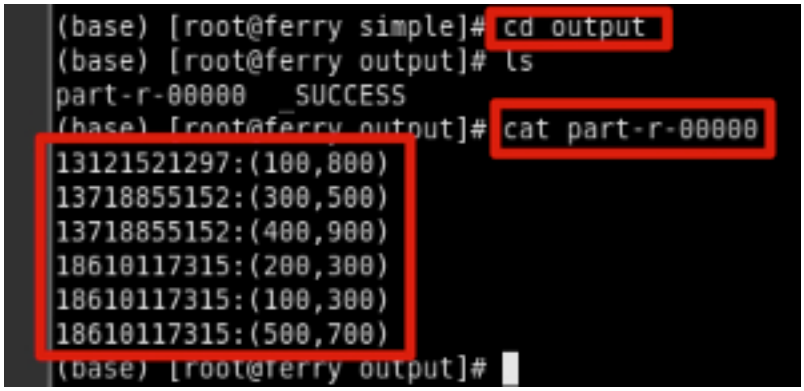
【案例 5.3】mapreduce 学生信息按年龄分区

原理：首先把学生的信息进行封装成新的类型，作为 map’阶段的输出值，因为不需要按键排序，所以键只需要设置为 NullWritable 类型即可。在 reduce 阶段对所有的值进行遍历之后直接输出即可



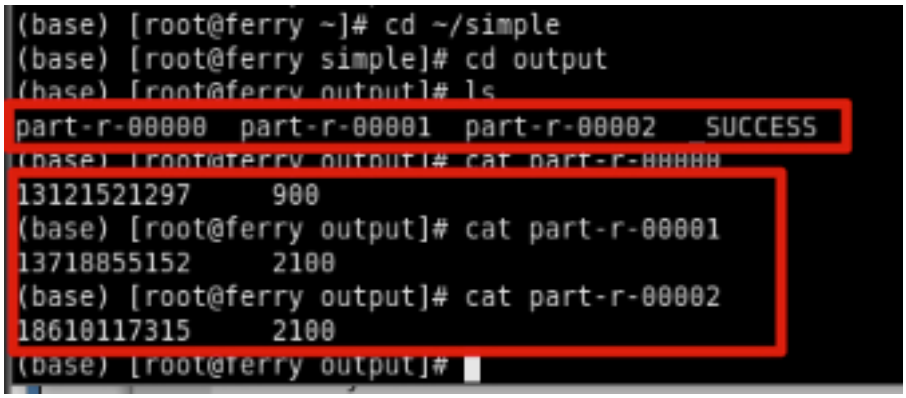
【案例 5.4】mapreduce 电话号码流量封装类作为键并排序

原理：首先按电话号码记录信息的封装类作为键进行排序，同时为电话记录的封装类型指定所属的数据类型（Writable），重写排序方法和规则。



【案例 5.5】mapreduce 电话号码流量统计并分区

原理：首先按电话号码作为键进行排序，相同键的内容形成一个集合，然后把相同键的所有内容值进行流量相加，最后按照指定分区条件进行分区输出。



2. 谈谈自己对这个实验的体会感受 xxxxx

通过本次实验，我掌握了基本的 MapReduce 编程方法；掌握用 MapReduce 解决一些常见的数据处理问题，包括数据去重、数据排序和数据挖掘等。本次实验到此结束，但我知道对 MapReduce 的学习是没有尽头的。学习就是一个不断积累的过程，学到东西，或多或少，以另一种方式将它记录下来。

总结了一下排除上述问题的一个自己的心得：

感觉刚开始 MapReduce 任务的坑比较多，很难定位到问题的所在，出现上述类问题主要有以下几点：1、集群环境处于不稳定状态；2、代码问题，可能是很小的一个细节；3、mapreduce 任务配置参数的问题。

总之，遇到问题，不要怕麻烦，从源头开始一步步排查，在排查的过程中不断总结，能力也就一步步在此过程中会慢慢提升。其实大多数人遇到问题最怕没有思路，不知道从哪里下手，这样会耗费大量时间。遇到问题能有个自己的思路是关键，我们首先得明白有哪些可能会导致出现这些问题，然后开始排查，这样处理问题才能有条不紊的进行下去，也是一种解决问题的比较好的方式。