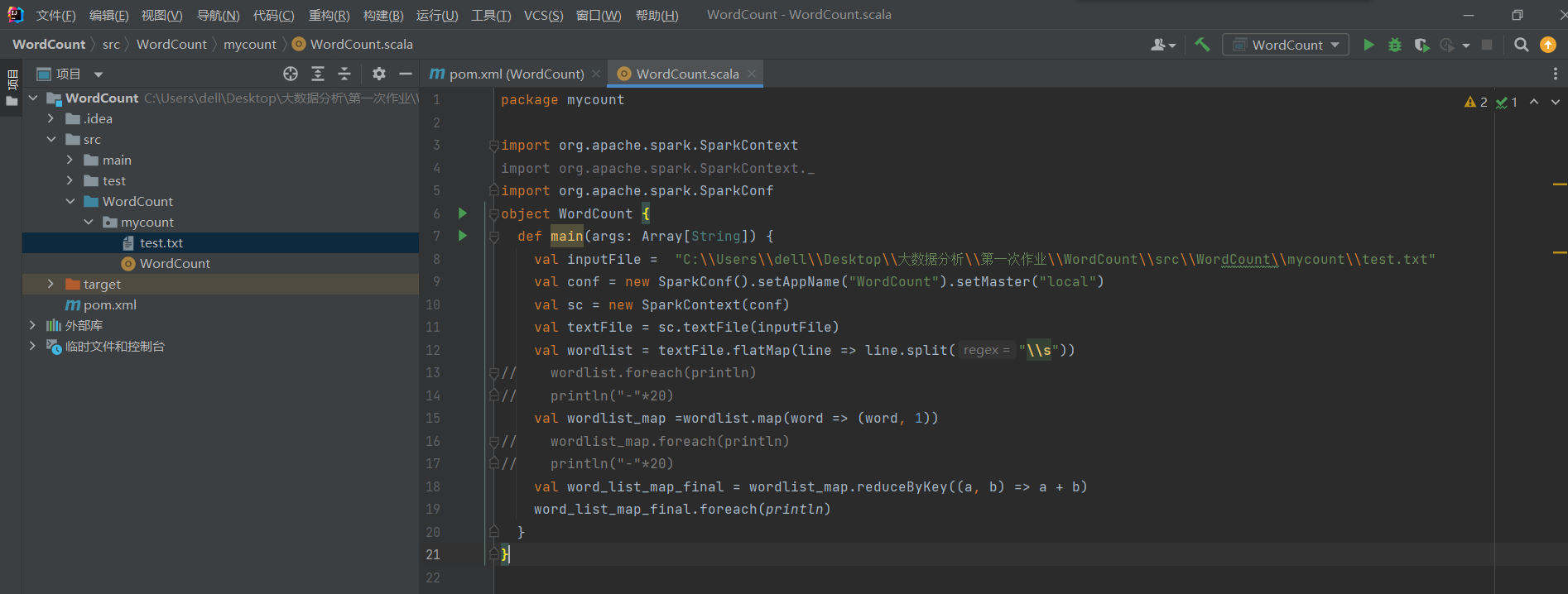
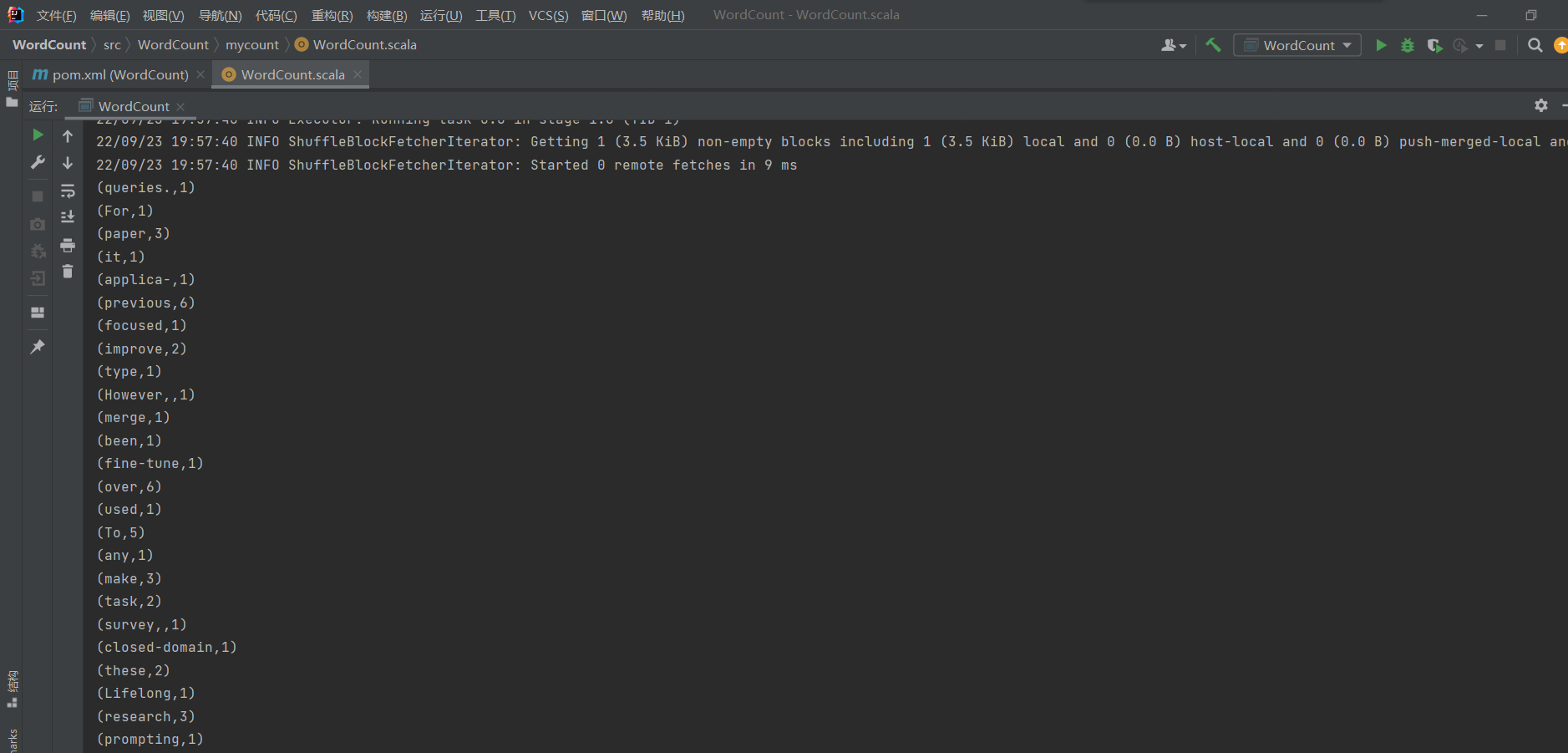
**《大数据分析》第一次作业实验报告**

201250070 郁博文

1. 源代码&运行结果





1. 实现思路

为了使用Spark的RDD（弹性分布式数据集），需要搭建Spark运行环境。值得指出的是，Spark的版本需要与Hadoop对应。

接下来就是使用Spark解决单词计数问题。

首先初始化Spark集群，由于需要读取的文件位于本地文件系统，所以直接使用local集群。

之后，由于Spark本身已经封装了textFile方法用于从分布式文件系统中读取文件，所以我们直接使用本地路径作为url读取文件。需要指出的是，test.txt文件由于编码问题，无法在idea中使用utf-8编码正常打开，所以我先在记事本中打开，然后复制内容，在idea中重新创建，这样才能正常读取。

# 在完成读取之后，由于Spark有flatmap方法用以扁平化数据集，我们通过匿名函数的方法，把读取的文件按照空格进行切割（这里是java的方法），形成一个扁平化的map结构。这样这个结构里存放的就都是单个单词。之后，使用map方法，为每个单词赋予初始出现次数1.使用reduceByKey方法对重复的数据集元素进行合并，显然，这里传入的匿名函数就是将原map结构中的元素值相加。最后把这个合并结果打印，这样就完成了单词计数。

1. 实验心得
2. 关于项目创建

刚接触Scala时，我使用的是idea中的Scala插件来创建scala项目。现在为了使用Spark的功能，我使用了maven的方法来创建项目。使用maven时需要写pom.xml。值得注意的是，配置文件中需要仔细检查spark和hadoop版本，否则会报错[java.lang.NoSuchMethodError: scala.Predef$.refArrayOps([Ljava/lang/Object;)Lscala/collection/mutable/ArrayOps;](https://www.cnblogs.com/itboys/p/6675075.html)

1. 关于单词分割

在这次作业中，约定单词按照空格隔开。部分分割由于本身带有标点，关于其是否为单词的判断具有二义性，例如“（i）”这个分割结果不是单词，而“e.g.”这个分割结果可以当成单词，因此这次作业不考虑标点对结果的影响。