重庆交通大学信息科学与工程学院

实验报告

班级：曙光 2101 班

姓名：李 幸 洋

学号：632107060506

实验项目名称：实验五　Spark 编程

实验项目性质：设计性

实验所属课程：大数据平台架构

实验室(中心)：逸夫楼 407

指导教师：何 伟

实验完成时间：2023 年 6 月 1 日

一、实验概述：

【实验目的】

1. 掌握 Scala 编程；
2. 掌握 Spark RDD 编程思想和方法；
3. 自学 Spark Streaming，Spark MIL 的开发。

【实验要求】

1. 保存程序，并自行存档；
2. 最终的程序都必须经过测试，验证是正确的；
3. 认真记录实验过程及结果，回答实验报告中的问题。

【实施环境】（使用的材料、设备、软件）

Linux 操作系统环境，VirtualBox 虚拟机，Java 开发环境，Hadoop。

二、实验内容

第 1 题 Scala 基础编程

【实验内容】

(1) 编写一个函数，从终端输入一个整数（1-9），输出相应的乘法表。

(2) *给你一个集合 List=（1,2,3,4，”abc”）,完成如下功能：

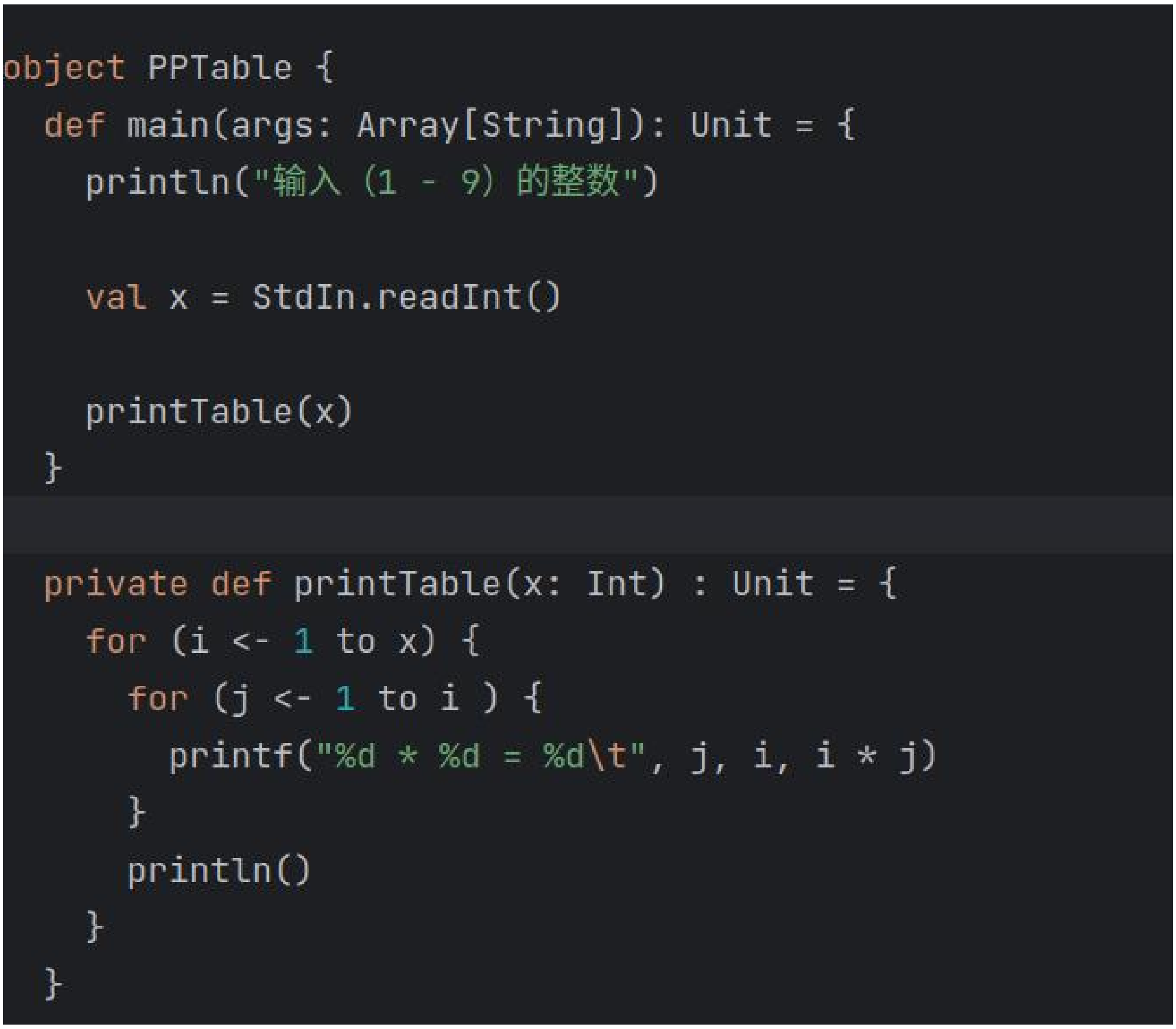
  a. 将集合中所有数字+1；

  b. 忽略掉非数字元素，返回一个新的集合（2,3,4,5）

【实验过程】（步骤、记录、数据、程序等）

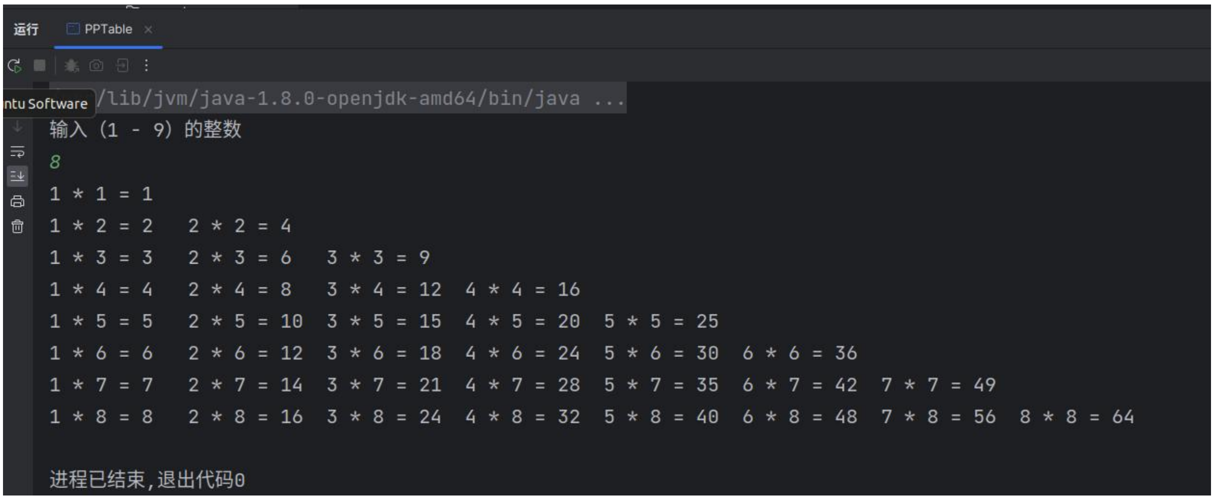
请提供相应代码及程序运行界面截图证明。

（1）代码：



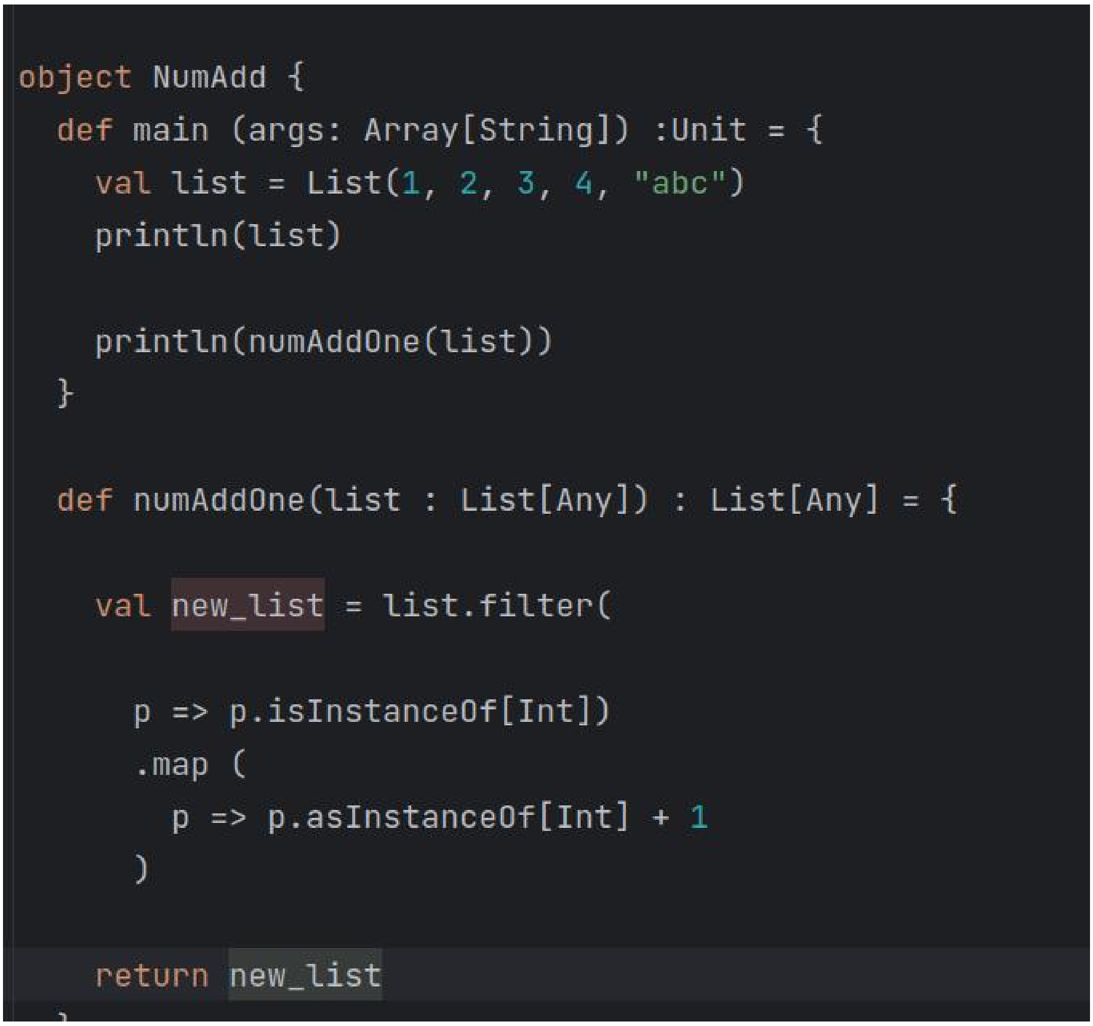
```
object PPTable {
  def main(args: Array[String]): Unit = {
    println("输入（1 - 9）的整数")

    val x = StdIn.readInt()

    printTable(x)
  }

  private def printTable(x: Int) : Unit = {
    for (i <- 1 to x) {
      for (j <- 1 to i ) {
        printf("%d * %d = %d\t", j, i, i * j)
      }
      println()
    }
  }
```

运行结果：



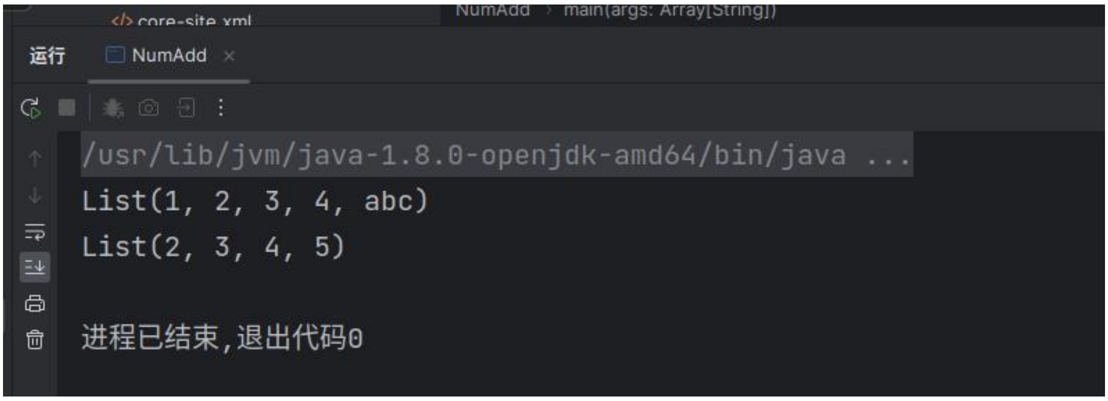
```
/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...
输入（1 - 9）的整数
8
1 * 1 = 1
1 * 2 = 2   2 * 2 = 4
1 * 3 = 3   2 * 3 = 6   3 * 3 = 9
1 * 4 = 4   2 * 4 = 8   3 * 4 = 12  4 * 4 = 16
1 * 5 = 5   2 * 5 = 10  3 * 5 = 15  4 * 5 = 20  5 * 5 = 25
1 * 6 = 6   2 * 6 = 12  3 * 6 = 18  4 * 6 = 24  5 * 6 = 30  6 * 6 = 36
1 * 7 = 7   2 * 7 = 14  3 * 7 = 21  4 * 7 = 28  5 * 7 = 35  6 * 7 = 42  7 * 7 = 49
1 * 8 = 8   2 * 8 = 16  3 * 8 = 24  4 * 8 = 32  5 * 8 = 40  6 * 8 = 48  7 * 8 = 56  8 * 8 = 64

进程已结束,退出代码0
```

(2) 代码：



```scala
object NumAdd {
  def main (args: Array[String]) :Unit = {
    val list = List(1, 2, 3, 4, "abc")
    println(list)

    println(numAddOne(list))
  }

  def numAddOne(list : List[Any]) : List[Any] = {

    val new_list = list.filter(

      p => p.isInstanceOf[Int])
      .map (
        p => p.asInstanceOf[Int] + 1
      )

    return new_list
```

运行结果：



```
/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...
List(1, 2, 3, 4, abc)
List(2, 3, 4, 5)

进程已结束,退出代码0
```

第 2 题. Scala 综合编程

【实验内容】

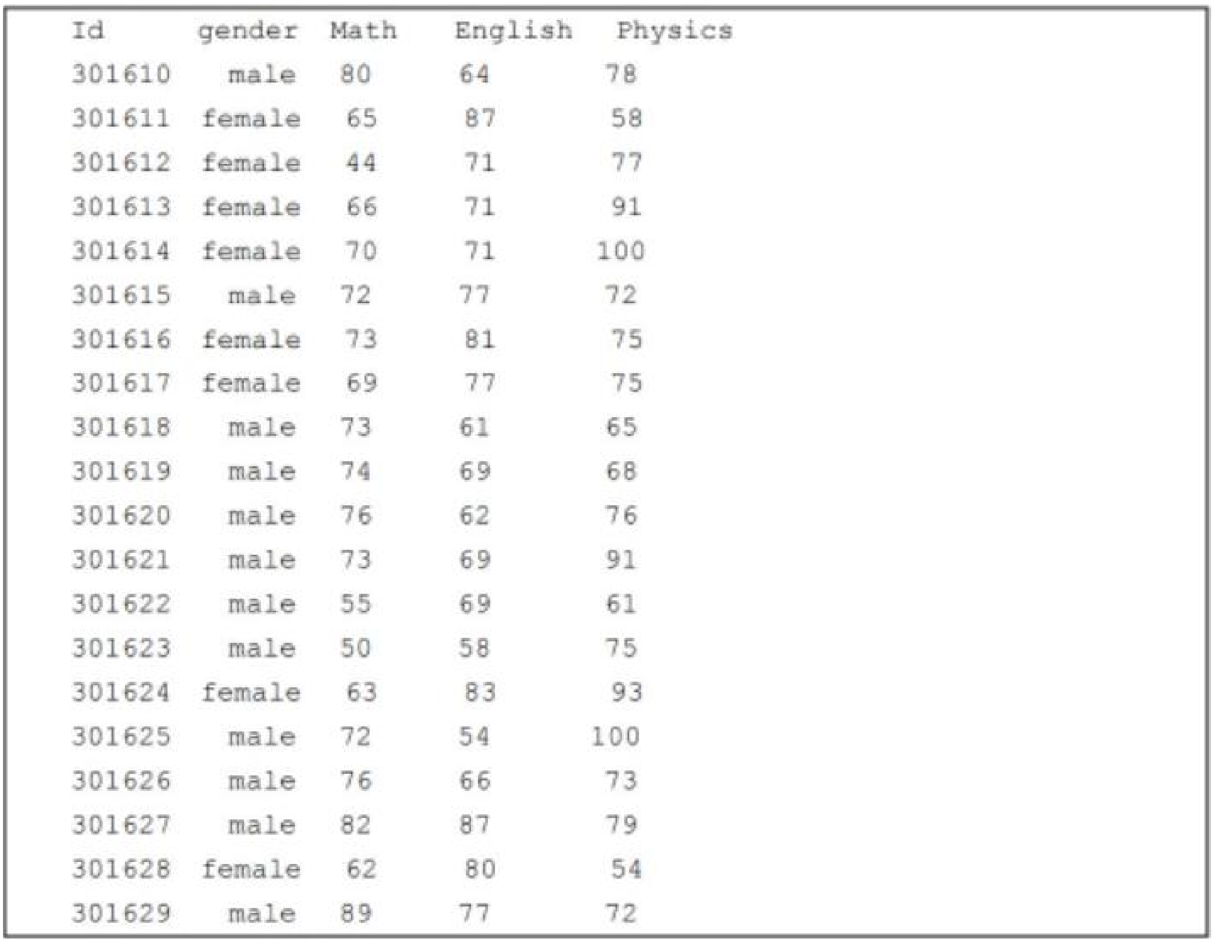
学生的成绩清单如下所示。第一行为表头，字段的意思分别为学号，性别，课程名 1，课程名 2 等，后面每一行代表一个学生信息，各字段之间用空格分开。学生数量不低于 10 行。

Id	**gender**	**Math**	**English**	**Physics**
301610	male	80	64	78
301611	female	65	87	58

……….

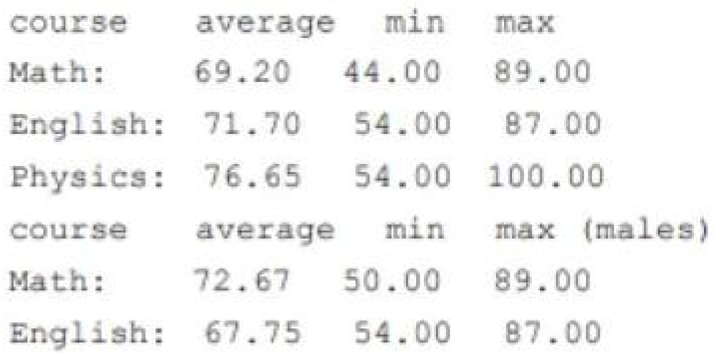
对于给定上述格式的成绩清单，要求采用函数式编程，统计出各门课程的平均成绩，最高分和最低分。然后按照男女学生分别统计每门课程的平均成绩，最高分和最低分。

成绩单数据集一：

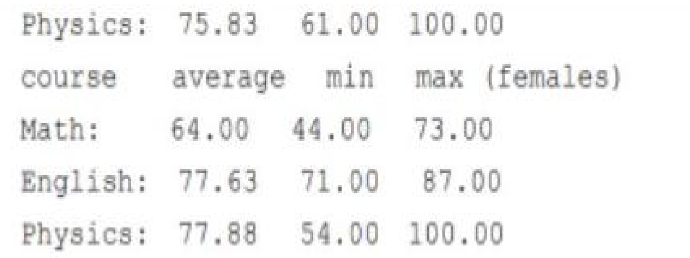


Id	gender	Math	English	Physics
301610	male	80	64	78
301611	female	65	87	58
301612	female	44	71	77
301613	female	66	71	91
301614	female	70	71	100
301615	male	72	77	72
301616	female	73	81	75
301617	female	69	77	75
301618	male	73	61	65
301619	male	74	69	68
301620	male	76	62	76
301621	male	73	69	91
301622	male	55	69	61
301623	male	50	58	75
301624	female	63	83	93
301625	male	72	54	100
301626	male	76	66	73
301627	male	82	87	79
301628	female	62	80	54
301629	male	89	77	72

输出结果为：

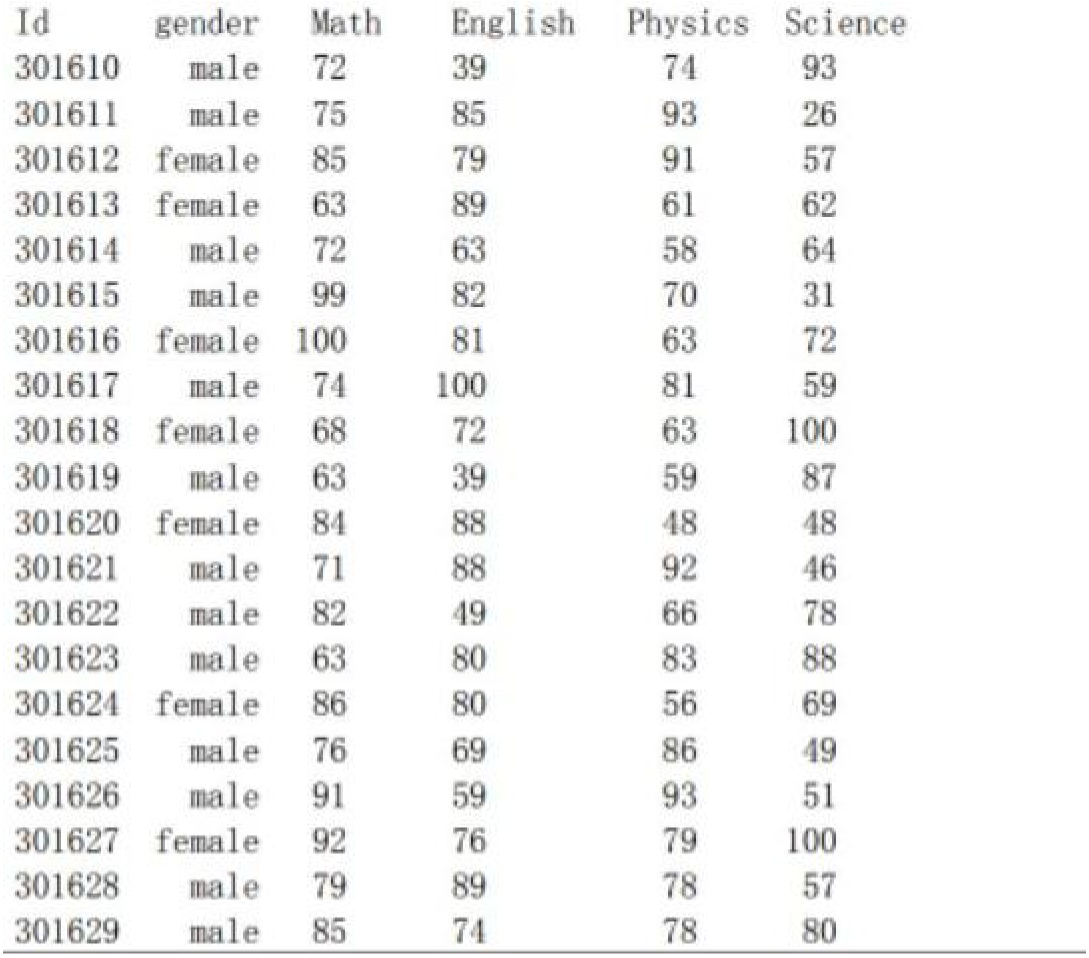


```
course      average   min    max
Math:       69.20    44.00   89.00
English:    71.70    54.00   87.00
Physics:    76.65    54.00  100.00
course      average   min    max (males)
Math:       72.67    50.00   89.00
English:    67.75    54.00   87.00
```



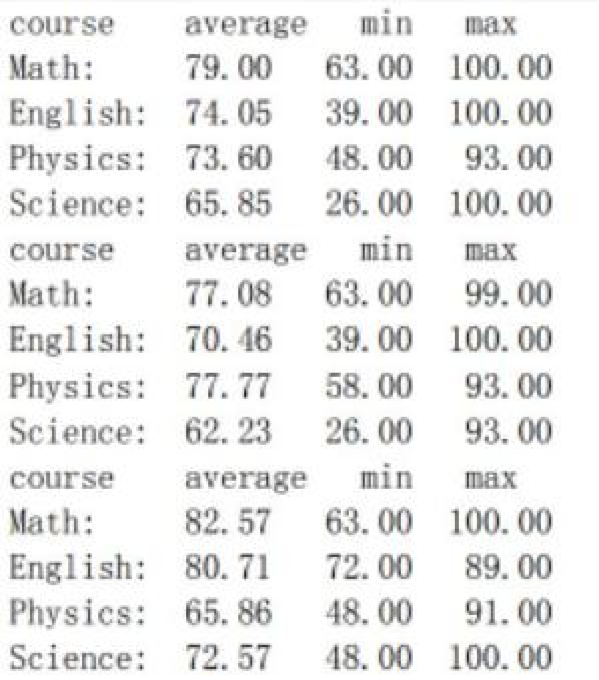
```
Physics:  75.83   61.00  100.00
course    average   min   max (females)
Math:     64.00   44.00   73.00
English:  77.63   71.00   87.00
Physics:  77.88   54.00  100.00
```

成绩单数据集二：



Id	gender	Math	English	Physics	Science
301610	male	72	39	74	93
301611	male	75	85	93	26
301612	female	85	79	91	57
301613	female	63	89	61	62
301614	male	72	63	58	64
301615	male	99	82	70	31
301616	female	100	81	63	72
301617	male	74	100	81	59
301618	female	68	72	63	100
301619	male	63	39	59	87
301620	female	84	88	48	48
301621	male	71	88	92	46
301622	male	82	49	66	78
301623	male	63	80	83	88
301624	female	86	80	56	69
301625	male	76	69	86	49
301626	male	91	59	93	51
301627	female	92	76	79	100
301628	male	79	89	78	57
301629	male	85	74	78	80

输出结果如下：

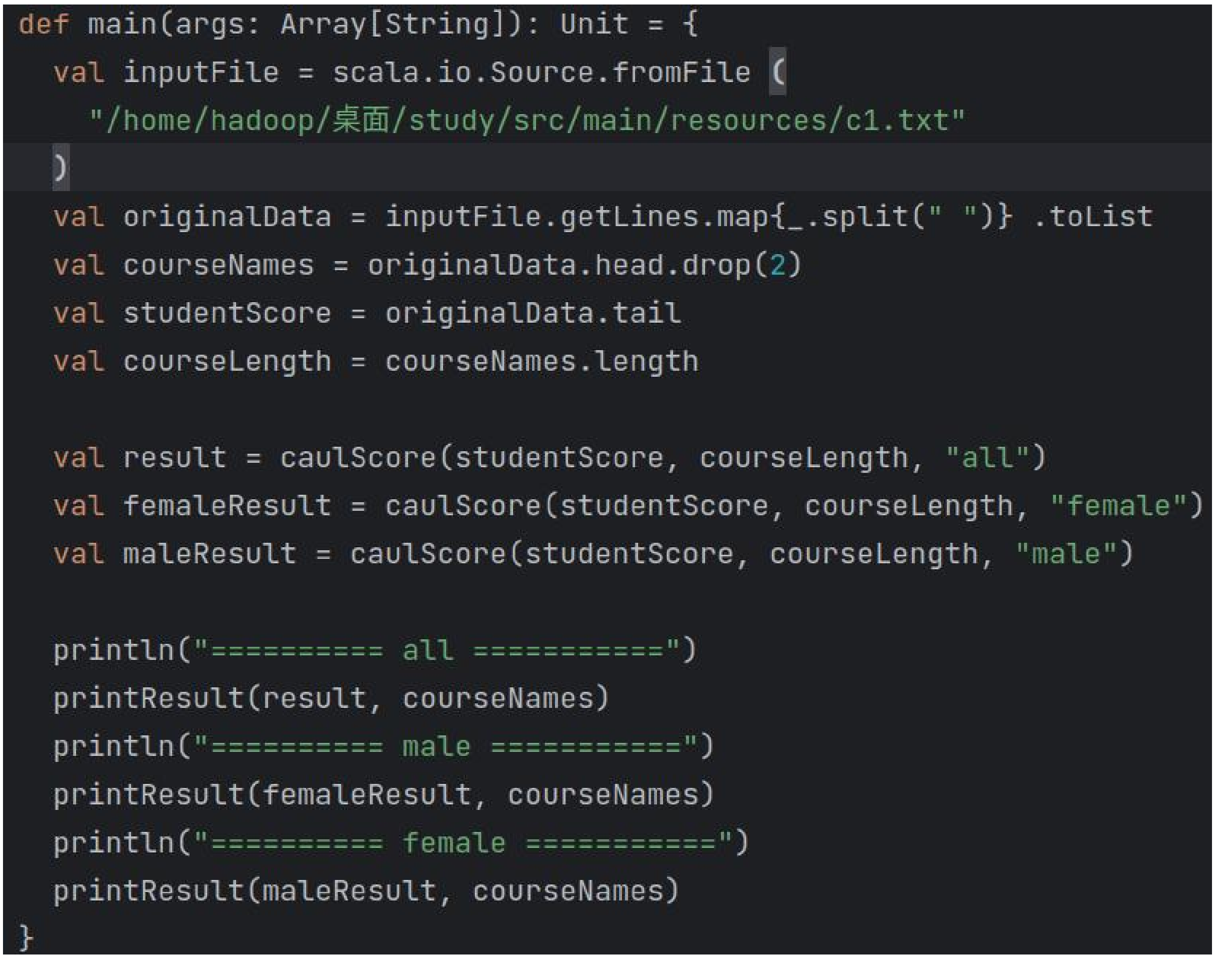


```
course     average   min     max
Math:      79.00     63.00   100.00
English:   74.05     39.00   100.00
Physics:   73.60     48.00    93.00
Science:   65.85     26.00   100.00
course     average   min     max
Math:      77.08     63.00    99.00
English:   70.46     39.00   100.00
Physics:   77.77     58.00    93.00
Science:   62.23     26.00    93.00
course     average   min     max
Math:      82.57     63.00   100.00
English:   80.71     72.00    89.00
Physics:   65.86     48.00    91.00
Science:   72.57     48.00   100.00
```

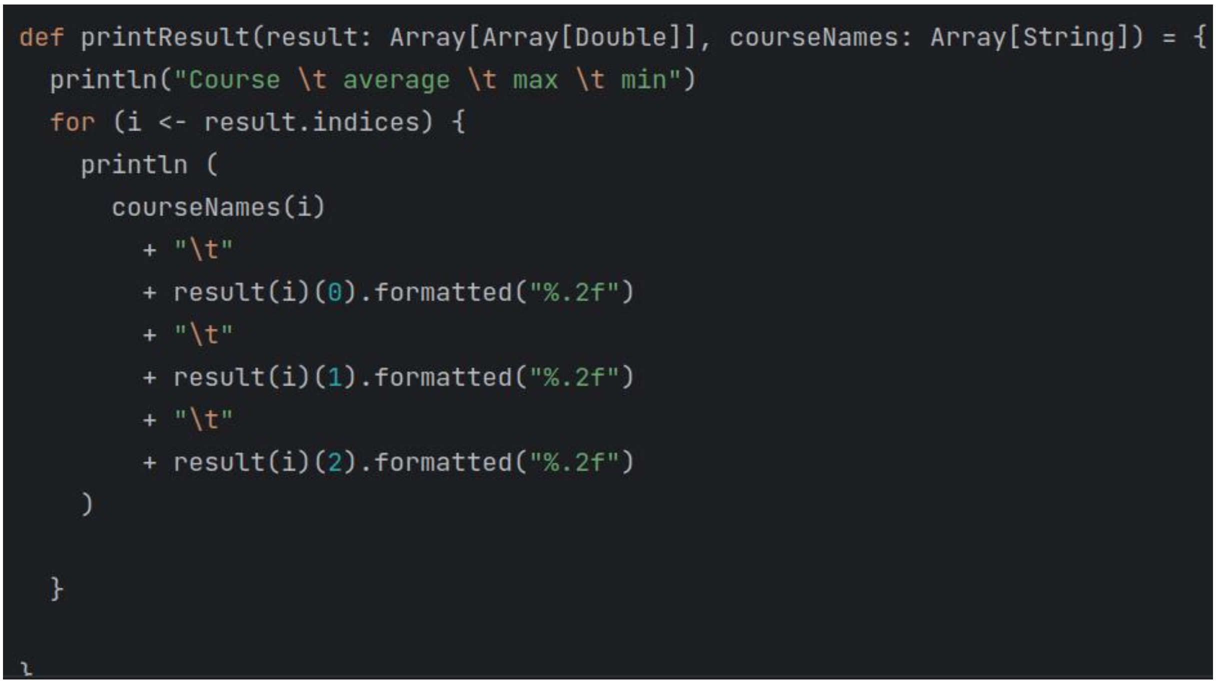
【实验过程】（步骤、记录、数据、程序等）

请提供相应代码及程序运行界面截图证明。

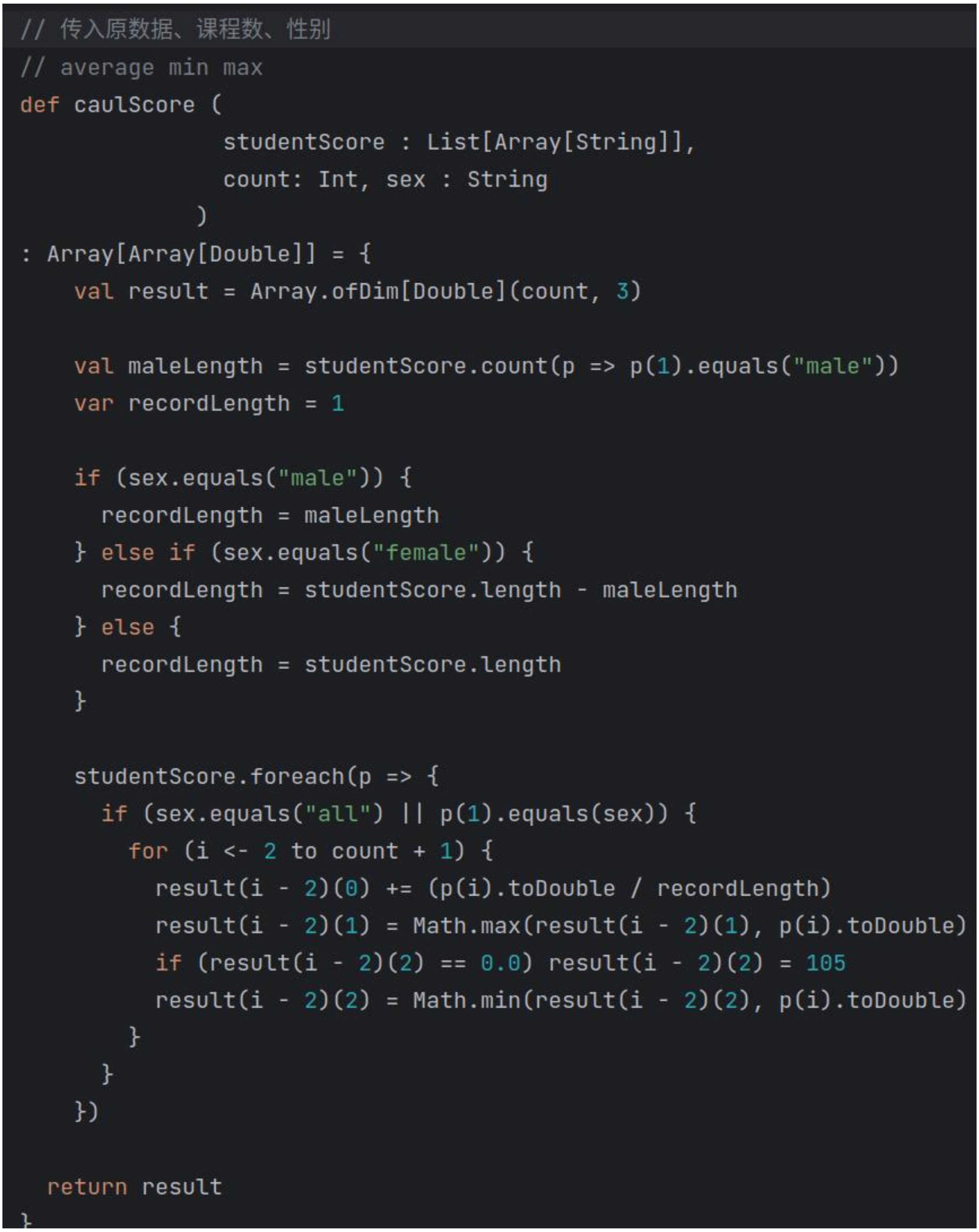
代码：



```
def main(args: Array[String]): Unit = {
  val inputFile = scala.io.Source.fromFile (
    "/home/hadoop/桌面/study/src/main/resources/c1.txt"
  )
  val originalData = inputFile.getLines.map{_.split(" ")} .toList
  val courseNames = originalData.head.drop(2)
  val studentScore = originalData.tail
  val courseLength = courseNames.length

  val result = caulScore(studentScore, courseLength, "all")
  val femaleResult = caulScore(studentScore, courseLength, "female")
  val maleResult = caulScore(studentScore, courseLength, "male")

  println("========== all ===========")
  printResult(result, courseNames)
  println("========== male ===========")
  printResult(femaleResult, courseNames)
  println("========== female ===========")
  printResult(maleResult, courseNames)
}
```



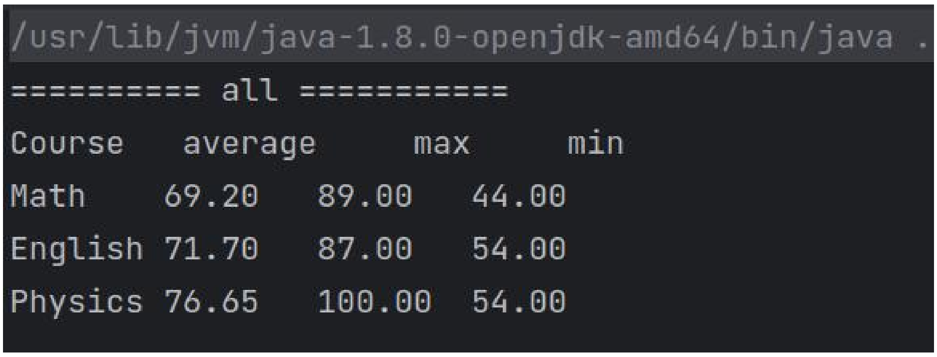
```
def printResult(result: Array[Array[Double]], courseNames: Array[String]) = {
  println("Course \t average \t max \t min")
  for (i <- result.indices) {
    println (
      courseNames(i)
        + "\t"
        + result(i)(0).formatted("%.2f")
        + "\t"
        + result(i)(1).formatted("%.2f")
        + "\t"
        + result(i)(2).formatted("%.2f")
    )

  }

}
```



```
// 传入原数据、课程数、性别
// average min max
def caulScore (
               studentScore : List[Array[String]],
               count: Int, sex : String
             )
: Array[Array[Double]] = {
    val result = Array.ofDim[Double](count, 3)

    val maleLength = studentScore.count(p => p(1).equals("male"))
    var recordLength = 1

    if (sex.equals("male")) {
      recordLength = maleLength
    } else if (sex.equals("female")) {
      recordLength = studentScore.length - maleLength
    } else {
      recordLength = studentScore.length
    }

    studentScore.foreach(p => {
      if (sex.equals("all") || p(1).equals(sex)) {
        for (i <- 2 to count + 1) {
          result(i - 2)(0) += (p(i).toDouble / recordLength)
          result(i - 2)(1) = Math.max(result(i - 2)(1), p(i).toDouble)
          if (result(i - 2)(2) == 0.0) result(i - 2)(2) = 105
          result(i - 2)(2) = Math.min(result(i - 2)(2), p(i).toDouble)
        }
      }
    })

  return result
}
```

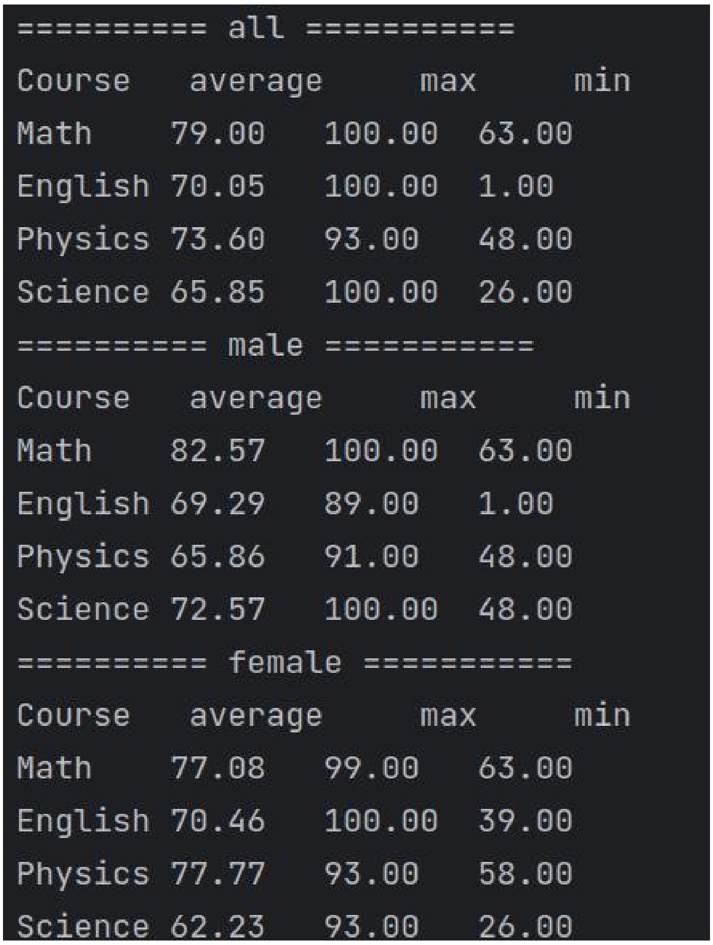
运行结果：

数据集 1



```
/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java .
========== all ===========
Course   average    max     min
Math     69.20    89.00   44.00
English  71.70    87.00   54.00
Physics  76.65    100.00  54.00
```

数据集 2



```
========== all ===========
Course   average    max     min
Math     79.00    100.00  63.00
English  70.05    100.00  1.00
Physics  73.60    93.00   48.00
Science  65.85    100.00  26.00
========== male ===========
Course   average    max     min
Math     82.57    100.00  63.00
English  69.29    89.00   1.00
Physics  65.86    91.00   48.00
Science  72.57    100.00  48.00
========== female ===========
Course   average    max     min
Math     77.08    99.00   63.00
English  70.46    100.00  39.00
Physics  77.77    93.00   58.00
Science  62.23    93.00   26.00
```

第 3 题 基于 Spark 的单词计数

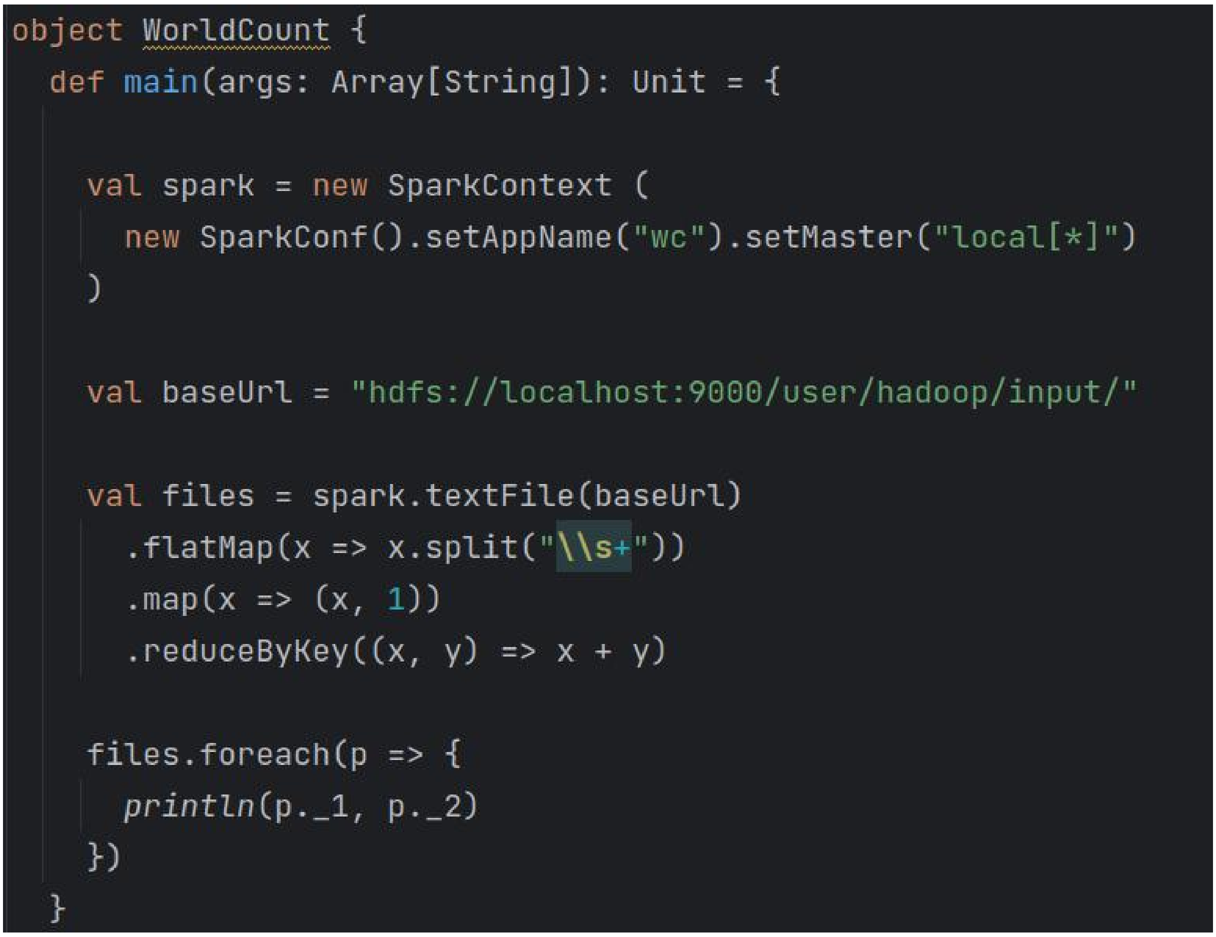
【实验内容】

针对 Mapreduce 实验的数据，使用 Spark 实现单词计数。

【实验过程】（步骤、记录、数据、程序等）

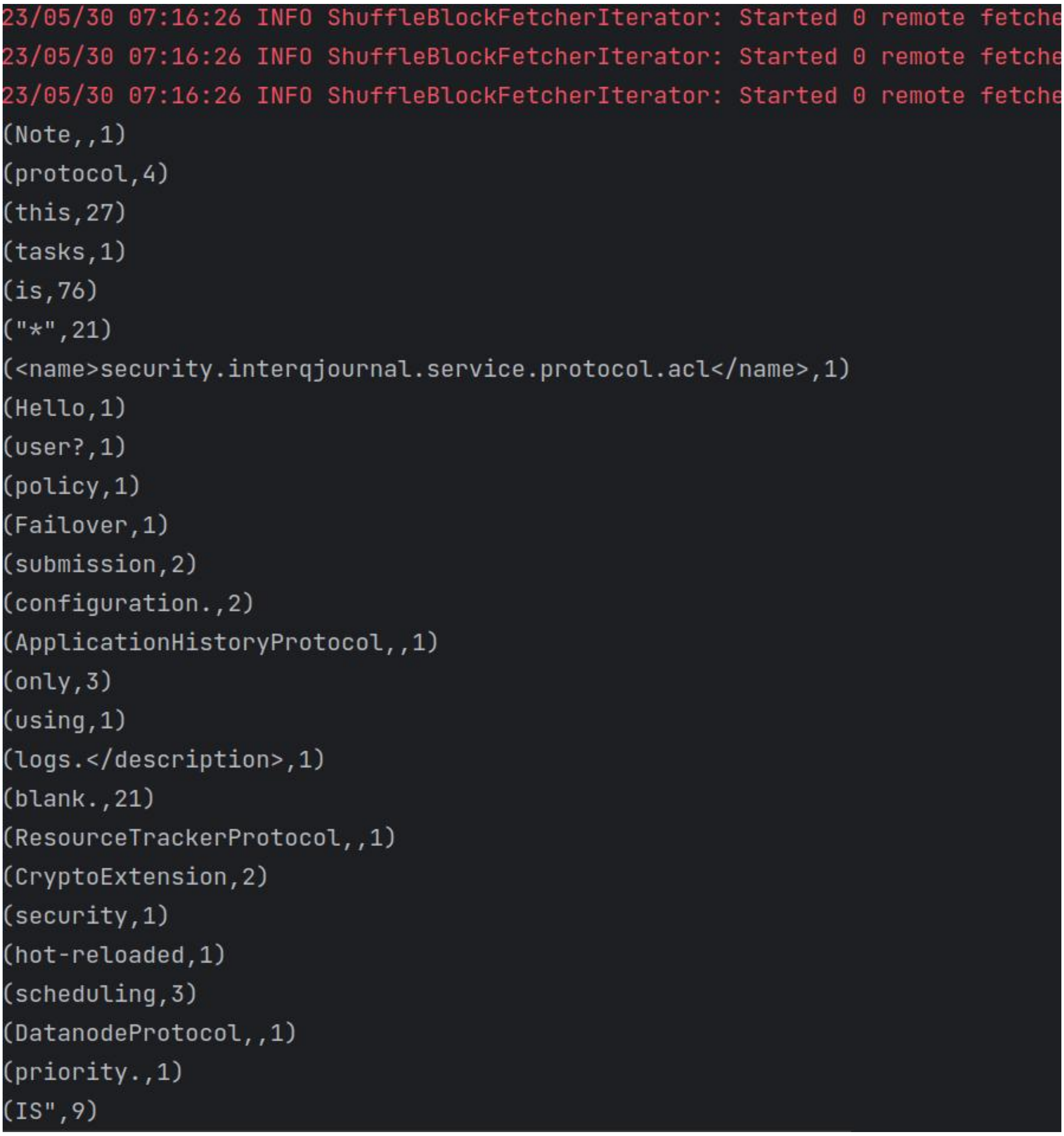
请提供相应的代码及程序运行界面截图证明。

代码：



```
object WorldCount {
  def main(args: Array[String]): Unit = {

    val spark = new SparkContext (
      new SparkConf().setAppName("wc").setMaster("local[*]")
    )

    val baseUrl = "hdfs://localhost:9000/user/hadoop/input/"

    val files = spark.textFile(baseUrl)
      .flatMap(x => x.split("\\s+"))
      .map(x => (x, 1))
      .reduceByKey((x, y) => x + y)

    files.foreach(p => {
      println(p._1, p._2)
    })
  }
```

运行结果



```
23/05/30 07:16:26 INFO ShuffleBlockFetcherIterator: Started 0 remote fetche
23/05/30 07:16:26 INFO ShuffleBlockFetcherIterator: Started 0 remote fetche
23/05/30 07:16:26 INFO ShuffleBlockFetcherIterator: Started 0 remote fetche
(Note,,1)
(protocol,4)
(this,27)
(tasks,1)
(is,76)
("*",21)
(<name>security.interqjournal.service.protocol.acl</name>,1)
(Hello,1)
(user?,1)
(policy,1)
(Failover,1)
(submission,2)
(configuration.,2)
(ApplicationHistoryProtocol,,1)
(only,3)
(using,1)
(logs.</description>,1)
(blank.,21)
(ResourceTrackerProtocol,,1)
(CryptoExtension,2)
(security,1)
(hot-reloaded,1)
(scheduling,3)
(DatanodeProtocol,,1)
(priority.,1)
(IS",9)
```

第 4 题 RDD 初级编程

【实验内容】

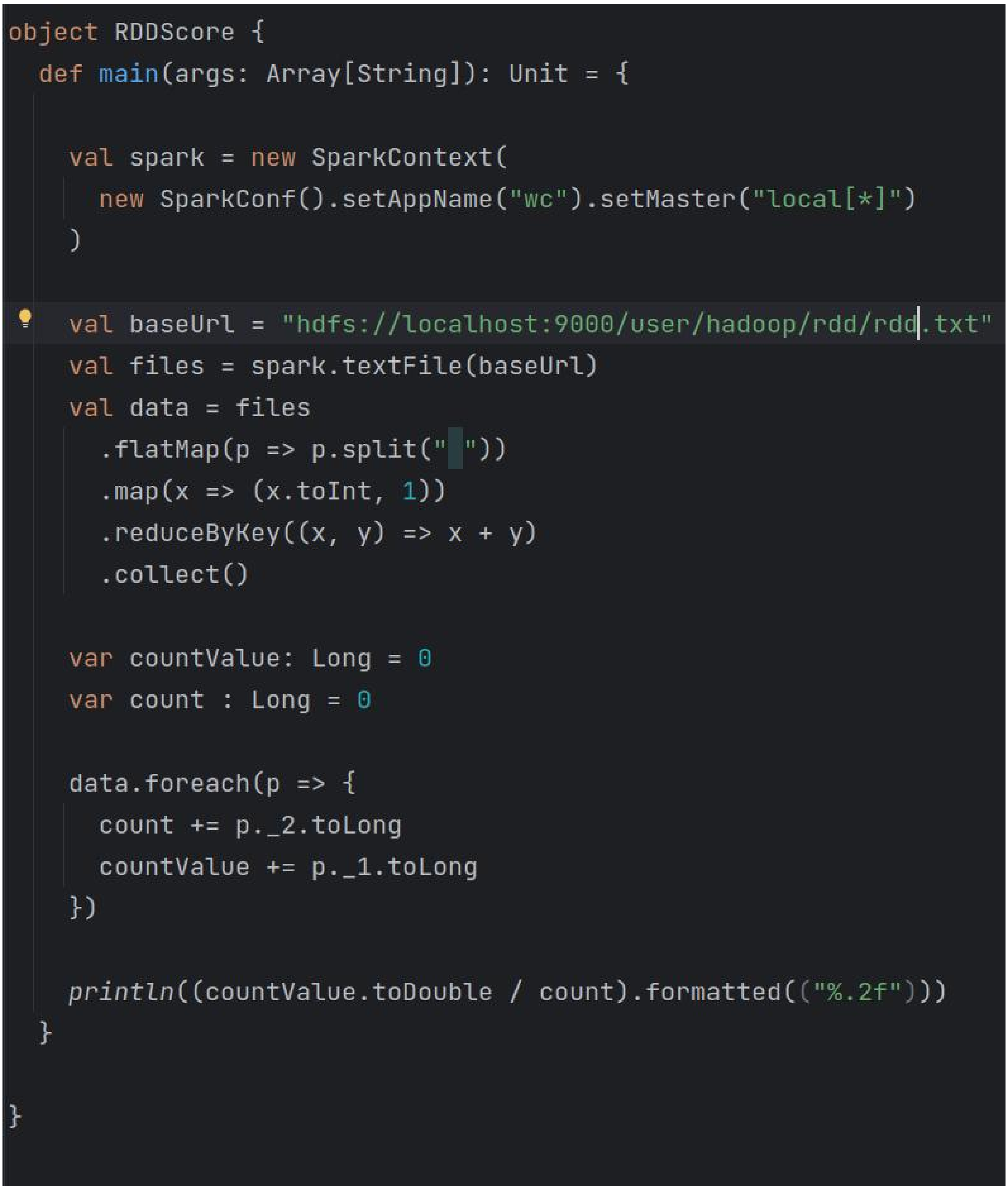
1. 在 RDD 读入数据{90，85，73，88，90}，通过 Spark 计算平均值并输出
2. RDD 读入数据：{“小明”:88}，{“魏芳”:70}，{“小明”:92}，{“魏芳”:83}分别统计每个人的总成绩和平均成绩并输出。

【实验过程】（步骤、记录、数据、程序等）

请提供相应代码及程序运行界面截图证明。

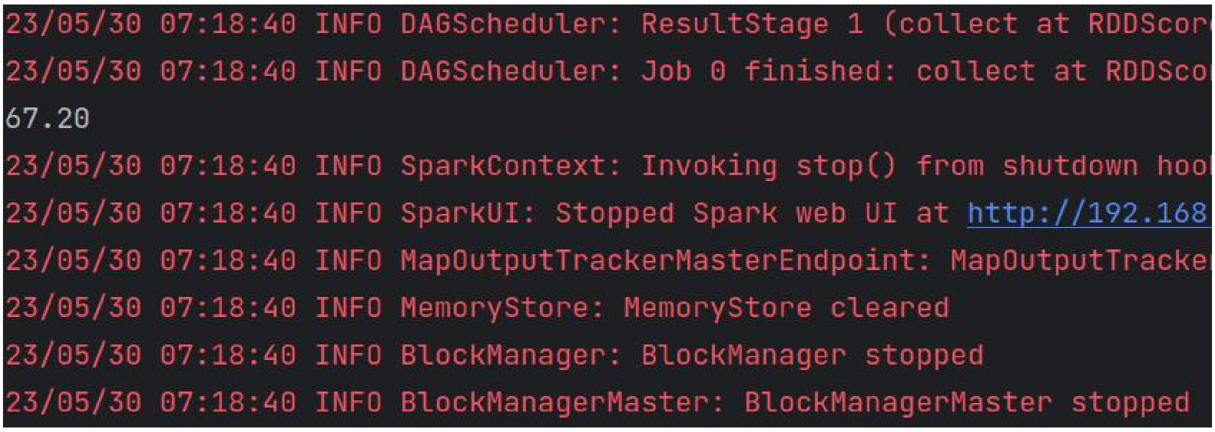
1.

代码



```
object RDDScore {
  def main(args: Array[String]): Unit = {

    val spark = new SparkContext(
      new SparkConf().setAppName("wc").setMaster("local[*]")
    )

    val baseUrl = "hdfs://localhost:9000/user/hadoop/rdd/rdd.txt"
    val files = spark.textFile(baseUrl)
    val data = files
      .flatMap(p => p.split(" "))
      .map(x => (x.toInt, 1))
      .reduceByKey((x, y) => x + y)
      .collect()

    var countValue: Long = 0
    var count : Long = 0

    data.foreach(p => {
      count += p._2.toLong
      countValue += p._1.toLong
    })

    println((countValue.toDouble / count).formatted(("%.2f")))
  }

}
```

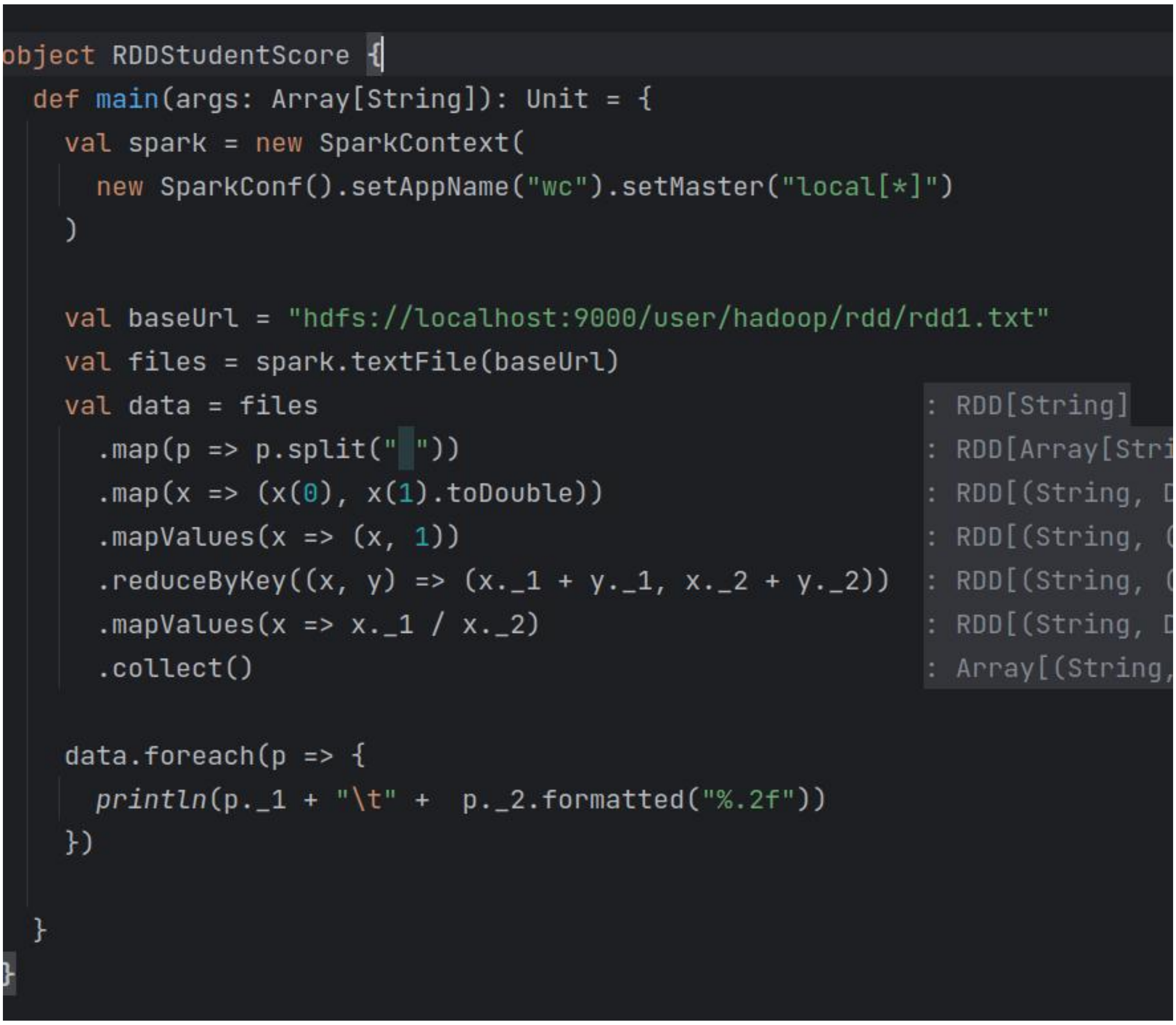
运行结果



```
23/05/30 07:18:40 INFO DAGScheduler: ResultStage 1 (collect at RDDScor
23/05/30 07:18:40 INFO DAGScheduler: Job 0 finished: collect at RDDSco
67.20
23/05/30 07:18:40 INFO SparkContext: Invoking stop() from shutdown hoo
23/05/30 07:18:40 INFO SparkUI: Stopped Spark web UI at http://192.168
23/05/30 07:18:40 INFO MapOutputTrackerMasterEndpoint: MapOutputTracke
23/05/30 07:18:40 INFO MemoryStore: MemoryStore cleared
23/05/30 07:18:40 INFO BlockManager: BlockManager stopped
23/05/30 07:18:40 INFO BlockManagerMaster: BlockManagerMaster stopped
```

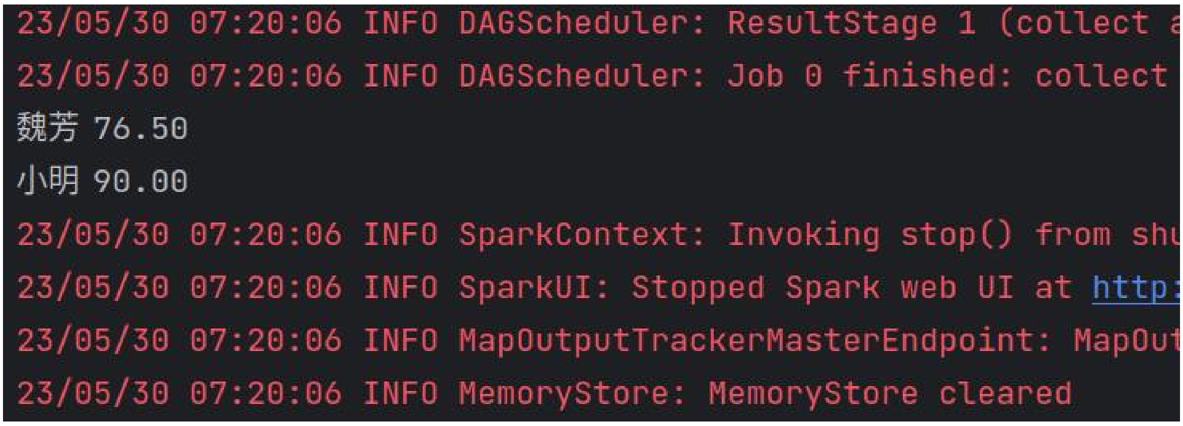
2.

代码：



```
object RDDStudentScore {
  def main(args: Array[String]): Unit = {
    val spark = new SparkContext(
      new SparkConf().setAppName("wc").setMaster("local[*]")
    )

    val baseUrl = "hdfs://localhost:9000/user/hadoop/rdd/rdd1.txt"
    val files = spark.textFile(baseUrl)
    val data = files                                                      : RDD[String]
      .map(p => p.split(" "))                                             : RDD[Array[Stri
      .map(x => (x(0), x(1).toDouble))                                    : RDD[(String, D
      .mapValues(x => (x, 1))                                             : RDD[(String, (
      .reduceByKey((x, y) => (x._1 + y._1, x._2 + y._2))                  : RDD[(String, (
      .mapValues(x => x._1 / x._2)                                        : RDD[(String, D
      .collect()                                                          : Array[(String,

    data.foreach(p => {
      println(p._1 + "\t" +  p._2.formatted("%.2f"))
    })

  }
}
```

运行结果：



```
23/05/30 07:20:06 INFO DAGScheduler: ResultStage 1 (collect a
23/05/30 07:20:06 INFO DAGScheduler: Job 0 finished: collect
魏芳 76.50
小明 90.00
23/05/30 07:20:06 INFO SparkContext: Invoking stop() from shu
23/05/30 07:20:06 INFO SparkUI: Stopped Spark web UI at http:
23/05/30 07:20:06 INFO MapOutputTrackerMasterEndpoint: MapOut
23/05/30 07:20:06 INFO MemoryStore: MemoryStore cleared
```