Лабораторная работа 5.

Работа с Spark

1. Входные данные.

Аналогичны входным данным лабораторных работ 3,4

2. Задача

Требуется определить для пары <аэропорт отлета, аэропорт прибытия> максимальное время опоздания, процент опоздавших+отмененных рейсов.

Также требуется связать полученную таблицу с названиями аэропортов.

3. Запуск Spark

Распаковываем в корневую папку пользователя архив

spark-2.0.1-bin-hadoop2.7.tgz

добавляем в ~/.bashrc строку

export PATH=~/spark-2.0.1-bin-hadoop2.7/bin/:$PATH

Для запуска задачи spark в дальнейшем используем строку

spark-submit --class <Имя класса с main функцией> --master yarn-client --num-executors 3 <путь к jar файлу>

пример pom.xml файла с настроенными dependencies приложен к проекту

3. Посдказки.

а. Инициализируем Spark

```
SparkConf conf = new SparkConf().setAppName("lab5");
JavaSparkContext sc = new JavaSparkContext(conf);
```

б. Загружаем исходные наборы данных в RDD с помощью метода

```
JavaSparkContext.textFile
```

```
в. Преобразуем RDD в RDD пару ключ значение с помощью метода
mapToPair
```

```
г. Создаем Java объекты для хранения данных – простые объекты
реализующие интерфейс Serializable
```

```
д. В качестве ключа для пары аэропортов используем класс Tuple2
```

```
с помощью функции reduce или аналогичных расчитываем  максимальное
время опоздания, процент опоздавших+отмененных рейсов
```

е. для связывания с таблицей аэропортов – предварительно выкачиваем список аэропортов в главную функцию с помощью метода collectAsMap

ё. создаем в основном методе main переменную broadcast

```
final Broadcast<Map<String, AirportData>> airportsBroadcasted = sc.broadcast(stringAirportDataMap);
```

ж. в методе map преобразуем итоговый RDD содержащий статистические данные – обогащаем его именами аэропортов, обращаясь внутри функций к объекту `airportsBroadcasted.value()`