|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6) \_\_\_\_\_\_

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/12 Интеллектуальный анализ больших**

**данных в системах поддержки принятия решений**

**Отчет**

**по лабораторной работе № 10**

**Название:** Spark

**Дисциплина:** Языки программирования для работы с большими данными

Студент гр. ИУ6-23М **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_** Г. Е. Горский **\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_**П.В. Степанов **\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2023

**Цель:** ознакомиться с базовыми принципами языка Java для работы с большими данными

**Вариант 1:**

1. Выбрать любой датасет на kaggle.com
2. Сделать 10 выборок данных по выбранной предметной области

**Решение:**

Добавим Dataset c сайта kaggle.com. Dataset представлен на рисунке 1.

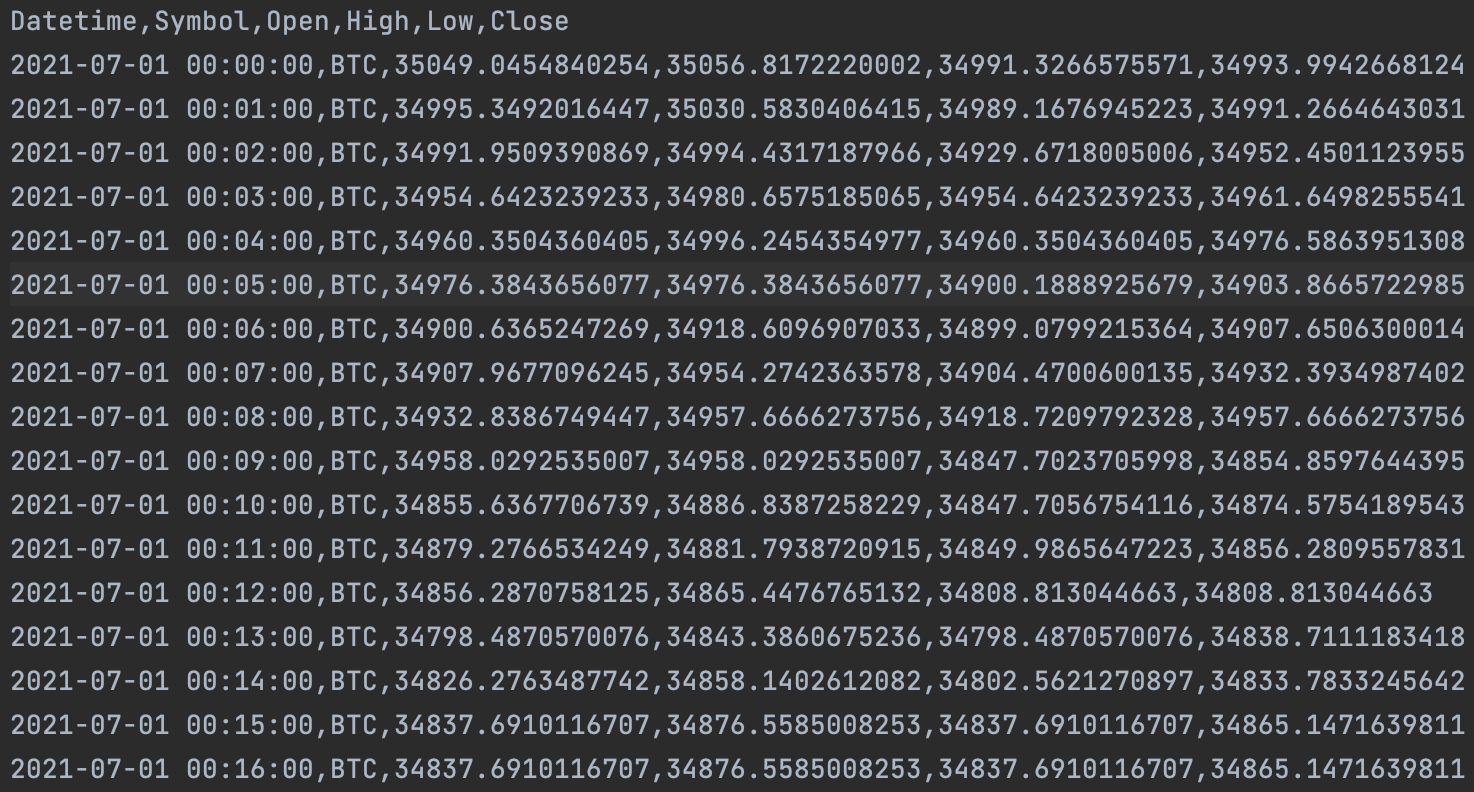


Рисунок 1 — Dataset

package org.example;  
import org.apache.spark.SparkConf;  
import org.apache.spark.SparkContext;  
import org.apache.spark.api.java.JavaSparkContext;  
import org.apache.spark.sql.Dataset;  
import org.apache.spark.sql.Row;  
import org.apache.spark.sql.SparkSession;  
public class Main {  
 public static void main(String[] args) {  
 SparkSession spark = SparkSession.builder()  
 .appName("RandomSplitExample")  
 .master("local[\*]")  
 .getOrCreate();  
 Dataset<Row> data = spark.read().format("csv")  
 .option("header", "true")  
 .load("/Users/lol/Desktop/untitled/Lab10\_\_\_/coinbase-extracted-second.csv");  
 data.createOrReplaceTempView("coinbase");  
 spark.sql("select \* from coinbase ").show();  
 spark.sql("select \* from coinbase where coinbase.Symbol='USDT'").show();  
 spark.sql("select coinbase.Symbol, AVG(coinbase.Open) from coinbase group by coinbase.Symbol ").show();  
 spark.sql("select coinbase.Symbol, date\_trunc('week',coinbase.Datetime) as weeks, sum(coinbase.Open ) as total\_usd from coinbase group by 1,2").show();  
 spark.sql("select \* from coinbase where coinbase.Symbol='USDT' order by coinbase.High desc").show();  
 spark.sql("select \* from coinbase order by 4").show();  
 spark.sql("select coinbase.Datetime, coinbase.Symbol, coinbase.Close from coinbase where coinbase.Symbol='USDT' and coinbase.Low < 1 group by 1,2,3 order by 1").show();  
 spark.sql("select coinbase.Symbol, date\_trunc('week',coinbase.Datetime) as weeks, sum(coinbase.Open ) as total\_usd from coinbase where coinbase.Symbol in ('USDC', 'USDT', 'DAI','BUSD','TUSD','FRAX','USDP','USDD') group by 1,2").show();  
 // Останавливаем SparkSession  
 spark.stop();  
 }  
}

Результат выполнения представлен на рисунке 2.

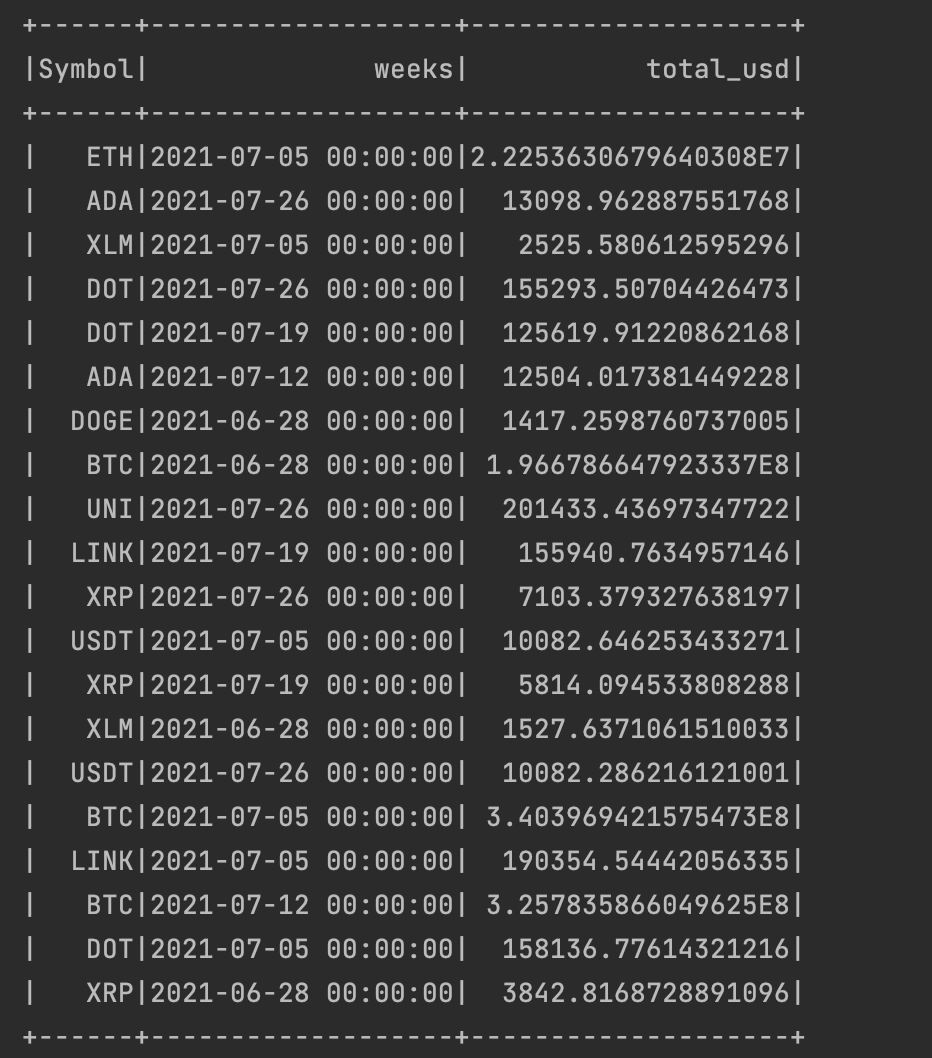
****

Рисунок 2 — пример выполнения

**Вывод:** были разработаны классы и методы согласно вариантам.