Министерство образования Республики Беларусь

Учреждение высшего образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ОТЧЕТ

по лабораторной работе №3

на тему

**Работа с пакетом SPARK**

|  |  |
| --- | --- |
| Студенты группы 150504  Проверила | Шевцов В.С  Крук А.И  Герман Ю.О. |

Минск 2023

Цель: Изучить технику обработки строк spark scala.

**Краткие теоретические сведения**.

Будем использовать теоретический материал курса. Вот основные выдержки по SPARK.

Нужно распаковать загруженную библиотеку.

Чтобы запустить SPARK, нужно открыть командное окно в режиме администратора и ввести команду

**C:\Users\scotn\Downloads\spark-3.4.0-bin-hadoop3\bin\spark-shell**

**c:\Users\HP\AppData\Local\Coursier\data\bin\spark\bin\spark-shell**

Имея в распоряжении объект spark и sc, можно двигаться дальше.

Создаем журнал регистрации ошибок

scala> spark.sparkContext.setLogLevel("ERROR")

Это вообще-то нам не очень-то и нужно.

Далее создаем объект-конференции

scala> val conf = new org.apache.spark.SparkConf()

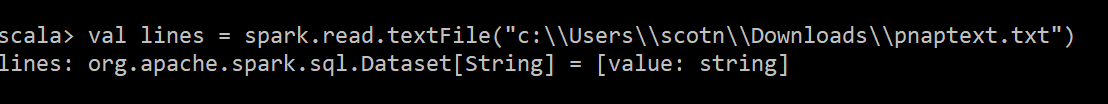
conf: org.apache.spark.SparkConf = [org.apache.spark.SparkConf@1e2a39b7](mailto:org.apache.spark.SparkConf@1e2a39b7)

Вот теперь мы читаем текстовый файл:

scala> val lines = spark.read.textFile("c:\\Users\\scotn\\Downloads\\pnaptext.txt")

Прописываем полный путь. Да еще дублируем симол-разделитель - \\

Строки прочтены

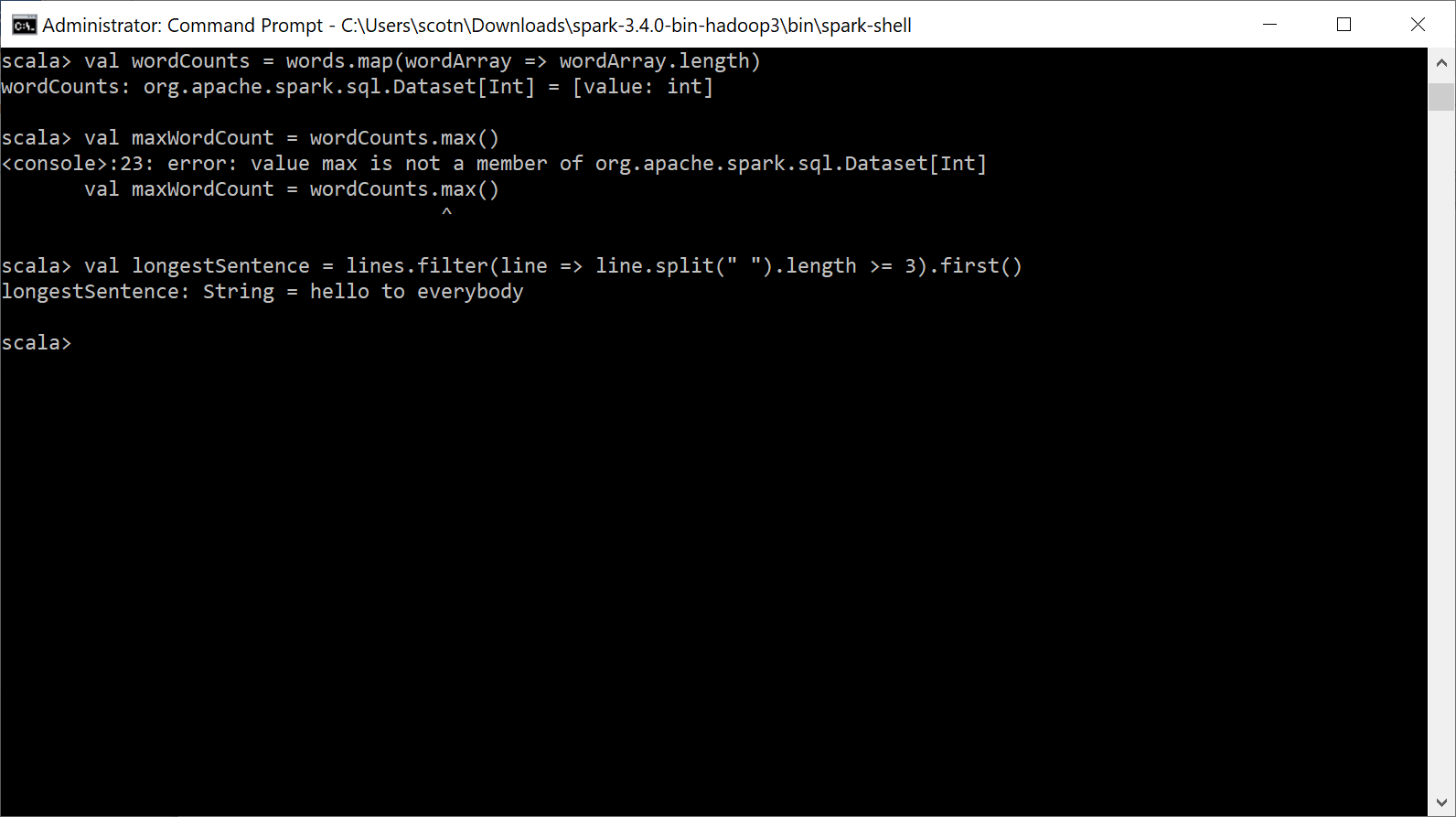


Получаем массив всех строк

scala> val words = lines.map(line => line.split(" "))

words: org.apache.spark.sql.Dataset[Array[String]] = [value: array<string>]

Выводим предложение, содержащее не менее трех слов



В нашем текстовом файле хранится такой текст

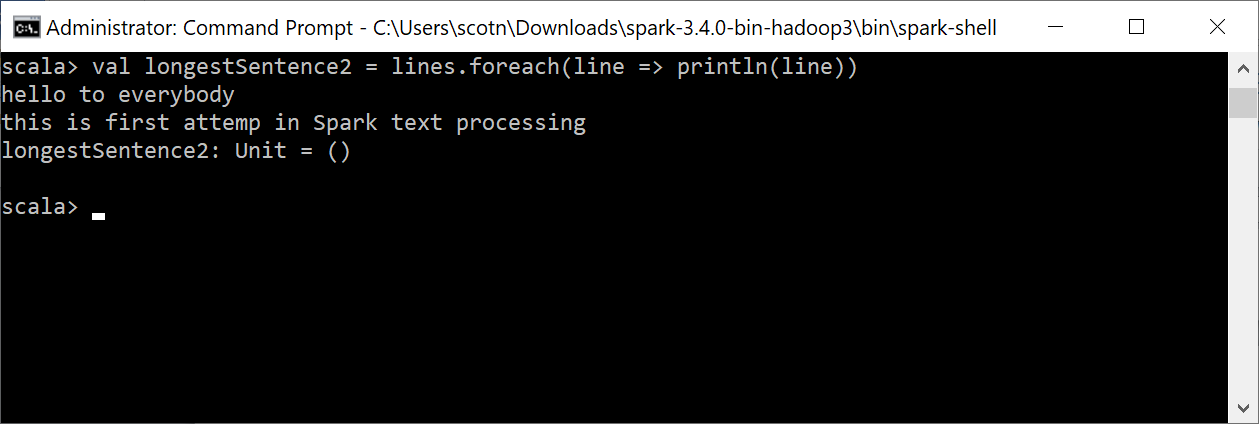
hello to everybody

this is first attemp in Spark text processing

Be careful with people No all of them sympthatize you

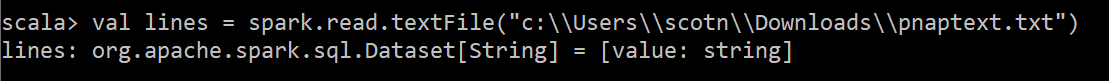
Выводим строки из текстового файла

val longestSentence2 = lines.foreach(line => println(line))

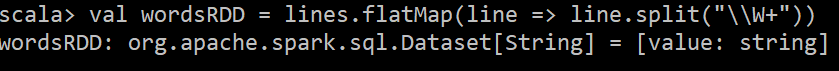


**Повторение.**

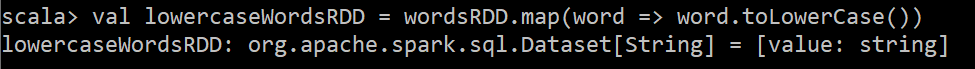
Начинаем с чтения текстового файла



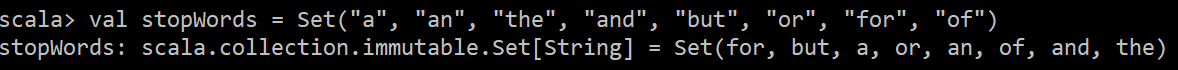
**Расщепляем текст на отдельные слова.** При этом используем регулярное выражение\\W+ (означает любую последовательность символов)

****

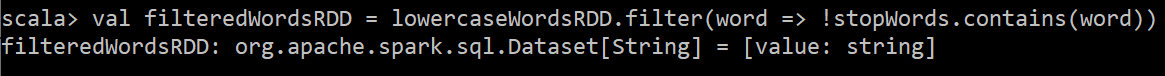
Слова приводим к нижнему регистру



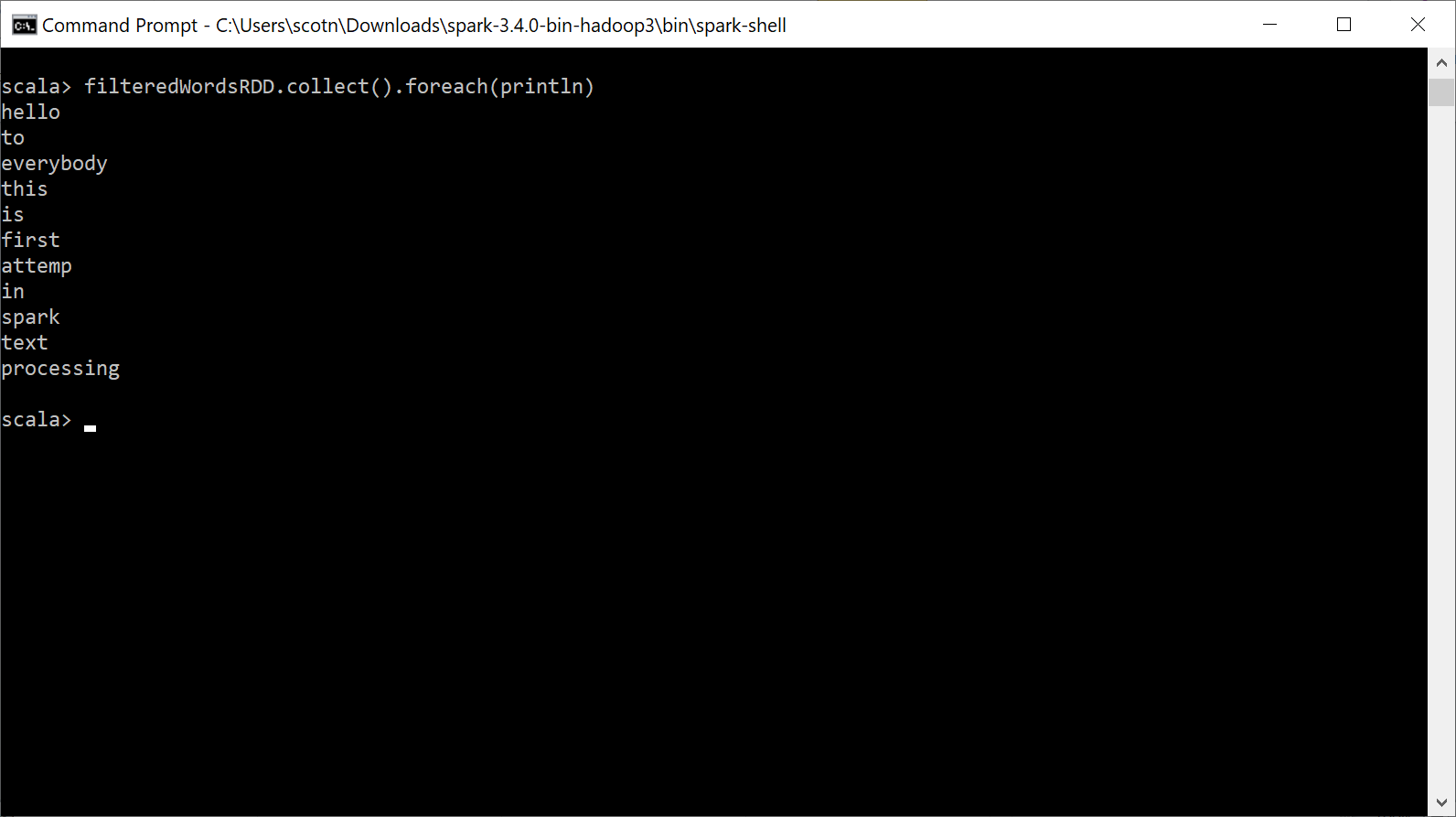
Создаем множество stop-слов

****

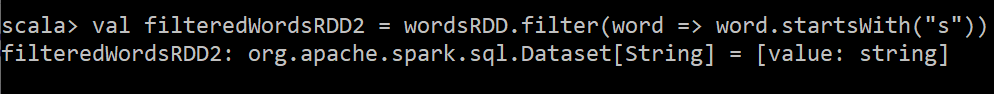
Удаляем стоп-слова из текста



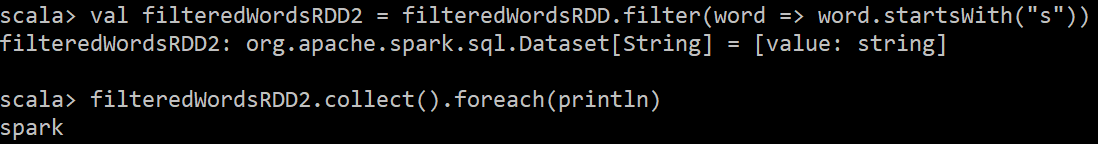
Выводим отфильтрованное множество слов



Отфильтруем слова, начинающиеся на s:



Выведем их



Итак, мы научились формировать множества слов и фильтровать их.

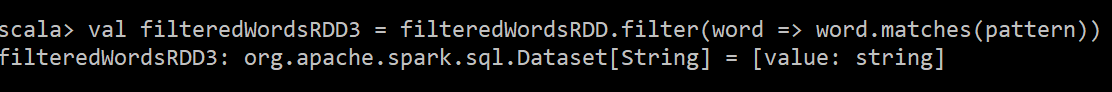
Познакомимся чуть ближе с регулярными выражениями

Создадим паттерн

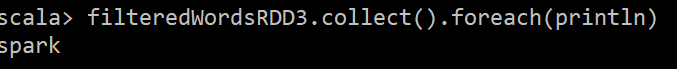


Этот паттерн определяет все слова, начинающиеся на s, содержащие где-то посередине r и заканчивающиеся на k

Сформируем отфильтрованное множество по паттерну



Выведем результат на консоль



**Задание**

**Вариант 2.**

Создать собственный текстовый файл на английском или немецком языке – **4-5** предложений. Тема - программирование

1. Вывести все слова из текстового файла, исключая stop-слова

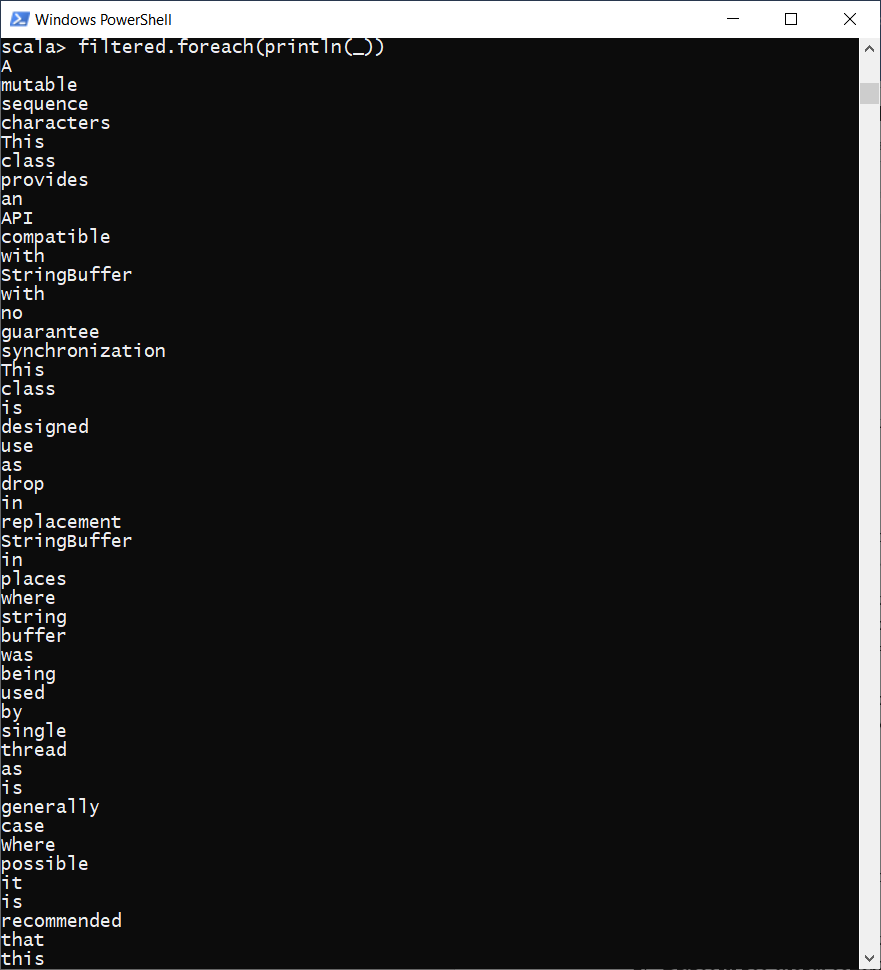
val text = spark.read.textFile("D:\\Homework\\Labs\\Scala\\textFile.txt")

val stopWords = Set("a", "and", "for", "or", "of", "but", "the")

var filtered = text.flatMap(line => line.split("\\W+"))

filtered = filtered.filter(word => !stopWords.contains(word))

filtered.foreach(println(\_))

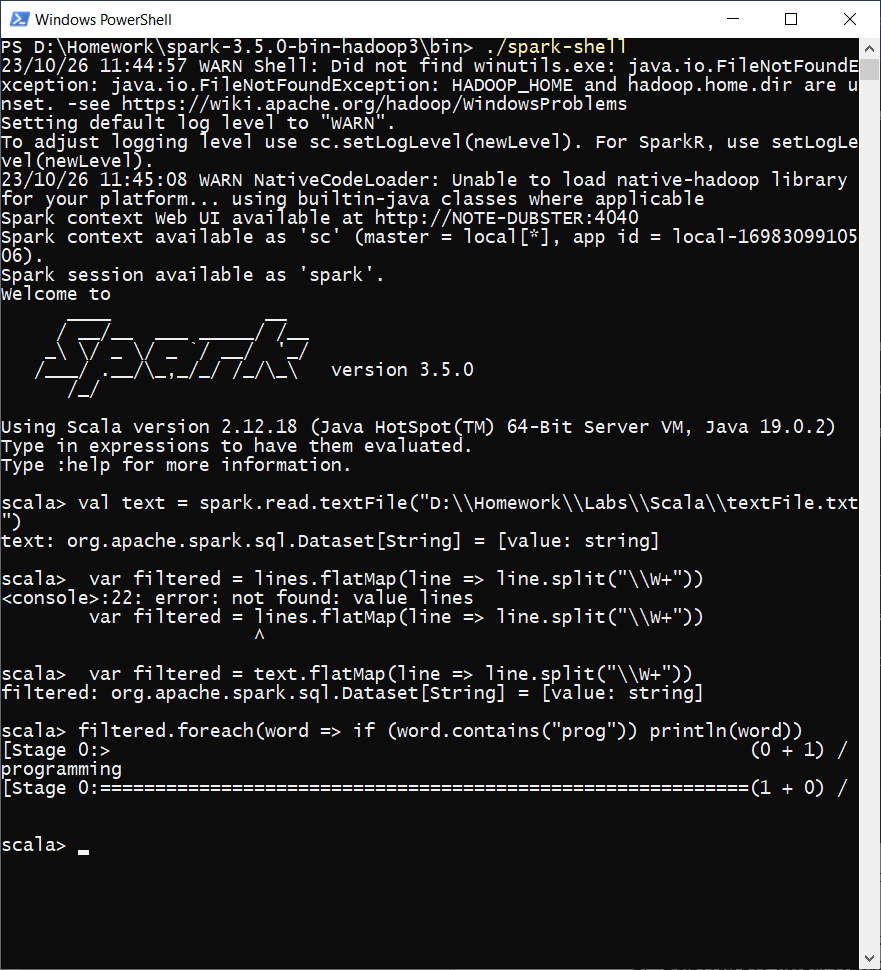


1. Вывести все слова, содержащие вхождение prog

val text = spark.read.textFile("D:\\Homework\\Labs\\Scala\\textFile.txt")

var filtered = text.flatMap(line => line.split("\\W+"))

filtered.foreach(word => if (word.contains("prog")) println(word))

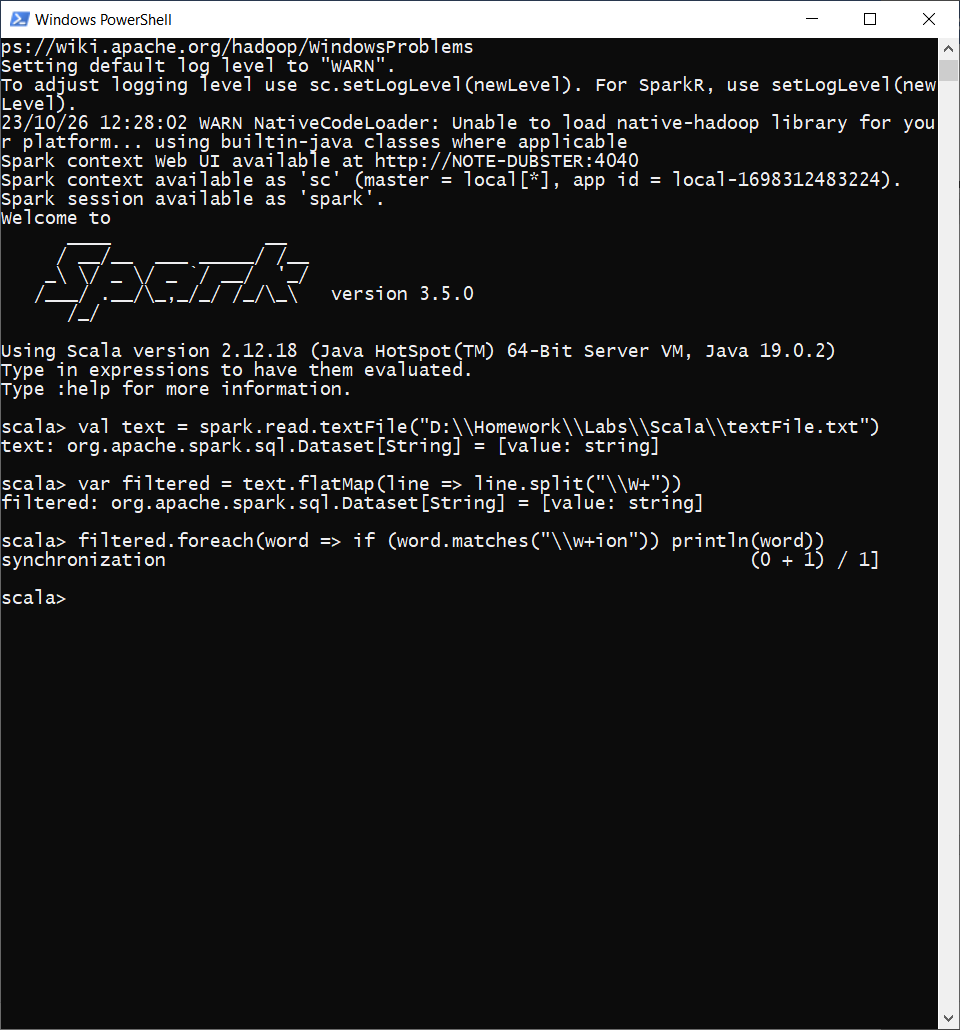


1. Вывести все слова, заканчивающиеся на ion

val text = spark.read.textFile("D:\\Homework\\Labs\\Scala\\textFile.txt")

var filtered = text.flatMap(line => line.split("\\W+"))

filtered.foreach(word => if (word.contains("prog")) println(word))

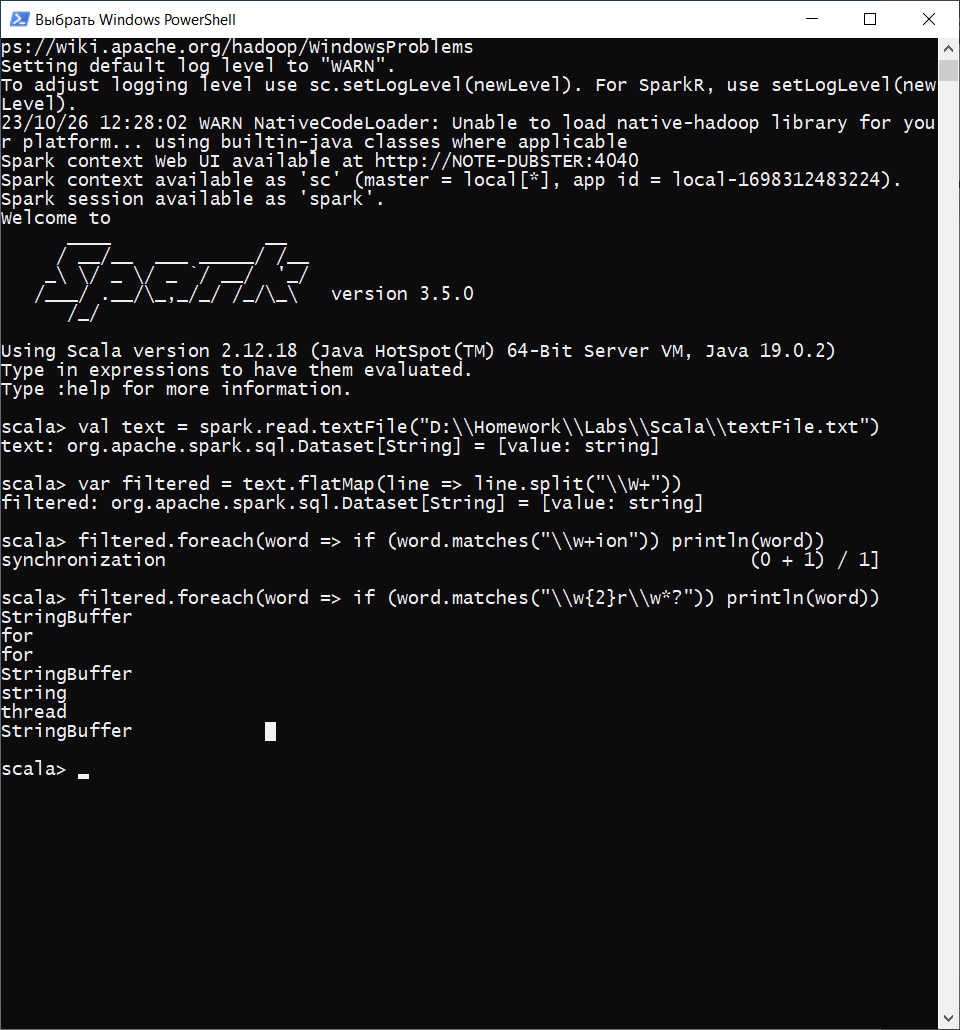


1. Вывести все слова третья буква которых r

val text = spark.read.textFile("D:\\Homework\\Labs\\Scala\\textFile.txt")

var filtered = text.flatMap(line => line.split("\\W+"))

filtered.foreach(word => if (word.matches("\\w+ion")) println(word))

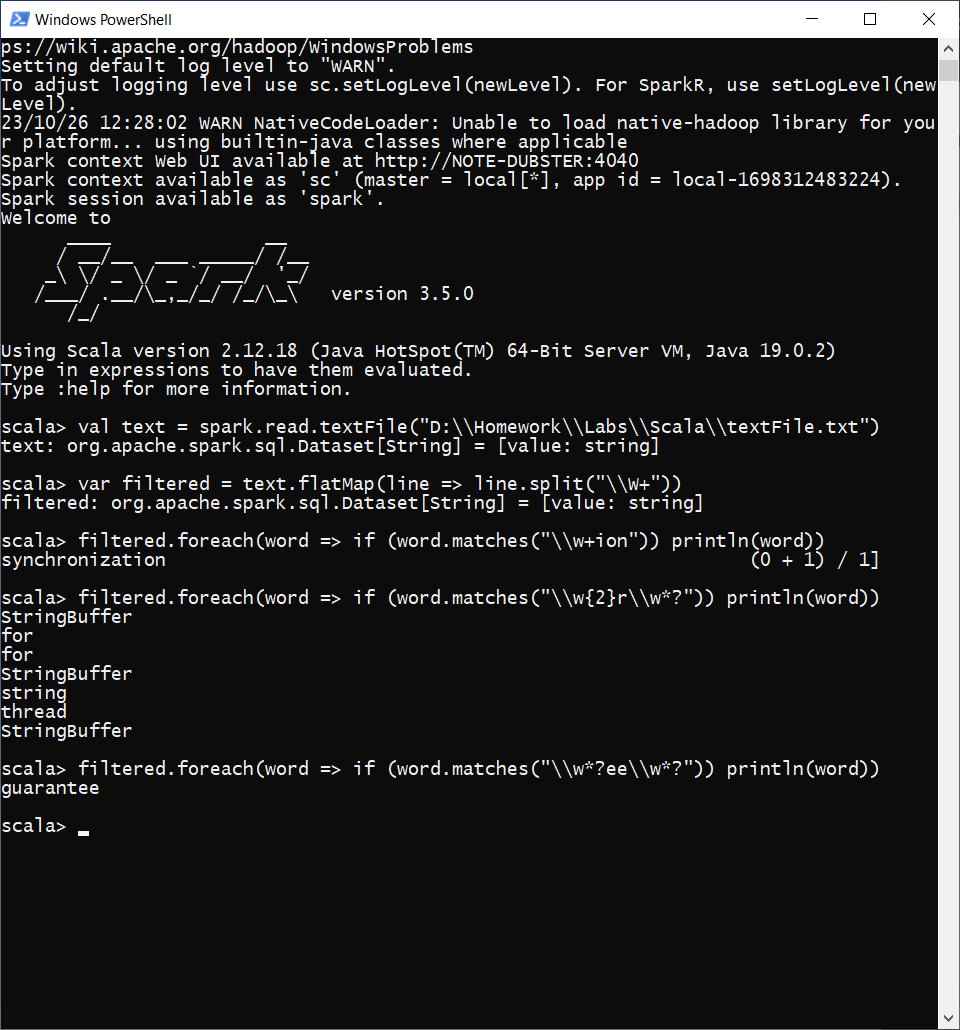


1. Вывести все слова, содержащие как минимум две буквы e

val text = spark.read.textFile("D:\\Homework\\Labs\\Scala\\textFile.txt")

var filtered = text.flatMap(line => line.split("\\W+"))

filtered.foreach(word => if (word.matches("\\w\*?ee\\w\*?")) println(word))



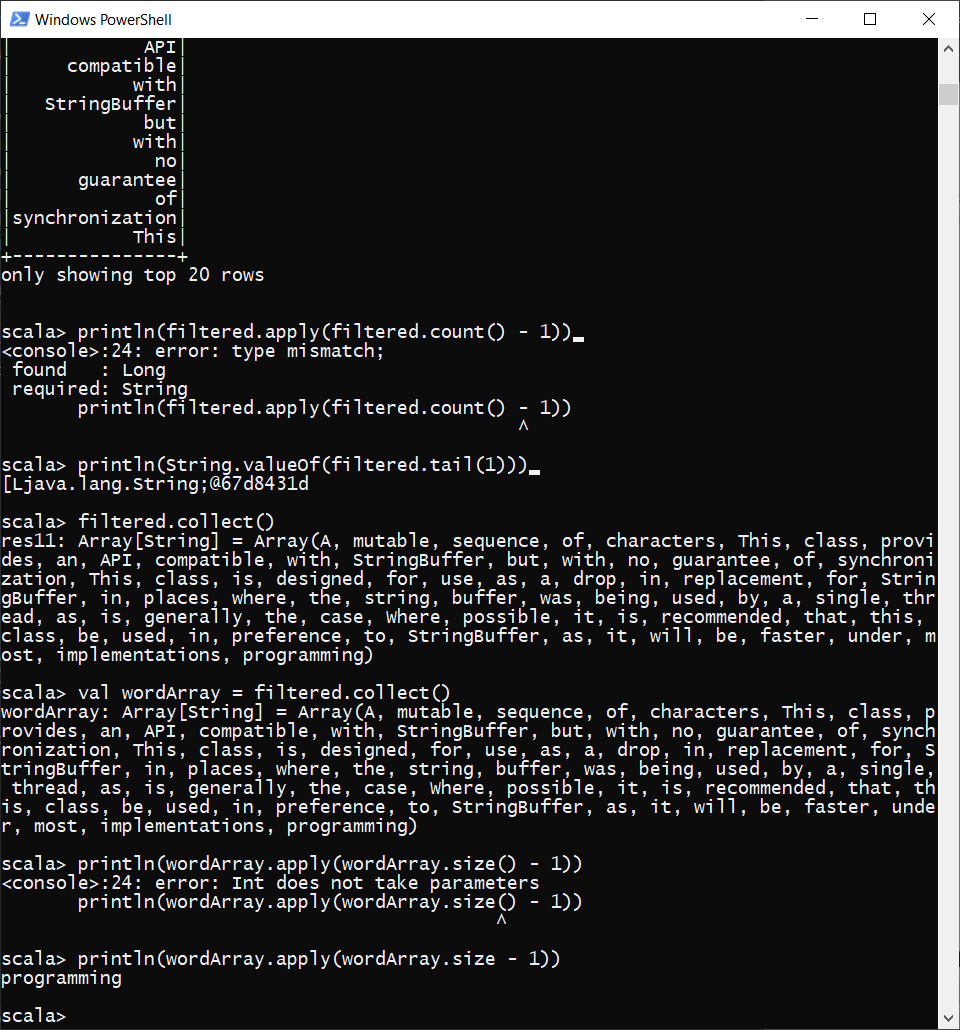
1. Вывести последнее слово

val text = spark.read.textFile("D:\\Homework\\Labs\\Scala\\textFile.txt")

var filtered = text.flatMap(line => line.split("\\W+"))

val wordArray = filtered.collect()

println(wordArray.apply(wordArray.size - 1))



**Заключение**

В ходе лабораторной работы была изучена техника работы с пакетом SPARK.