## Содержание текстового файла pnaptext.txt :

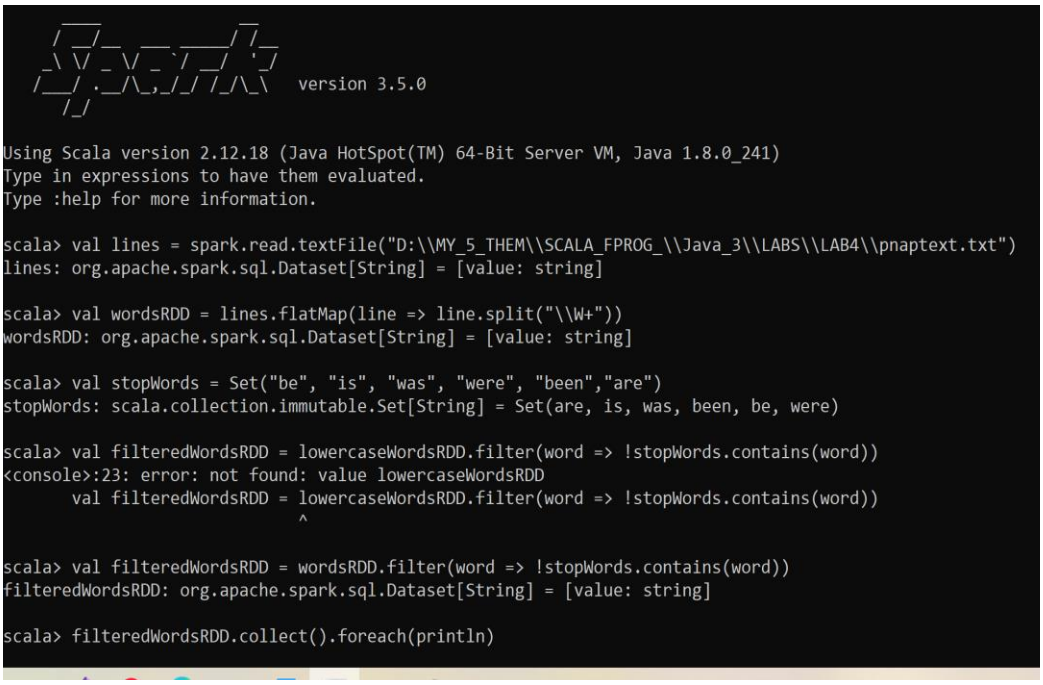
hello to everybody
this is text about sport mostly about football
when you are play in football, be careful
many girls like boys who play football
football will help you communicate with people

## Выполнение работы:

### Вариант 3.

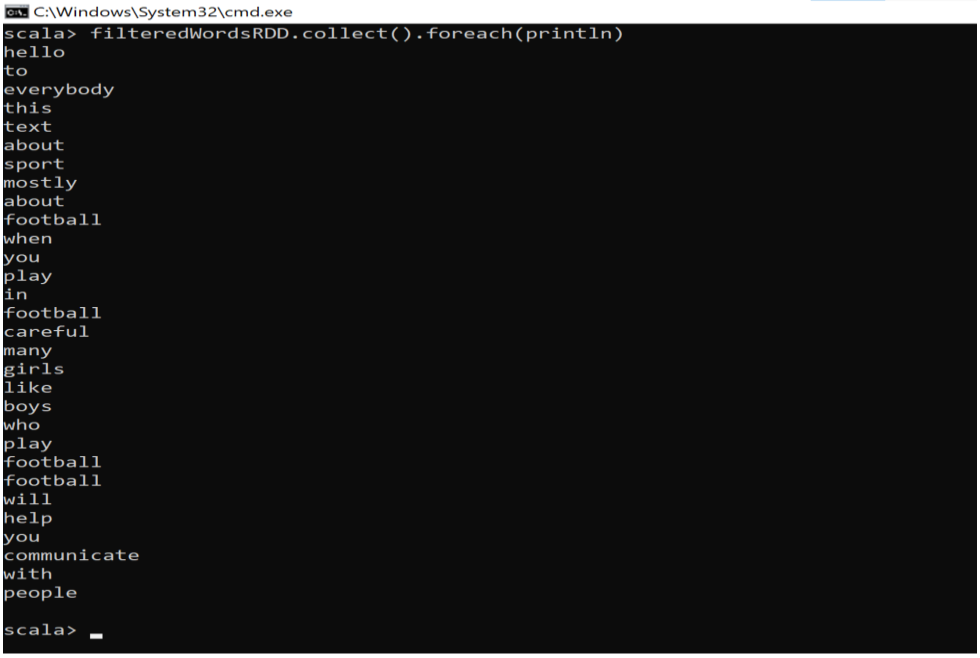
Создать собственный текстовый файл на английском или немецком языке – **4-5** предложений. Тема - спорт

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



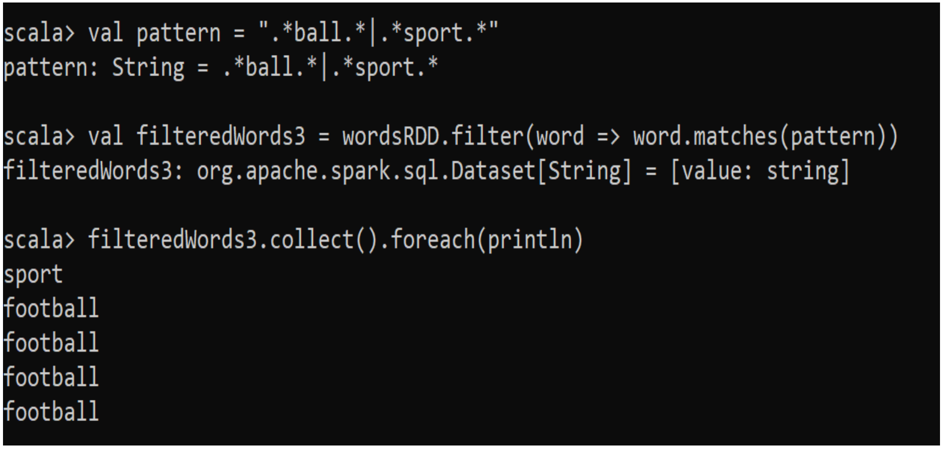
```
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.5.0
      /_/

Using Scala version 2.12.18 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_241)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val lines = spark.read.textFile("D:\\MY_5_THEM\\SCALA_FPROG_\\Java_3\\LABS\\LAB4\\pnaptext.txt")
lines: org.apache.spark.sql.Dataset[String] = [value: string]

scala> val wordsRDD = lines.flatMap(line => line.split("\\W+"))
wordsRDD: org.apache.spark.sql.Dataset[String] = [value: string]

scala> val stopWords = Set("be", "is", "was", "were", "been","are")
stopWords: scala.collection.immutable.Set[String] = Set(are, is, was, been, be, were)

scala> val filteredWordsRDD = lowercaseWordsRDD.filter(word => !stopWords.contains(word))
<console>:23: error: not found: value lowercaseWordsRDD
       val filteredWordsRDD = lowercaseWordsRDD.filter(word => !stopWords.contains(word))
                              ^

scala> val filteredWordsRDD = wordsRDD.filter(word => !stopWords.contains(word))
filteredWordsRDD: org.apache.spark.sql.Dataset[String] = [value: string]

scala> filteredWordsRDD.collect().foreach(println)
```

## Вывод программы:



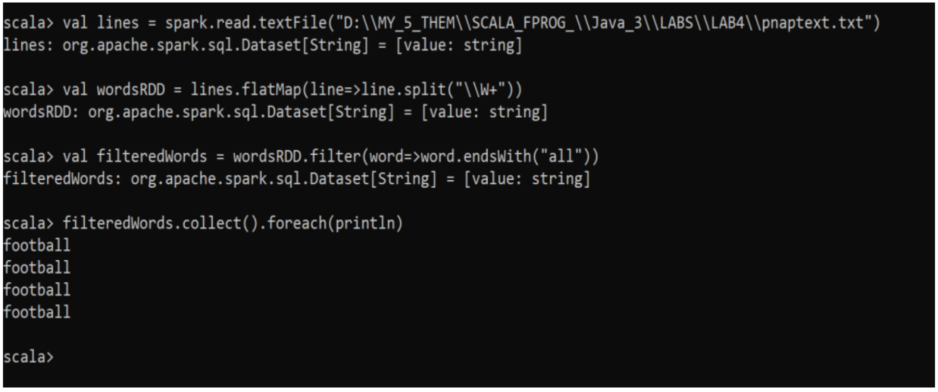
```
scala> filteredWordsRDD.collect().foreach(println)
hello
to
everybody
this
text
about
sport
mostly
about
football
when
you
play
in
football
careful
many
girls
like
boys
who
play
football
football
will
help
you
communicate
with
people

scala> _
```

2. Вывести все слова, содержащие вхождение ball или sport



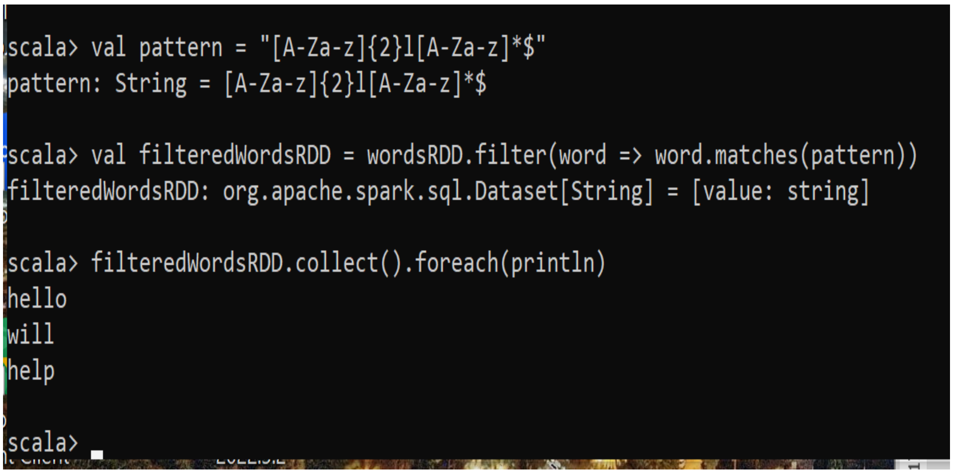
```
scala> val pattern = ".*ball.*|.*sport.*"
pattern: String = .*ball.*|.*sport.*

scala> val filteredWords3 = wordsRDD.filter(word => word.matches(pattern))
filteredWords3: org.apache.spark.sql.Dataset[String] = [value: string]

scala> filteredWords3.collect().foreach(println)
sport
football
football
football
football
```

3. Вывести все слова, заканчивающиеся на ion(all)



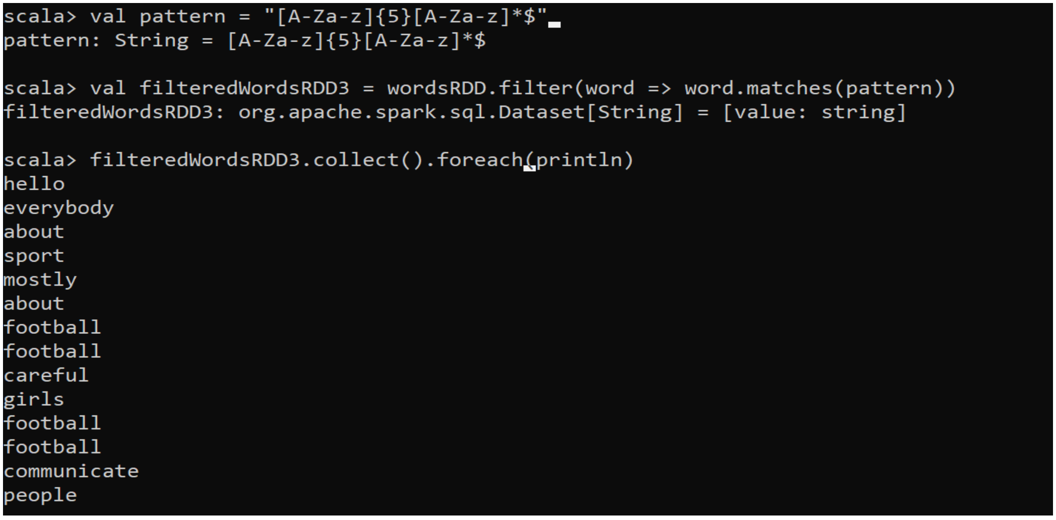
```
scala> val lines = spark.read.textFile("D:\\MY_5_THEM\\SCALA_FPROG_\\Java_3\\LABS\\LAB4\\pnaptext.txt")
lines: org.apache.spark.sql.Dataset[String] = [value: string]

scala> val wordsRDD = lines.flatMap(line=>line.split("\\W+"))
wordsRDD: org.apache.spark.sql.Dataset[String] = [value: string]

scala> val filteredWords = wordsRDD.filter(word=>word.endsWith("all"))
filteredWords: org.apache.spark.sql.Dataset[String] = [value: string]

scala> filteredWords.collect().foreach(println)
football
football
football
football

scala>
```

4. Вывести все слова третья буква которых с (l)



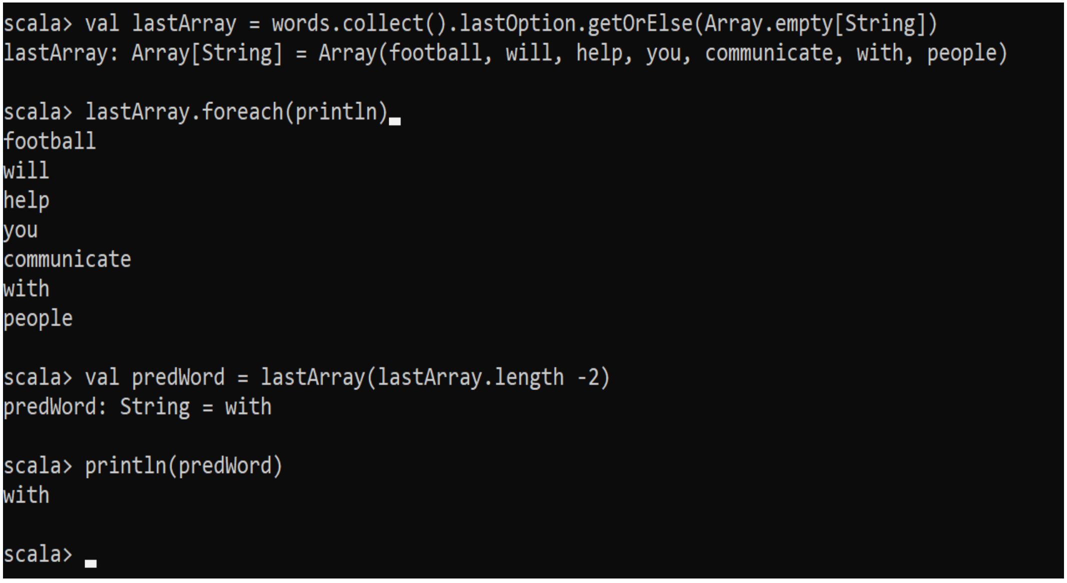
```
scala> val pattern = "[A-Za-z]{2}l[A-Za-z]*$"
pattern: String = [A-Za-z]{2}l[A-Za-z]*$

scala> val filteredWordsRDD = wordsRDD.filter(word => word.matches(pattern))
filteredWordsRDD: org.apache.spark.sql.Dataset[String] = [value: string]

scala> filteredWordsRDD.collect().foreach(println)
hello
will
help

scala>
```

5. Вывести все слова, длина которых больше 4



```
scala> val pattern = "[A-Za-z]{5}[A-Za-z]*$"
pattern: String = [A-Za-z]{5}[A-Za-z]*$

scala> val filteredWordsRDD3 = wordsRDD.filter(word => word.matches(pattern))
filteredWordsRDD3: org.apache.spark.sql.Dataset[String] = [value: string]

scala> filteredWordsRDD3.collect().foreach(println)
hello
everybody
about
sport
mostly
about
football
football
careful
girls
football
football
communicate
people
```

6. Вывести предпоследнее слово

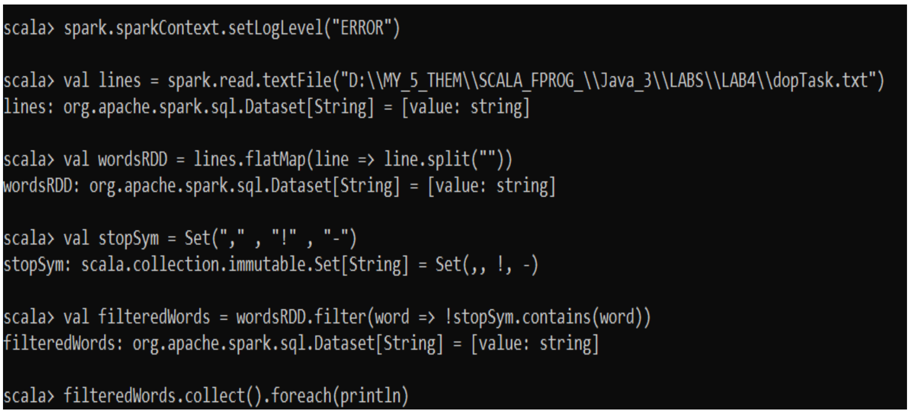


```
scala> val lastArray = words.collect().lastOption.getOrElse(Array.empty[String])
lastArray: Array[String] = Array(football, will, help, you, communicate, with, people)

scala> lastArray.foreach(println)
football
will
help
you
communicate
with
people

scala> val predWord = lastArray(lastArray.length -2)
predWord: String = with

scala> println(predWord)
with

scala>
```

7. Дополнительное задание

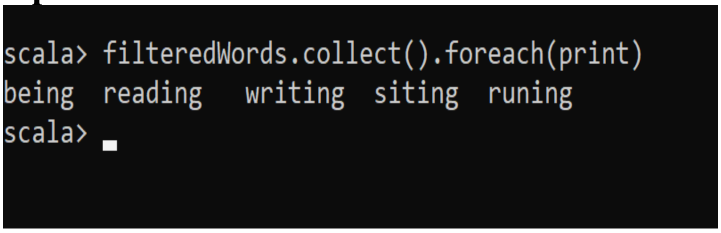
**Содержание текстового файла dopTask.txt:**
being , reading ! , writing -
siting , runing !

1. удалить из текста все стоп-символы.



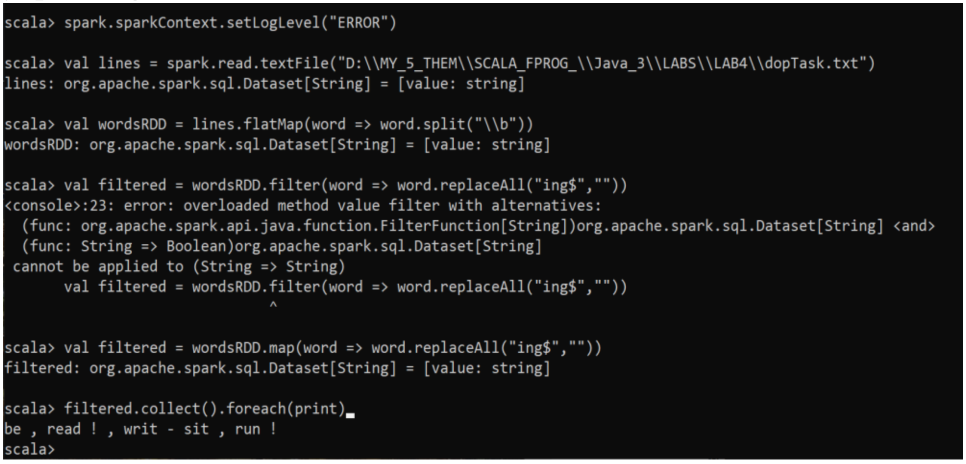
```
scala> spark.sparkContext.setLogLevel("ERROR")

scala> val lines = spark.read.textFile("D:\\MY_5_THEM\\SCALA_FPROG_\\Java_3\\LABS\\LAB4\\dopTask.txt")
lines: org.apache.spark.sql.Dataset[String] = [value: string]

scala> val wordsRDD = lines.flatMap(line => line.split(""))
wordsRDD: org.apache.spark.sql.Dataset[String] = [value: string]

scala> val stopSym = Set("," , "!" , "-")
stopSym: scala.collection.immutable.Set[String] = Set(,, !, -)

scala> val filteredWords = wordsRDD.filter(word => !stopSym.contains(word))
filteredWords: org.apache.spark.sql.Dataset[String] = [value: string]

scala> filteredWords.collect().foreach(println)
```

## Вывод программы:



```
scala> filteredWords.collect().foreach(print)
being  reading   writing  siting  runing
scala>
```

2. убрать ing окончания



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

scala> val lines = spark.read.textFile("D:\\MY_5_THEM\\SCALA_FPROG_\\Java_3\\LABS\\LAB4\\dopTask.txt")
lines: org.apache.spark.sql.Dataset[String] = [value: string]

scala> val wordsRDD = lines.flatMap(word => word.split("\\b"))
wordsRDD: org.apache.spark.sql.Dataset[String] = [value: string]

scala> val filtered = wordsRDD.filter(word => word.replaceAll("ing$",""))
<console>:23: error: overloaded method value filter with alternatives:
  (func: org.apache.spark.api.java.function.FilterFunction[String])org.apache.spark.sql.Dataset[String] <and>
  (func: String => Boolean)org.apache.spark.sql.Dataset[String]
 cannot be applied to (String => String)
       val filtered = wordsRDD.filter(word => word.replaceAll("ing$",""))
                               ^

scala> val filtered = wordsRDD.map(word => word.replaceAll("ing$",""))
filtered: org.apache.spark.sql.Dataset[String] = [value: string]

scala> filtered.collect().foreach(print)
be , read ! , writ - sit , run !
scala>
```