Федеральное государственное автономное

образовательное учреждение

Высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт космических и информационных технологий |
| институт |
| Программная инженерия |
| кафедра |

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ**

|  |
| --- |
| Отображения, трейты и функции высшего порядка |
| тема |

Вариант 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Преподаватель | |  |  |  | К. В. Богданов |
|  | |  | подпись, дата |  | инициалы, фамилия |
| Студент | КИ21-17/1Б, 032156940 |  |  |  | Н. А. Самарин |
|  | номер группы, зачётной книжки |  | подпись, дата |  | инициалы, фамилия |

Красноярск 2024

**СОДЕРЖАНИЕ**

1 Задание............................................................................................................... 3

2 Исходный код разработанного алгоритма..................................................... 3

3 Результат........................................................................................................... 3

**1 Задание**

Написать скрипт, принимающий в качестве первого параметра путь к  
текстовому файлу (plain text). Остальные параметры указаны в вариантах  
заданий. В соответствии с вариантом задания скрипт должен вывести в  
стандартный поток результаты обработки, а в случае возникновения ошибки  
вывести соответствующее сообщение в поток ошибок (таким образом, скрипт  
можно будет использовать в unix/linux pipeline). Конкретная реализация  
механизма чтения из файла остаётся на ваше усмотрение, однако, будьте  
готовы ответить на вопрос: “как нужно модифицировать скрипт, если размер  
файла значительно - до гигабайтов - возрастёт?” Словом в тексте считаем  
любую последовательность непробельных (отображаемых) символов, которая  
ограничена пробельными символами либо началом/концом строки.

Вариант 4 (24): Вернуть среднюю длину слова для каждой строки текста.

**2 Исходный код разработанного алгоритма**

Листинг 1 – Исходный код разработанного алгоритма

object Main {  
 def main(args: Array[String]): Unit = {  
 if (args.length < 1) {  
 throw new Exception("Please provide the path to the text file as a paramet  
 er.")  
 }  
 val filePath = args(0)  
 val source = scala.io.Source.fromFile(filePath)  
 val lines = try source.getLines().toList finally source.close()  
 val averageWordLengths = lines.map(calculateAverageWordLength)  
 averageWordLengths.foreach(println)  
 }  
 /\*\*  
 \* Вычисление средней длины слов в строке  
 \* @param line строка  
 \* @return средняя длина слов в строке  
 \*/  
 def calculateAverageWordLength(line: String): Double = {  
 val words = line.split("\\s+")  
 val totalLength = words.map(\_.length).sum  
 val averageLength = if (words.nonEmpty) totalLength / words.length.toDouble   
 else 0  
 averageLength  
 }  
}

**3 Результат**

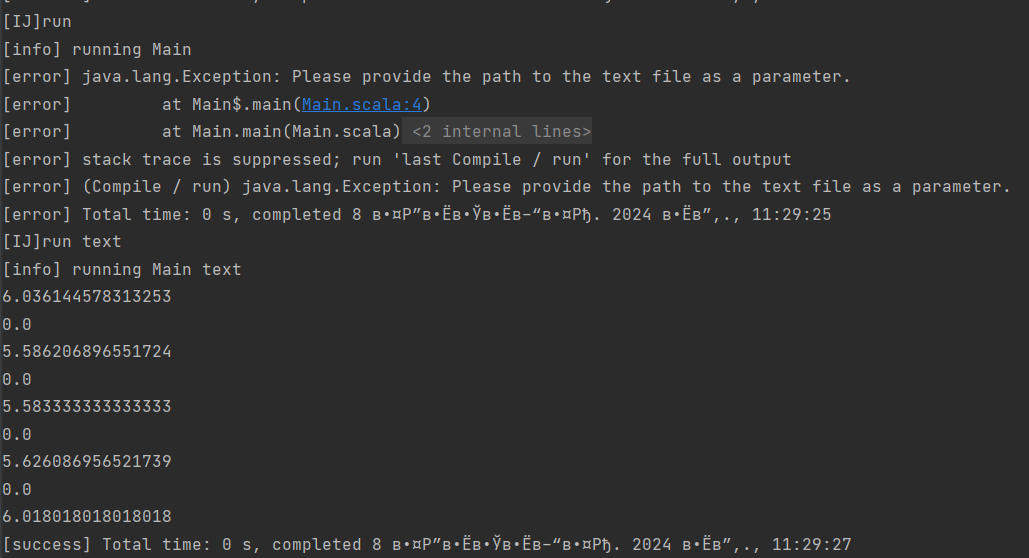


Рисунок 1 – Результаты работы алгоритма