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 СО СТРОКАМИ

|  |  |
| --- | --- |
| Студенты группы 150503:  Проверила: | Ходосевич М.А.  Семков А. Д.  Герман Ю. О. |

Минск 2023

Цель: изучить технику работы со строками в Scala.

1. **Краткие теоретические сведения:**

Scala предоставляет богатый набор функций для работы со строками. Вот некоторые расширенные функции, которые можно использовать со строками в Scala:

1. replaceAll: эта функция используется для замены всех вхождений строки другой строкой. Функция принимает два аргумента: первый аргумент — это заменяемое регулярное выражение, а второй аргумент — строка замены.
2. split: эта функция используется для разделения строки на массив подстрок на основе разделителя. Функция принимает один аргумент — строку-разделитель.
3. startsWith и endsWith: Эти функции используются для проверки того, начинается или заканчивается строка заданной подстрокой. Функции принимают один аргумент — проверяемую подстроку.
4. substring: выделяет подстроку из строки. Пример:
5. toCharArray: преобразует строку в массив символов
6. toLowerCase и toUpperCase: Преобразует символы строки в верхний и нижний регистр соответственно.
7. trim: отсекает концевые пробелы.
8. indexOf иlastIndexOf: получает первый и последний индекс подстроки в строке (то есть номер позиции, с которой начинается подстрока).
9. charAt: определяет символ, стоящий на указанной позиции
10. stripMargin: удаляет ведущие пробелы перед строкой

Примеры из методички:

Пример 1:

val str = "Hello, World!"

val newStr = str.replaceAll("o", "a")

println(newStr)

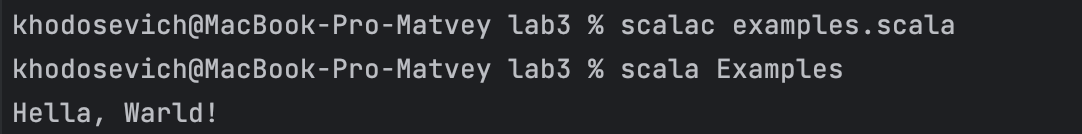


Рисунок 1 – результат примера 1

Пример 2:

val str = "apple,banana,orange"

val arr = str.split(",")

println(arr.mkString(" "))

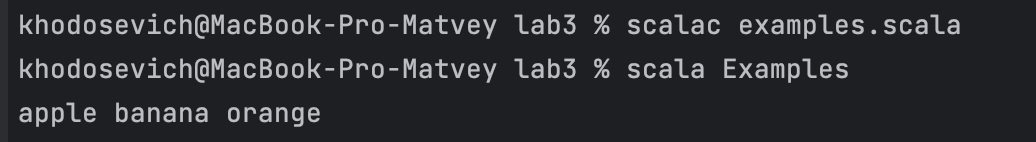


Рисунок 2 – результат примера 2

Пример 3:

val str = "Hello, World!"

println(str.startsWith("Hello"))

println(str.endsWith("!"))

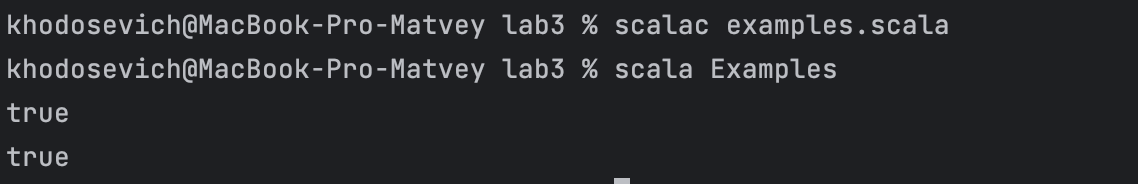


Рисунок 3 – результат примера 3

Пример 4:

val str = "Hello, World!"

val subStr = str.substring(7, 12)

println(subStr)

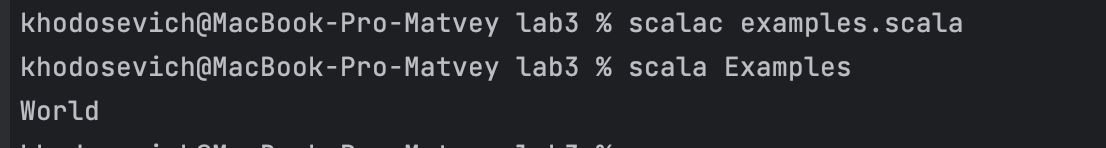


Рисунок 4 – результат примера 4

Пример 5:

val str = "Hello, World!"

val arr = str.toCharArray()

println(arr.mkString(" "))

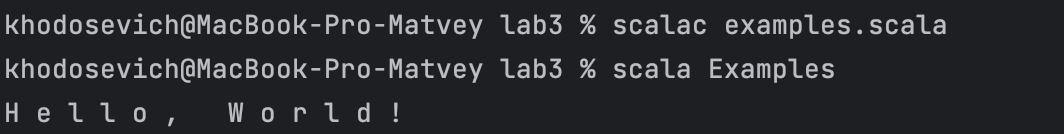


Рисунок 5 – результат примера 5

Пример 6:

val str = "Hello, World!"

println(str.toLowerCase) // "hello, world!"

println(str.toUpperCase) // "HELLO, WORLD!"

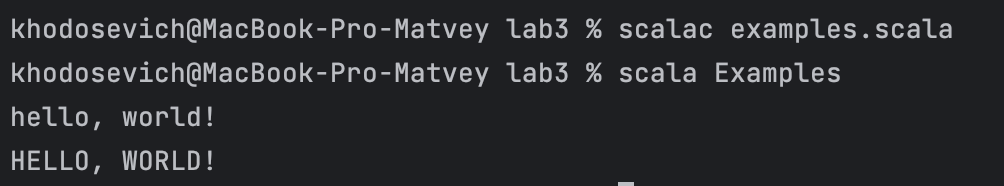


Рисунок 6 – результат примера 6

Пример 7:

val str = " Hello, World! "

println(str.trim)

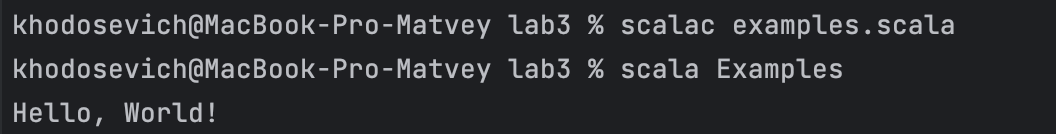


Рисунок 7 – результат примера 7

Пример 8:

val str = "Hello, World!"

println(str.indexOf("o"))

println(str.lastIndexOf("o"))

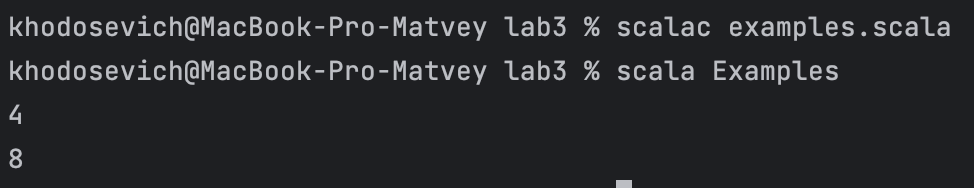


Рисунок 8 – результат примера 8

Пример 9:

val str = "Hello, World!"

println(str.charAt(7))

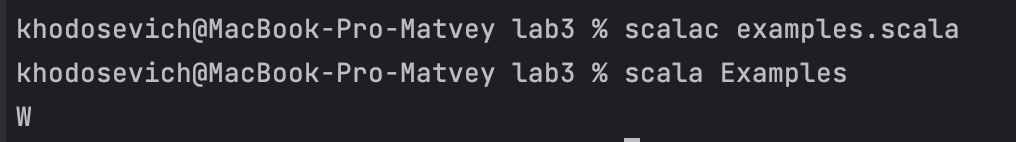


Рисунок 9 – результат примера 9

Пример 10:

val str =

"""

|Hello,

|World!

|""".stripMargin

println(str)

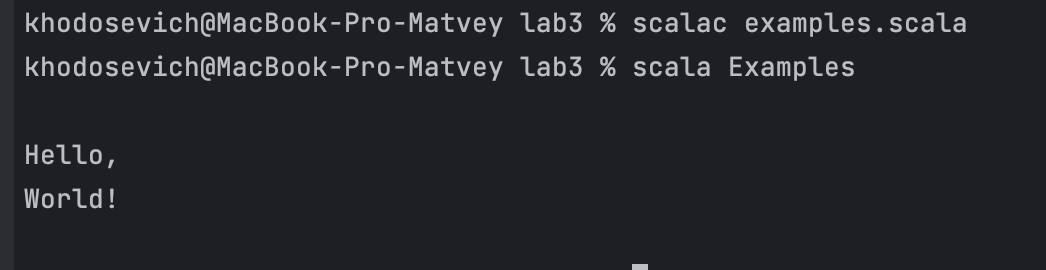


Рисунок 10 – результат примера 10

Пример 11:

val regex = """(\d{3})-(\d{2})-(\d{4})""".r

val str1 = "123-45-6789"

val str2 = "abc-12-3456"

val match1 = regex.findFirstMatchIn(str1)

val match2 = regex.findFirstMatchIn(str2)

match1 match {

case Some(m) => println(s"Match found: ${m.group(0)}")

case None => println("No match found")

}

match2 match {

case Some(m) => println(s"Match found: ${m.group(0)}")

case None => println("No match found")

}

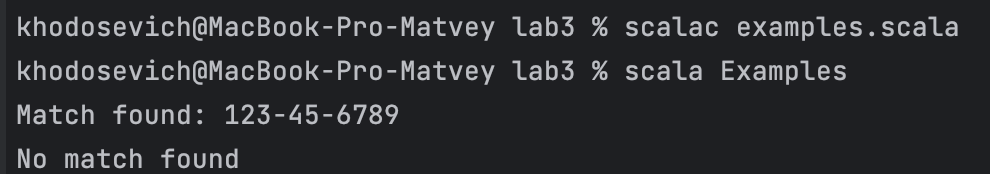


Рисунок 11 – результат примера 11

Пример 12:

val str = "The quick brown fox jumps over the lazy dog"

val pattern = "fox".r

val matches = pattern.findAllIn(str)

matches.foreach(println)

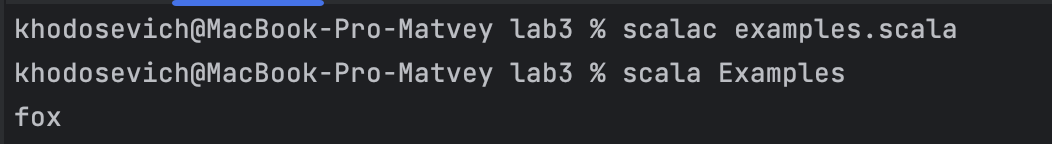


Рисунок 12 – результат примера 12

**2. Ход работы:**

**Вариант 4.**

1. Дан текст: ‘Passport AB-123-436’. C помощью техники регулярных выражений найти число вхождений цифры 3.

object Main1 {

def main(args: Array[String]): Unit = {

val text = "Passport AB-123-436"

val pattern = "3"

val occurrences = pattern.r.findAllIn(text).length

println(s"Число вхождений цифры 3: $occurrences") }}

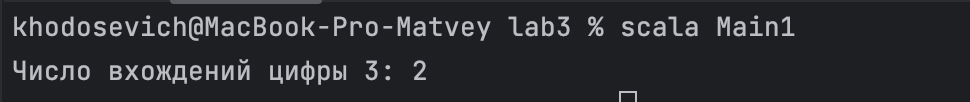


Рисунок 13 – результат выполнения первого задания

1. Дан текст: ‘Passport AB-123-436’. C помощью техники регулярных выражений найти сумму всех цифр в тексте.

object Main2 {

def main(args: Array[String]): Unit = {

val text = "Passport AB-123-436"

val digits = text.replaceAll("\\D", "").map(\_.asDigit)

val sum = digits.sum

println(s"Сумма всех цифр: $sum")}}

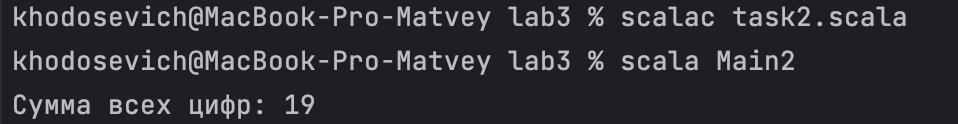


Рисунок 14 – результат выполнения второго задания

1. В тексте ‘123+723=846’ выделить все символы арифметических операций.

object Main3 {

def main(args: Array[String]): Unit = {

val text = "123+723=846"

val operations = text.filter(c => c == '+' || c == '-' || c == '\*' || c == '/' || c == '=' )

println(s"Символы арифметических операций: $operations")}}

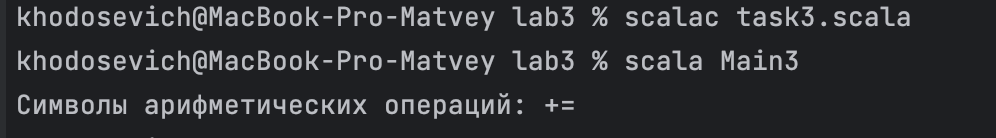


Рисунок 15 – результат выполнения третьего задания

1. Записать все слова в тексте в случайном порядке. Текст: big black cat runs after small poor catty. Всех слов 8. Номера 1,2,3, …,8. Заменить слово с номером 1 на слово с номером 4, слово с номером 2 на слово с номером 7, слово с номером 3 на слово с номером 6. Замена обоюдная, то есть слова просто меняются местами.

object Main4 {

def main(args: Array[String]): Unit = {

val text = "big black cat runs after small poor catty."

val words = text.split(" ").toList

val shuffledWords = util.Random.shuffle(words)

val replacedWords = shuffledWords

.updated(0, shuffledWords(3))

.updated(1, shuffledWords(6))

.updated(2, shuffledWords(5))

val resultString = replacedWords.mkString(" ")

println(resultString) }}

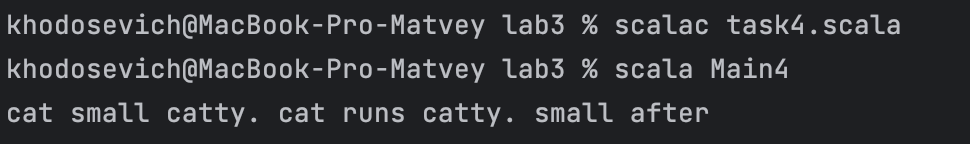


Рисунок 16 – результат выполнения четвертого задания

1. Дан текст: ‘Hello to Ronny old nail’. Найти удвоенные вхождения согласных.

object Main5 {

def findDoubledConsonants(text: String): List[String] = {

val consonants = "bcdfghjklmnpqrstvwxyz"

val matches = for {

i <- 0 until text.length - 1

if consonants.contains(text(i).toLower) && consonants.contains(text(i + 1).toLower) && text(i) == text(i + 1)

} yield text.substring(i, i + 2)

matches.toList

}

def main(args: Array[String]): Unit = {

val text = "Hello to Ronny old nail"

val doubledConsonants = findDoubledConsonants(text)

doubledConsonants.foreach(println)}}

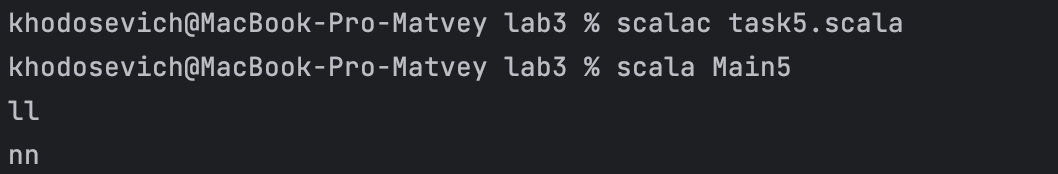


Рисунок 17 – результат выполнения пятого задания

1. Дан текст: ‘Hello to Ronny old nail’. Найти число слов, записанных с большой буквы.

object Main6 {

def countWordsStartingWithUpperCase(text: String): Int = {

val words = text.split("\\s+")

words.count(word => word.nonEmpty && word(0).isUpper)

}

def main(args: Array[String]): Unit = {

val text = "Hello to Ronny old nail"

val result = countWordsStartingWithUpperCase(text)

println(s"Число слов, начинающихся с большой буквы: $result")}}

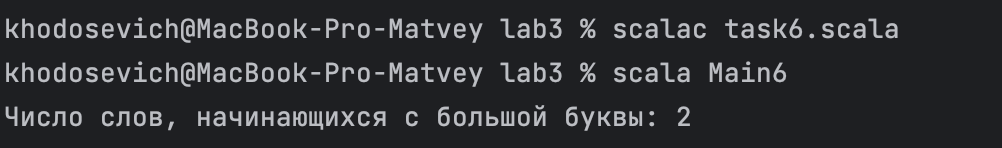


Рисунок 18 – результат выполнения шестого задания

1. Дан текст: ‘Hello to everybody’. Подсчитать число всех вхождений буквы o

object Main6 {

def countWordsStartingWithUpperCase(text: String): Int = {

val words = text.split("\\s+")

words.count(word => word.nonEmpty && word(0).isUpper)

}

def main(args: Array[String]): Unit = {

val text = "Hello to Ronny old nail"

val result = countWordsStartingWithUpperCase(text)

println(s"Число слов, начинающихся с большой буквы: $result")}}

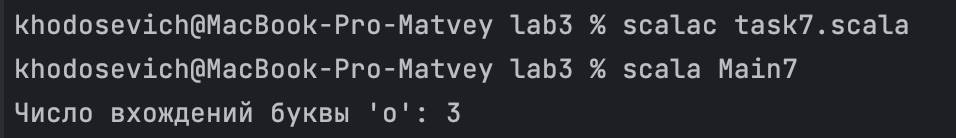


Рисунок 19 – результат выполнения седьмого задания

1. Дан текст: ‘Hello agaiN to everybody Katty, Mikky’. Вывести все слова, начинающиеся со строчной буквы.

object Main8 {

def findLowercaseWords(text: String): List[String] = {

val words = text.split("\\W+").toList

words.filter(\_.nonEmpty).filter(\_.head.isLower)

}

def main(args: Array[String]): Unit = {

val text = "Hello agaiN to everybody Katty, Mikky"

val lowercaseWords = findLowercaseWords(text)

lowercaseWords.foreach(println)}}

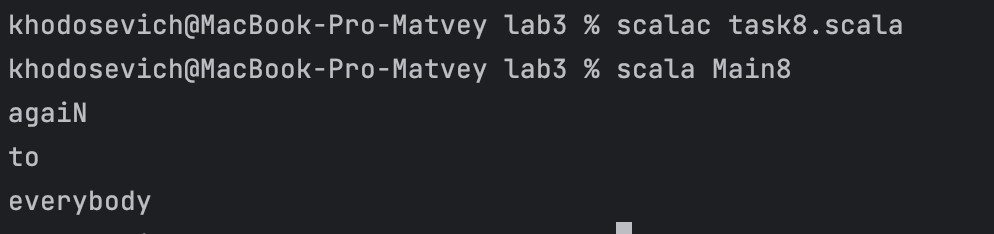


Рисунок 20 – результат выполнения восьмого задания

Вывести суммарное число всех гласных в собственном тексте.

object Main8 {

def countVowels(text: String): Int = {

val vowels = "aeiouAEIOU"

text.count(vowels.contains)

}

def main(args: Array[String]): Unit = {

val text = "Hello agaiN to everybody Katty, Mikky"

val vowelsCount = countVowels(text)

println(s"Суммарное число всех гласных: $vowelsCount")}

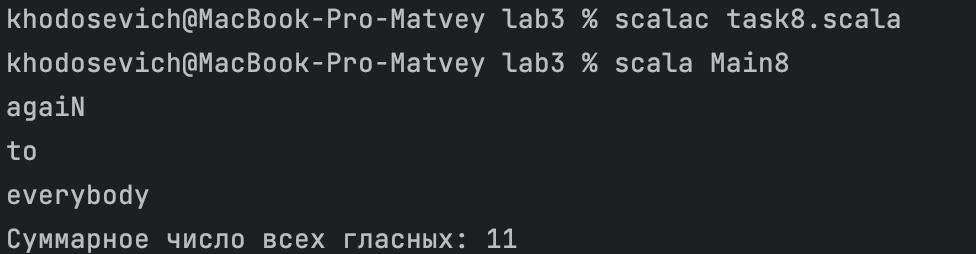


Рисунок 21 – результат выполнения дополнительного задания

1. **Вывод:**

В ходе выполнения данной лабораторной работы нами были проанализированы и изучены техники работы со списками в Scala.