БГУИР

# Кафедра ЭВМ

### Отчет по лабораторной работе № 3

**Тема: «Работа со строками»**

Выполнили:

студенты группы 130501 Блинников В.А.

Гнездилов А.М.

Проверила:

**к.т.н., доцент Герман Ю.О.**

#### Минск 2023

1. **ЦЕЛЬ РАБОТЫ**

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: 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.

split: Эта функция используется для разделения строки на массив подстрок на основе разделителя. Функция принимает один аргумент — строку-разделитель.

StartsWith и endsWith: Эти функции используются для проверки того, начинается или заканчивается строка заданной подстрокой. Функции принимают один аргумент — проверяемую подстроку.

substring: Выделяет подстроку из строки. Пример:

toCharArray: Преобразует строку в массив символов

Пример:

val str = "Hello, World!"

val arr = str.toCharArray()

println(arr.mkString(" ")) // "H e l l o , W o r l d !"

toLowerCase и toUpperCase: Преобразует символы строки в верхний и нижний регистр соответственно.

Пример:

val str = "Hello, World!"

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

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

trim: Отсекает концевые пробелы.

Пример:

val str = " Hello, World! "

println(str.trim) // "Hello, World!"

indexOf иlastIndexOf: Получает первый и последний индекс подстроки в строке (то есть номер позиции, с которой начинается подстрока).

charAt: Определяет символ, стоящий на указанной позиции.

В Scala регулярные выражения представлены классом scala.util.matching.Regex, который предоставляет множество методов для сопоставления строк и управления ими на основе регулярных выражений.

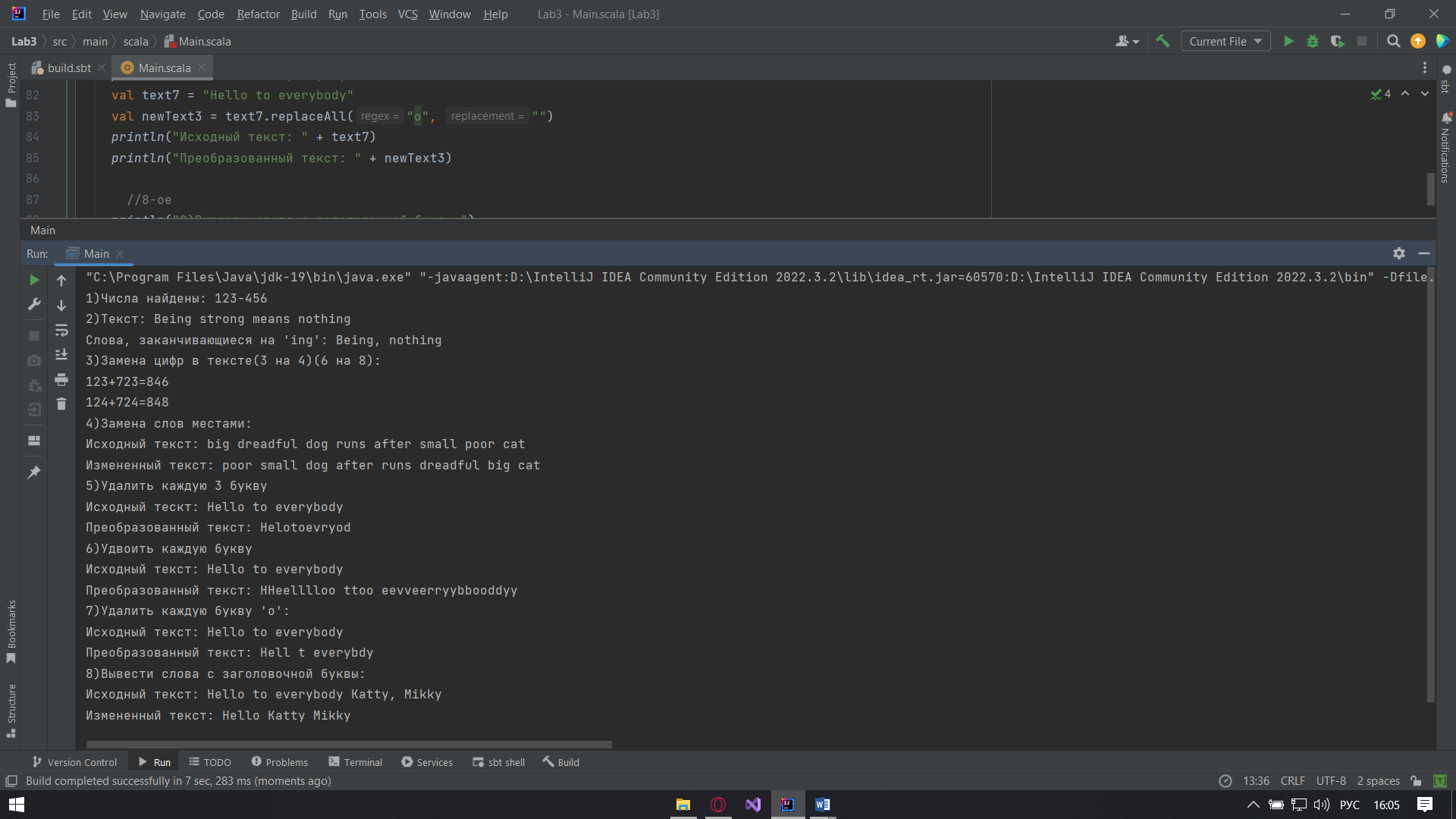
1. **ХОД РАБОТЫ**

**Вариант 3.**

1. Дан текст: ‘Passport AB-123-456’. C помощью техники регулярных выражений найти числа в этом тексте
2. Найти в тексте ‘Being strong means nothing’ все слова, заканчивающиеся на ing.
3. В тексте ‘123+723=846’ заменить 3 на 4, 6 – на 8.
4. Записать все слова в тексте в случайном порядке. Текст: big dreadful dog runs after small poor cat. Всех слов 8. Номера 1,2,3, …,8. Заменить слово с номером 1 на слово с номером 7, слово с номером 2 на слово с номером 6, слово с номером 4 на слово с номером 5. Замена обоюдная, то есть слова просто меняются местами.
5. Дан текст: ‘Hello to everybody’. Выбросить каждую третью букву.
6. Дан текст: ‘Hello to everybody’. Удвоить каждую букву в слове
7. Дан текст: ‘Hello to everybody’. Удалить все вхождения буквы o
8. Дан текст: ‘Hello to everybody Katty, Mikky’. Вывести все слова, начинающиеся с заглавной буквы.

**Листинг кода**

object Main {  
 def main(args: Array[String]): Unit = {  
 //1-ое  
 val regex = """\d+""".r // Регулярное выражение для поиска последовательности цифр  
 val str = "Passport AB-123-456"  
 val matches = regex.findAllIn(str).toList // Найти все совпадения  
  
 if (matches.nonEmpty) {  
 val result = matches.mkString("-")  
 print(s"1)Числа найдены: $result ")  
 print("\n")  
 } else {  
 println("1)Числа не найдены")  
 }  
  
 //2-ое  
 val text = "Being strong means nothing"  
 val pattern = "\\b\\w+ing\\b".r // Регулярное выражение для поиска слов, заканчивающихся на "ing"  
 val matches2 = pattern.findAllIn(text)  
 println("2)Текст: " + text)  
 if (matches2.nonEmpty) {  
 val result2 = matches2.mkString(", ") //преобразуем в строку  
 println(s"Слова, заканчивающиеся на 'ing': $result2")  
 matches2.foreach(println)  
 } else {  
 println("Слова, заканчивающиеся на 'ing', не найдены")  
 }  
  
 //3-ee  
 val str3 = "123+723=846"  
 val newstr3 = str3.replaceAll("3", "4")  
 val newstr3final = newstr3.replaceAll("6", "8")  
 println("3)Замена цифр в тексте(3 на 4)(6 на 8): \n" + str3)  
 println(newstr3final)  
  
//4-ое  
 println("4)Замена слов местами:")  
 val text1 = "big dreadful dog runs after small poor cat"  
 val words = text1.split(" ") // Разбиваем текст на слова  
  
 // Проверяем, что у нас достаточно слов для замены  
 if (words.length >= 8) {  
 val word1 = words(0)  
 val word2 = words(1)  
 val word4 = words(3)  
  
 // Выполняем замены  
 words(0) = words(6)  
 words(6) = word1  
 words(1) = words(5)  
 words(5) = word2  
 words(3) = words(4)  
 words(4) = word4  
  
 // Объединяем слова в новом порядке  
 val result3 = words.mkString(" ")  
 println(s"Исходный текст: $text1")  
 println(s"Измененный текст: $result3")  
 } else {  
 println("Недостаточно слов для замены")  
 }  
  
 //5-ое  
 println("5)Удалить каждую 3 букву")  
 val text5 = "Hello to everybody"  
 val result = text5.zipWithIndex.collect { case (char, index) if (index + 1) % 3 != 0 => char }  
 val newText5 = result.mkString  
 println("Исходный тескт: " + text5)  
 println("Преобразованный текст: "+newText5)  
  
 //6-ое  
 println("6)Удвоить каждую букву")  
 val text6 = "Hello to everybody"  
 val wordss = text6.split(" ")  
 val updatedWords = wordss.map(word => word.flatMap(char => char.toString \* 2))  
 val newText2 = updatedWords.mkString(" ")  
 println("Исходный текст: " + text6)  
 println("Преобразованный текст: "+newText2)  
  
 //7-ое  
 println("7)Удалить каждую букву 'о':")  
 val text7 = "Hello to everybody"  
 val newText3 = text7.replaceAll("o", "")  
 println("Исходный текст: " + text7)  
 println("Преобразованный текст: " + newText3)  
  
 //8-ое  
 println("8)Вывести слова с заголовочной буквы:")  
 val text8 = "Hello to everybody Katty, Mikky"  
 val words8 = text8.split("\\s+|,\\s\*")  
 val newWords8 = words8.filter(word => word.headOption.exists(\_.isUpper))  
 val result8 = newWords8.mkString(" ")  
 println(s"Исходный текст: $text8")  
 println(s"Измененный текст: $result8")  
 }  
}



1. **ВЫВОД**

Изучили технику работы со строками в Scala.