**vНЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **“СИНЕРГИЯ”»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | Программирование |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | Информационные системы и программирование |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | Очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Отчет по практической работе №4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | Массивы и строки | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | Разработка программных модулем |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Ходоров Дмитрий Михайлович |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | VДКИП-111прог |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Сибирев И. В. |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2025 г.**

# Аннотация

По согласованию во время вебинара работы выполнена на языке Kotlin т.к. он используется в работе и лучше мне знаком

# Задание №1(лабораторная работа 5, вариант 5)

Исходный код программы приведен в Листинге 1, скриншот работы программы на Рисунке 1

//Вариант 5, лабораторная работа 5

fun findMaxElement(array: DoubleArray): Double {

return array.maxOrNull() ?: 0.0

}

fun sumBeforeLastPositive(array: DoubleArray): Double {

val lastPositiveIndex = array.indexOfLast { it > 0 }

return if (lastPositiveIndex == -1) 0.0

else array.take(lastPositiveIndex).sum()

}

fun compressArray(array: DoubleArray, a: Double, b: Double): DoubleArray {

val filtered = array.filter { Math.abs(it) !in a..b }

return DoubleArray(array.size) { index ->

filtered.getOrNull(index) ?: 0.0

}

}

val array = doubleArrayOf(1.0, -2.0, 3.0, -4.0, 5.0, -6.0)

val a = 1.0

val b = 3.0

println("Максимальный элемент: ${findMaxElement(array)}") // 5

println("Сумма элементов до последнего положительного: ${sumBeforeLastPositive(array)}") // -2

println("Сжатый массив: ${compressArray(array, a, b).joinToString()}") // -4 5 -6 0 0 0

Листинг 1 - Исходный код программы

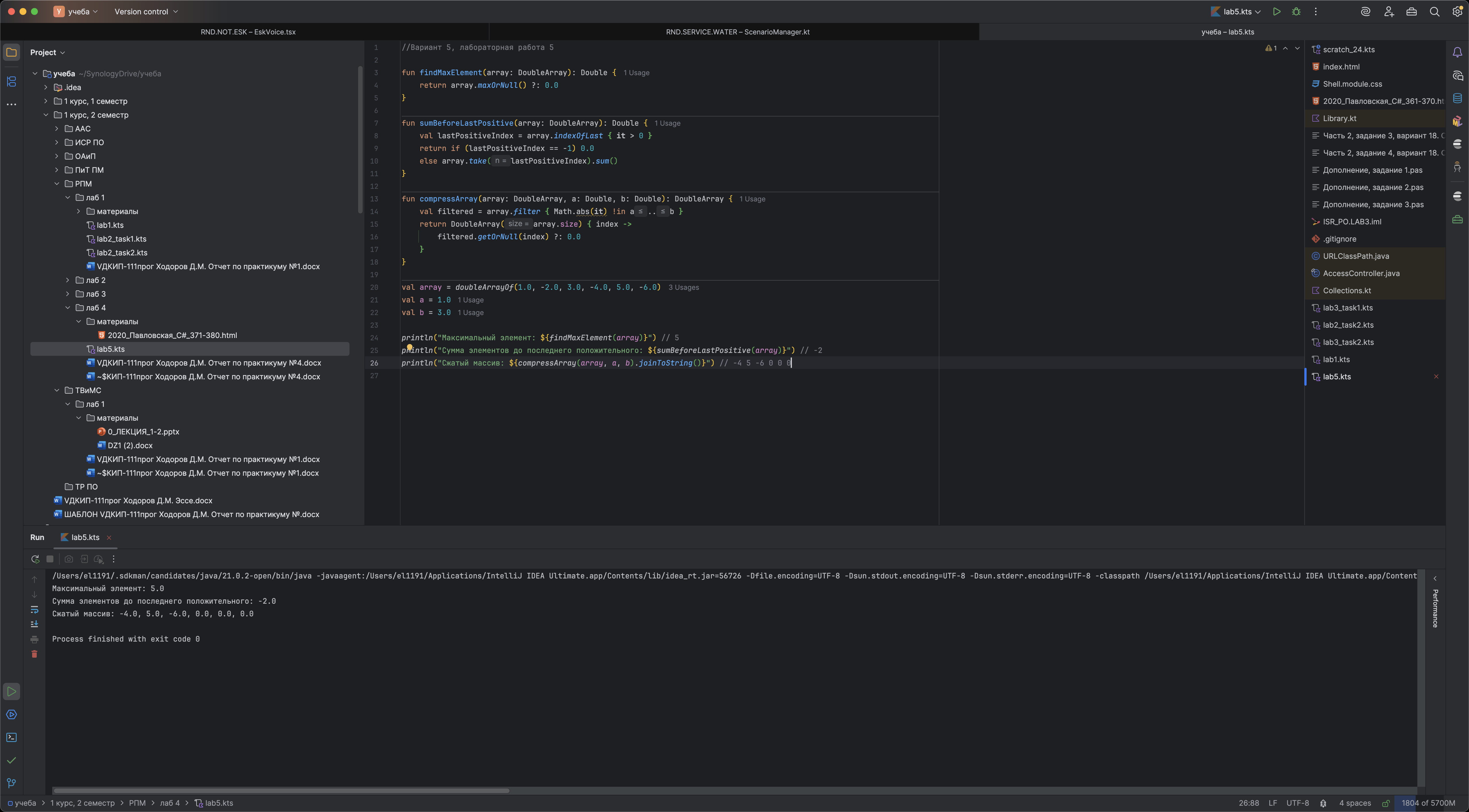


Рисунок 1 - Результат работы программы

# Задание №2(лабораторная работа 6, вариант 5)

Исходный код программы приведен в Листинг 2, скриншот работы программы на Рисунке 2

//Вариант 5, лабораторная работа 6

fun sumOfColumnsWithoutNegatives(matrix: Array<IntArray>): Int {

var totalSum = 0

for (col in matrix.indices) {

var hasNegative = false

for (row in matrix.indices) {

if (matrix[row][col] < 0) {

hasNegative = true

break

}

}

if (!hasNegative) {

for (row in matrix.indices) {

totalSum += matrix[row][col]

}

}

}

return totalSum

}

fun minSumOfParallelDiagonals(matrix: Array<IntArray>): Int {

val n = matrix.size

val sums = mutableListOf<Int>()

for (s in 0 until 2 \* n - 1) {

var currentSum = 0

for (i in 0 until n) {

val j = s - i

if (j in 0 until n) {

currentSum += kotlin.math.abs(matrix[i][j])

}

}

sums.add(currentSum)

}

return sums.minOrNull() ?: 0

}

val matrix = arrayOf(

intArrayOf(-1, 2, 3),

intArrayOf(4, 5, 6),

intArrayOf(7, 8, 9)

)

println("Сумма столбцов без отрицательных элементов: ${sumOfColumnsWithoutNegatives(matrix)}")

println("Минимум среди сумм модулей диагоналей: ${minSumOfParallelDiagonals(matrix)}")

Листинг 2 - Исходный код программы

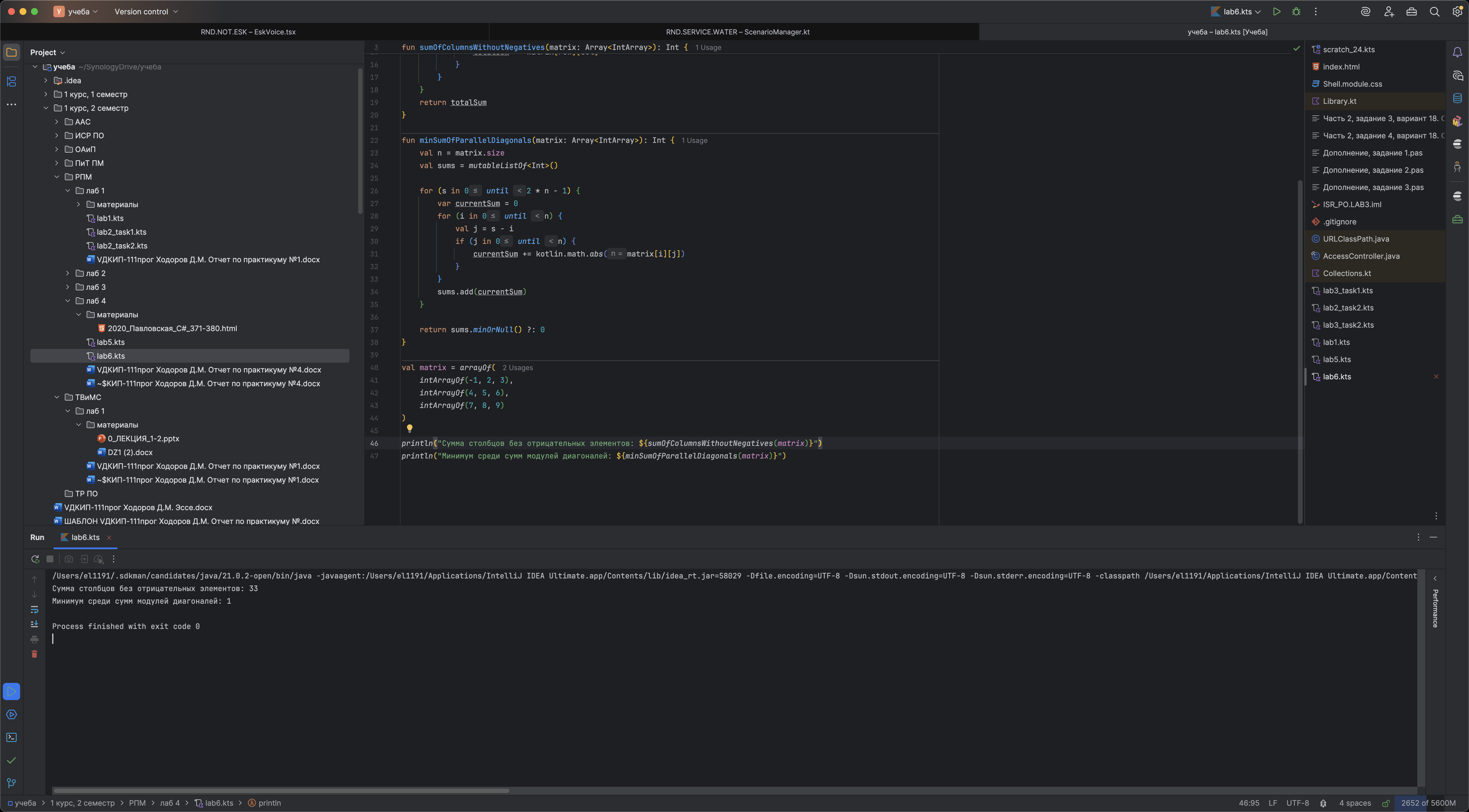


Рисунок 2 - Результат работы программы

# Задание №3(лабораторная работа 7, вариант 5)

Исходный код программы приведен в Листинге 3, скриншот работы программы на Рисунке 3. В качестве файла для входа использовался текстовый файл с содержимым «Hello world from text file», получен вывод «world Hello text from file»

import java.io.File

//Вариант 5, лабораторная работа 7

fun processText(text: String): String {

val words = text.split("\\s+".toRegex()).filter { it.isNotBlank() }

val swappedWords = swapAdjacentWords(words)

return swappedWords.joinToString(" ")

}

fun swapAdjacentWords(words: List<String>): List<String> {

val result = mutableListOf<String>()

var i = 0

while (i < words.size) {

if (i + 1 < words.size) {

result.add(words[i + 1])

result.add(words[i])

i += 2

} else {

result.add(words[i])

i++

}

}

return result

}

val text = File("lab7\_input.txt").readText()

val result = processText(text)

println(result)

Листинг 3 - Исходный код программы

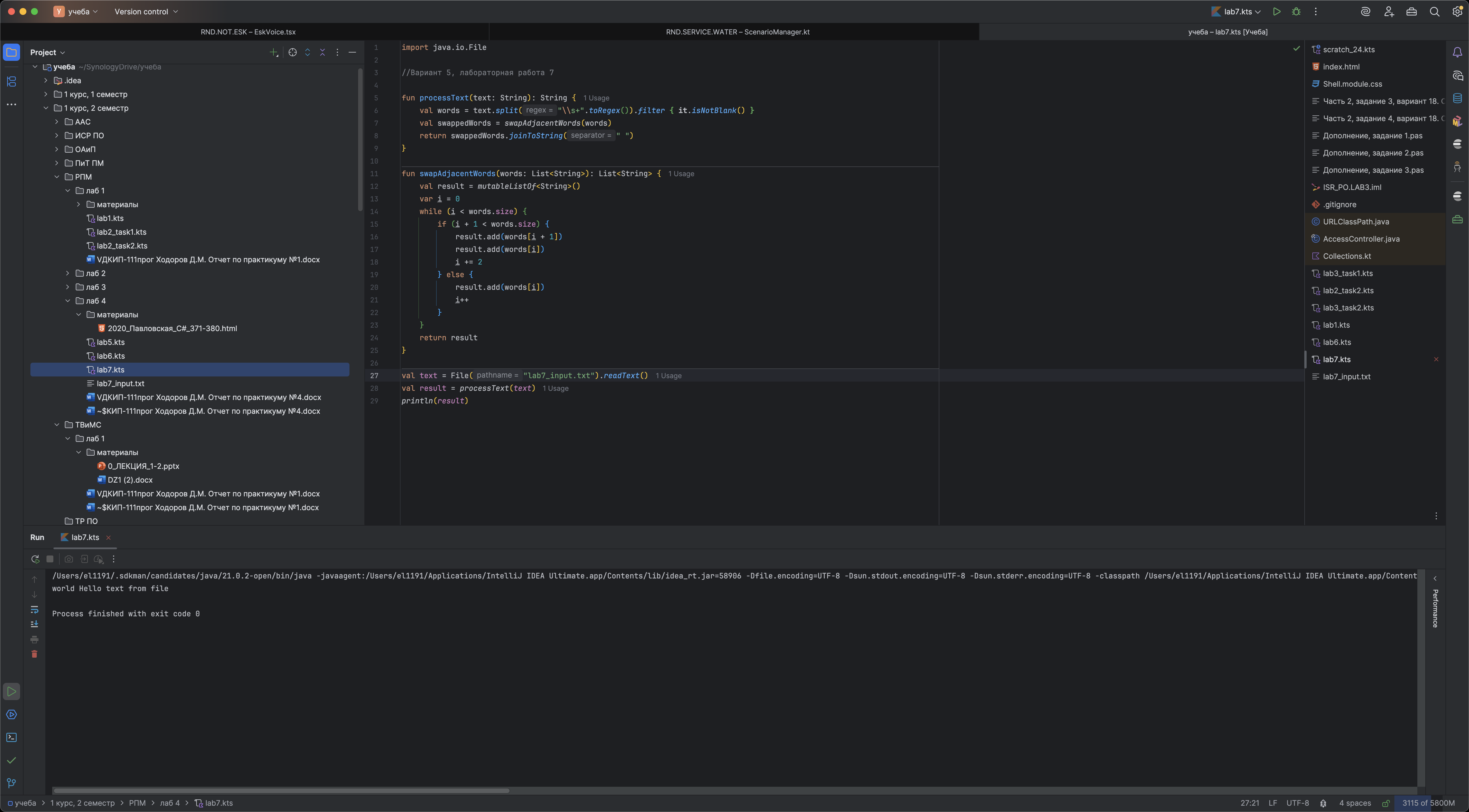


Рисунок 3 - Результат работы программы