

Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Дисциплина: Программирование мобильных информационных систем

Отчет  
по лабораторной работе №3  
на тему:  
**«Функциональное программирование и лямбда-выражения»**

Выполнил:  
ст. гр. 214302  
Короткая Д.О.

Проверил:  
Усенко Ф.В

Минск 2024

**Задание:** Лямбда-выражения для работы с многомерными данными: Напишите программу, которая использует лямбда-выражения для анализа и обработки многомерных данных, таких как матрицы или тензоры. Программа должна поддерживать сложные операции, такие как транспонирование, слияние и нормализация.

**Листинг кода:**

```
fun inputMatrix(name: String): Array<IntArray> {
    print("Введите количество строк $name матрицы:")
    val r = readLine()!!.toInt()
    print("Введите количество столбцов $name матрицы:")
    val c = readLine()!!.toInt()

    val matrix = Array(r) { IntArray(c) }
    println("Введите элементы $name матрицы:")
    for (i in 0 until r) {
        for (j in 0 until c) {
            print("Элемент [$i][$j]:")
            matrix[i][j] = readLine()!!.toInt()
        }
    }
    return matrix
}

fun printMatrix(matrix: Array<DoubleArray>) {
    for (row in matrix) {
        println(row.joinToString(" "))
    }
}

fun printMatrix(matrix: Array<IntArray>) {
    for (row in matrix) {
        println(row.joinToString(" "))
    }
}

fun main() {
    // универс функц для заполнения двух матриц
    val matrix1 = inputMatrix("первой")
    val matrix2 = inputMatrix("второй")

    val transpose: (Array<IntArray>) -> Array<IntArray> = { matrix ->
        Array(matrix[0].size) { col ->
            IntArray(matrix.size) { row -> matrix[row][col] }
        }
    }
    val transposedMatrix = transpose(matrix1)
    println("Транспонирование:")
    printMatrix(transposedMatrix)
```

```

        val merge = fun(mat1: Array<IntArray>, mat2: Array<IntArray>):
Array<IntArray> {
            return Array(mat1.size) { i -> mat1[i] + mat2[i] }
        }
        val mergedMatrix = merge(matrix1, matrix2)
        println("Слияние:")
        printMatrix(mergedMatrix)

        val normalizedMatrix = normalize(matrix1)
        println("Нормализация:")
        printMatrix(normalizedMatrix)
    }

    fun normalize(matrix: Array<IntArray>): Array<DoubleArray> {
        val max = matrix.flatMap { it.asList() }.maxOrNull()?.toDouble() ?:
1.0
        return Array(matrix.size) { i ->
            DoubleArray(matrix[i].size) { j -> matrix[i][j] / max }
        }
    }
}

```

## Контрольные вопросы:

### 1. Какие типы циклов существуют в Kotlin?

В Kotlin есть три типа циклов: `for`: для итерации по коллекциям, массивам или диапазонам. `while`: выполняется, пока условие истинно. `do...while`: выполняется хотя бы один раз, затем проверяется условие.

### 2. Как работает цикл `for` для итерации по элементам массива?

`val array = arrayOf(1, 2, 3) for (element in array) { println(element)}` Этот цикл проходит по каждому элементу массива `array` и выводит его.

### 3. Что такое диапазон в Kotlin, и как его создать?

Диапазон — это последовательность значений, созданная с помощью оператора `...`. Например: `val range = 1..5` // Диапазон от 1 до 5 включительно