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

Отчёт

по лабораторной работе №2

на тему

**Циклы, диапазоны и массивы**

Выполнил: Проверил:

ст. гр. 214302 Усенко Ф.В.

Петрусевич Д.А.

Минск 2024

Задание: **Анализ и оптимизация последовательностей:** Создайте программу, которая принимает последовательность чисел, анализирует её на предмет наличия циклов, возрастающих и убывающих последовательностей, а затем выводит оптимизированный вариант последовательности, удаляя избыточные элементы.

Листинг кода:

import kotlin.math.sqrt  
import kotlin.math.abs  
import kotlin.random.Random  
  
fun main() {  
 val size = 3  
 while (true) {  
 println("Нажмите '1' для генерации новой матрицы или 'q' для выхода:")  
 val input = readLine()  
  
 when (input) {  
 "1" -> generateNewMatrix(size)  
 "q" -> {  
 println("Выход из программы.")  
 return  
 }  
 else -> println("Неверный ввод. Пожалуйста, нажмите '1' или 'q'.")  
 }  
 }  
}  
  
private fun generateNewMatrix(size: Int) {  
 val matrix = generateRandomMatrix(size)  
 printMatrix(matrix)  
 println("Диагональная доминантность: ${isDiagonallyDominant(matrix)}")  
 println("Положительная определенность: ${isPositiveDefinite(matrix)}")  
 println("Симметричность: ${isSymmetric(matrix)}")  
}  
  
private fun generateRandomMatrix(size: Int): Array<DoubleArray> {  
 val matrix = Array(size) { DoubleArray(size) }  
 for (i in 0 until size) {  
 for (j in 0 until size) {  
 matrix[i][j] = Random.nextDouble(1.0, 50.0)  
 }  
 }  
 return matrix  
}  
  
private fun printMatrix(matrix: Array<DoubleArray>) {  
 for (row in matrix) {  
 println(row.joinToString("\t") { String.format("%.0f", it) })  
 }  
}  
  
private fun isDiagonallyDominant(matrix: Array<DoubleArray>): Boolean {  
 for (i in matrix.indices) {  
 val diagonal = abs(matrix[i][i])  
 val sum = matrix[i].filterIndexed { j, \_ -> i != j }.sumOf { abs(it) }  
 if (diagonal <= sum) return false  
 }  
 return true  
}  
  
private fun isPositiveDefinite(matrix: Array<DoubleArray>): Boolean {  
 val cholesky = choleskyDecomposition(matrix)  
 return cholesky != null  
}  
  
private fun choleskyDecomposition(matrix: Array<DoubleArray>): Array<DoubleArray>? {  
 val size = matrix.size  
 val result = Array(size) { DoubleArray(size) }  
  
 for (i in 0 until size) {  
 for (j in 0 until i + 1) {  
 var sum = 0.0  
 for (k in 0 until j) {  
 sum += result[i][k] \* result[j][k]  
 }  
 result[i][j] = if (i == j) {  
 sqrt(matrix[i][i] - sum)  
 } else {  
 (matrix[i][j] - sum) / result[j][j]  
 }  
 }  
 if (result[i][i] <= 0) return null  
 }  
 return result  
}  
  
private fun isSymmetric(matrix: Array<DoubleArray>): Boolean {  
 for (i in matrix.indices) {  
 for (j in i + 1 until matrix[i].size) {  
 if (matrix[i][j] != matrix[j][i]) return false  
 }  
 }  
 return true  
}

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.

1. **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