Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

**«Программирование мобильных информационных систем»**

**Лабораторная работа № 2**

«Циклы, диапазоны и массивы»

Проверил: Выполнил:

Усенко Ф.В. Лисовский О.Ю.

Минск 2024

**Цель работы:**

Закрепить понимание работы циклов, диапазонов и массивов в Kotlin. Научиться эффективно использовать эти конструкции для решения различных задач.

Вариант № 12

1. Задание: Напишите программу, которая принимает многомерный массив данных, вычисляет основные статистические показатели (среднее, медиана, дисперсия) и выводит результат в удобном формате.

import kotlin.math.pow

fun mean(data: List<Double>): Double {

return if (data.isNotEmpty()) data.sum() / data.size else 0.0

}

fun median(data: List<Double>): Double? {

if (data.isEmpty()) return null

val sortedData = data.sorted()

return if (sortedData.size % 2 == 0) {

(sortedData[sortedData.size / 2 - 1] + sortedData[sortedData.size / 2]) / 2

} else {

sortedData[sortedData.size / 2]

}

}

fun variance(data: List<Double>): Double? {

if (data.size < 2) return null // Нужны как минимум 2 элемента для расчета дисперсии

val meanValue = mean(data)

return data.map { (it - meanValue).pow(2) }.sum() / data.size

}

fun analyzeData(data: Array<Array<Double>>) {

for ((index, row) in data.withIndex()) {

val rowData = row.toList()

println("Ряд $index:")

if (rowData.isEmpty()) {

println(" Недостаточно данных для анализа")

} else {

val meanValue = mean(rowData)

val medianValue = median(rowData)

val varianceValue = variance(rowData)

println(" Среднее: $meanValue")

if (medianValue != null) {

println(" Медиана: $medianValue")

} else {

println(" Невозможно посчитать медиану: недостаточно данных")

}

if (varianceValue != null) {

println(" Дисперсия: $varianceValue")

} else {

println(" Невозможно посчитать дисперсию: недостаточно данных")

}

}

println()

}

}

fun manualInput(): Array<Array<Double>> {

println("Введите количество рядов:")

val numberOfRows = readLine()?.toIntOrNull() ?: 0

val data = Array(numberOfRows) { arrayOf<Double>() }

for (i in 0 until numberOfRows) {

println("Введите элементы для ряда ${i + 1}, разделенные пробелом:")

val rowInput = readLine() ?: ""

val rowData = rowInput.split(" ").mapNotNull { it.toDoubleOrNull() }.toTypedArray()

data[i] = rowData

}

return data

}

fun automaticInput(): Array<Array<Double>> {

return arrayOf(

arrayOf(1.0, 2.0, 3.0, 4.0, 5.0),

arrayOf(10.0, 20.0, 30.0, 40.0, 50.0),

arrayOf(100.0, 200.0, 300.0, 400.0, 500.0),

arrayOf() // Пустой ряд для тестирования

)

}

fun main() {

while (true) {

println("Выберите способ ввода данных:")

println("1. Ввести данные вручную")

println("2. Заполнить данные автоматически")

println("0. Выйти из программы")

val choice = readLine()?.toIntOrNull()

if (choice == 0) {

println("Выход из программы.")

break

}

val data = when (choice) {

1 -> manualInput()

2 -> automaticInput()

else -> {

println("Некорректный выбор, данные будут заполнены автоматически.")

automaticInput()

}

}

println("\nАнализ многомерных данных:")

analyzeData(data)

println("Хотите повторить анализ? (да/нет)")

val repeat = readLine()

if (repeat?.toLowerCase() != "да") {

println("Выход из программы.")

break

}

}

}

Вывод: в ходе работы закрепил понимание работы циклов, диапазонов и массивов в Kotlin, научился эффективно использовать эти конструкции для решения различных задач. Создал программу, которая решает квадратное уравнение, находит корни уравнения и целые решения в заданном пользователем диапазоне. Так же решает квадратные уравнения, в которых отрицательный дискриминант и получает комплексный ответ.