Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем

Отчёт

по лабораторной работе №3

на тему:

**ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ**

**И ЛЯМБДА-ВЫРАЖЕНИЯ**

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ф.В. Усенко

(подпись)

Выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Предченко В.М.

(подпись) гр. 214301

Минск, 2024

**Цель:** Разработать программное средство, которое принимает большой массив данных, разбивает его на части и параллельно выполняет анализ с использованием лямбда-выражений.

**Код**

fun main() {  
 val size = 1\_000\_000  
 val numbers = List(size) { (1..100).random() }  
  
 // Разбивка массива на части  
 val chunkSize = (size / Runtime.getRuntime().availableProcessors()) + 1  
 val results = mutableListOf<DoubleArray>()  
  
 // Параллельный анализ (симуляция параллелизма)  
 numbers.chunked(chunkSize).forEach { chunk ->  
 val sum = chunk.sum()  
 val average = chunk.average()  
 val median = chunk.sorted().let { sorted ->  
 if (sorted.size % 2 == 0) {  
 (sorted[sorted.size / 2 - 1] + sorted[sorted.size / 2]) / 2.0  
 } else {  
 sorted[sorted.size / 2].toDouble()  
 }  
 }  
 results.add(doubleArrayOf(sum.toDouble(), average, median))  
 }  
  
  
 val totalSum = results.sumOf { it[0] }  
 val overallAverage = totalSum / size  
 val overallMedians = results.map { it[2] }.sorted()  
 val overallMedian = if (overallMedians.size % 2 == 0) {  
 (overallMedians[overallMedians.size / 2 - 1] + overallMedians[overallMedians.size / 2]) / 2.0  
 } else {  
 overallMedians[overallMedians.size / 2]  
 }  
  
 println("Total Sum: $totalSum")  
 println("Overall Average: $overallAverage")  
 println("Overall Median: $overallMedian")  
}

**Контрольные вопросы**

**1 Как объявить функцию в Kotlin? В чем разница между обычной функцией и однострочной функцией?**

В Kotlin функцию можно объявить с помощью ключевого слова fun. Разница между обычной и однострочной функцией заключается в том, что обычная функция может содержать несколько строк кода, и для ее тела обычно используются фигурные скобки. Однострочная функция использует выражение, котрое возвращает результат, и может не требовать фигурных скобок, если тело функции состоит из одного выражения.

**2 Что такое функция высшего порядка, и как её использовать? Приведите пример.**

Функция высшего порядка – это функция, которая принимает другую функцию в качестве параметра или возвращает ее.

Пример:

fun operateOnNumbers(a: Int, b: Int, operation: (Int, Int) -> Int): Int {

return operation(a, b)

}

fun main() {

val sum = operateOnNumbers(5, 3, { x, y -> x + y })

println("Sum: $sum") // Вывод: Sum: 8

val product = operateOnNumbers(5, 3, { x, y -> x \* y })

println("Product: $product") // Вывод: Product: 15

}

**3 Как передать функцию в качестве параметра другой функции? Приведите пример.**

Функцию можно передать в качестве параметра, указав её тип. Тип функции определяется как (Тип\_параметра1, Тип\_параметра2) -> Тип\_возвращаемого\_значения.

Пример:

fun calculate(a: Int, b: Int, operation: (Int, Int) -> Int): Int {

return operation(a, b)

}

fun main() {

val result = calculate(10, 5, ::add) // Используем ссылку на функцию

println("Result: $result") // Вывод: Result: 15

}

fun add(x: Int, y: Int): Int {

return x + y

}

**Вывод**: Было изучено функциональное программирование. Была выявлена разница обычных функций от однострочной, а также познакомилась с лямбда-функцией.