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 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сердюк К.В.

(подпись) гр. 214301

Минск, 2024

**Цель: Обработка данных с функциями высшего порядка:** Напишите программу, которая принимает список данных и несколько функций высшего порядка для их обработки. Программа должна комбинировать результаты выполнения этих функций и выводить общий результат.

**Код**

fun main() {

val numbers = listOf(10, 33, 9)

val sum: (List<Int>) -> Int = {

val sum = it.sum()

println("Сумма чисел $it: $sum")

sum

}

val pr: (List<Int>) -> Int = {

val pr = it.reduce { acc, i -> acc \* i }

println("Произведение чисел $it: $pr")

pr

}

val max: (List<Int>) -> Int = {

val max = it.maxOrNull() ?: 0

println("Наибольшее значение среди чисел $it: $max")

max

}

val min: (List<Int>) -> Int = {

val min = it.minOrNull() ?: 0

println("Наименьшее значение среди чисел $it: $min")

min

}

val finalResult = performCalculations(

numbers, ::processData, ::sumData, listOf(sum, pr, max, min)

)

println("Финальный результат: $finalResult")

val counter = createCounter()

println(counter())

println(counter())

println(counter())

println(counter())

}

fun <T> processData(data: List<T>, operations: List<(List<T>) -> Any>): List<Any> {

return operations.map { it(data) }

}

fun <T> sumData(data: List<T>, operations: List<(List<T>) -> Any>): Int {

return operations.sumOf {

val result = it(data)

if (result is Int) result else 0

}

}

fun performCalculations(

numbers: List<Int>,

processFn: (List<Int>, List<(List<Int>) -> Any>) -> List<Any>,

sumFn: (List<Int>, List<(List<Int>) -> Any>) -> Int,

operations: List<(List<Int>) -> Any>

): String {

val processResults = processFn(numbers, operations)

println("Результаты processData: $processResults")

val sumResult = sumFn(numbers, operations)

println("Результат sumData: $sumResult")

val average = sumResult / operations.size

return "Сумма: $sumResult, Среднее значение: $average"

}

fun createCounter(): () -> Int{

var counter = 0

return {++counter}

}

**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.