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.

Васильков Н.О.

Минск 2024

Задание: Поиск подмассивов с заданной суммой: Напишите программу, которая находит все подмассивы в заданном массиве, сумма элементов которых равна заданному значению. Программа должна учитывать различные длины подмассивов и выводить их индексы.

Листинг кода:

fun getInputArray(): IntArray {  
  
 while (true) {  
 try {  
 print("Введите массив чисел через пробел: ")  
 val input = readLine()  
  
 if (input.isNullOrBlank()) {  
 throw NullPointerException("Ввод не может быть пустым.")  
 }  
  
 // Преобразуем строку в массив целых чисел  
 val array = input.split(" ").map **{** it.toInt() **}**.toIntArray()  
  
 // Если массив пустой, то бросаем исключение  
 if (array.isEmpty()) {  
 throw IllegalArgumentException("Массив не может быть пустым.")  
 }  
  
 return array  
 } catch (e: NullPointerException) {  
 println(e.message)  
 } catch (e: NumberFormatException) {  
 println("Неправильный ввод. Убедитесь, что все элементы - это целые числа.")  
 } catch (e: IllegalArgumentException) {  
 println(e.message)  
 }  
 }  
}  
  
fun getTargetSum(): Int {  
 while (true) {  
 try {  
 print("Введите целевую сумму: ")  
 val input = readLine()  
  
 if (input.isNullOrBlank()) {  
 throw NullPointerException("Ввод не может быть пустым.")  
 }  
  
 val targetSum = input.toInt()  
 return targetSum  
 } catch (e: NullPointerException) {  
 println(e.message)  
 } catch (e: NumberFormatException) {  
 println("Неправильный ввод. Введите корректное целое число.")  
 }  
 }  
}  
  
fun findSubarraysWithSum(arr: IntArray, targetSum: Int) {  
 var found = false // Флаг, который отслеживает, были ли найдены подмассивы  
  
 for (i in 0 until arr.size ) {  
 var currentSum = 0  
 for (j in i until arr.size) {  
 currentSum += arr[j]  
 if (currentSum == targetSum && j>i) {  
 println("Подмассив с суммой $targetSum найден: индексы [$i, $j]")  
 found=true  
 }  
 }  
 }  
 if (!found) {  
 println("Нет подмассива с суммой $targetSum.")  
 }  
}  
  
fun main() {  
  
 while (true) {  
 val arr = *getInputArray*() // Получаем массив от пользователя  
 val targetSum = *getTargetSum*() // Получаем целевую сумму от пользователя  
  
 *findSubarraysWithSum*(arr, targetSum) // Находим подмассивы с заданной суммой  
  
 while (true) {  
 print("Хотите продолжить? (y/n): ")  
 val input = readLine()  
 when (input) {  
 "y" -> break // Выходим из внутреннего цикла, чтобы продолжить выполнение  
 "n" -> {  
 println("Программа завершена.")  
 return // Выходим из main, чтобы завершить программу  
 }  
 else -> println("Некорректный ввод. Введите 'y' для продолжения или 'n' для выхода.")  
 }  
 }  
 }  
  
}

Контрольные вопросы:

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

В Kotlin есть три основных типа циклов:

1. **for** — используется для итерации по элементам коллекций, массивов, диапазонов и других итерируемых объектов.
   * Применяется, когда известно количество элементов или диапазон для перебора.
2. **while** — выполняет код, пока указанное условие истинно.
   * Используется, когда количество итераций заранее неизвестно, и условие проверяется перед выполнением цикла.
3. **do-while** — выполняет код как минимум один раз, а затем проверяет условие.
   * Используется, когда необходимо выполнить тело цикла хотя бы один раз перед проверкой условия.
4. **Чем полезны методы joinToString и sort для работы с массивами?**

Метод joinToString используется для создания строки из элементов массива, задавая разделитель, префикс, суффикс и другие параметры.

val array = arrayOf(1, 2, 3, 4, 5)

val result = array.joinToString(

separator = ", ",

prefix = "[",

postfix = "]"

)

println(result) // Вывод: [1, 2, 3, 4, 5]

Метод sort изменяет порядок элементов массива, упорядочивая их по возрастанию (или в соответствии с указанным компаратором).

**Пример использования:**

kotlin

val array = arrayOf(5, 3, 1, 4, 2)

array.sort()

println(array.joinToString()) // Вывод: 1, 2, 3, 4, 5

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