Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем

Дисциплина: Программирование мобильных информационных систем

Отчёт

К лабораторной работе № 2

на тему:

**ЦИКЛЫ, ДИАПАЗОНЫ, МАССИВЫ**

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

(подпись)

Выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Шубин

(подпись) 214302

Минск, 2024

**Цель работы:** закрепить понимание работы циклов, диапазонов и массивов в Kotlin. Научиться эффективно использовать эти конструкции для решения различных задач.

**Задание для самостоятельной работы (Вариант 8).**

**Пересечение и объединение массивов с условиями:** Напишите программу, которая находит пересечение и объединение двух массивов чисел, но с учетом заданных пользователем условий (например, числа должны быть четными и больше 10).

**Код программы.**

fun main() {  
  
 print("Введите желаемое количество элементов в массивах: ")  
 var size = readlnOrNull()?.toIntOrNull()  
 while(size == null) {  
 println("Ошибка ввода. Попробуйте ещё раз")  
 size = readlnOrNull()?.toIntOrNull()  
 continue  
 }  
  
 while (true) {  
  
 var cycleCount = 0  
 print("Введите элементы 1-го массива из $size чисел через пробел: ")  
 val array1 = readlnOrNull()?.split(" ")?.filter { it.isNotEmpty() }?.map { it.toInt() }  
  
 if (array1 == null || array1.size != size) {  
 println("Пожалуйста, введите $size чисел.")  
 continue  
 }  
  
 print("Введите элементы 2-го массива из $size чисел через пробел: ")  
 val array2 = readlnOrNull()?.split(" ")?.filter { it.isNotEmpty() }?.map { it.toInt() }  
  
 if (array2 == null || array2.size != size) {  
 println("Пожалуйста, введите $size чисел.")  
 continue  
 }  
 val userCondition1: (Int) -> Boolean = userMenu()  
 var complexCondition: (Int) -> Boolean = { true }  
 while(cycleCount < 1) {  
 println("Желаете выбрать дополнительное условие? (это можно сделать только один раз. " +  
 "Выбрать более двух условий для фильтрации нельзя)\n1. ДА\nНЕТ - любая другая клавиша")  
 val input = readlnOrNull()?.toIntOrNull()  
 complexCondition = if(input == 1){  
 val userCondition2: (Int) -> Boolean = userMenu()  
 ({ num -> userCondition1(num) && userCondition2(num) })  
 }else {  
 userCondition1  
 }  
 cycleCount++  
 }  
  
 val (intersection, union) = filterArrays(complexCondition, array1, array2)  
  
 println("Пересечение: $intersection")  
 println("Объединение: $union")  
 break  
 }  
}  
  
fun filterArrays(condition: (Int) -> Boolean, array1: List<Int>, array2: List<Int>):  
 Pair<Set<Int>, Set<Int>> {  
 val filteredArray1 = array1.filter(condition)  
 val filteredArray2 = array2.filter(condition)  
  
 val intersection = filteredArray1.intersect(filteredArray2)  
 val union = filteredArray1.union(filteredArray2)  
  
 return Pair(intersection, union)  
}  
  
fun userMenu(): (Int) -> Boolean {  
 while(true) {  
 println(  
 "Выберите условие для фильтрации:" +  
 "\n1. Числа должны быть чётными\n2. Числа должны быть нечётными" +  
 "\n3. Числа должны делиться без остатка на выбранное число" +  
 "\n4. Числа должны быть больше указанного числа" +  
 "\n5. Числа должны быть меньше указанного числа"  
 )  
  
 var input = readlnOrNull()?.toIntOrNull()  
 var userCondition: (Int) -> Boolean  
  
 val condition = when (input) {  
 1 -> {  
 userCondition = {it % 2 == 0}  
 }  
 2 ->{  
 userCondition = {it % 2 != 0}  
 }  
 3 -> {  
 println("Введите желаемый делитель: ")  
 val customDivider = readlnOrNull()?.toIntOrNull()  
 userCondition = {it % customDivider!! == 0}  
 }  
 4 -> {  
 println("Введите число для сравнения: ")  
 val customComparedNumber = readlnOrNull()?.toIntOrNull()  
 userCondition = {it > customComparedNumber!! }  
 }  
 5 -> {  
 println("Введите число для сравнения: ")  
 val customComparedNumber = readlnOrNull()?.toIntOrNull()  
 userCondition = {it < customComparedNumber!! }  
 }  
 else -> {  
 println("Ошибка ввода. Выберите существующий пункт меню")  
 continue  
 }  
 }  
 return userCondition  
 }  
}

**Контрольные вопросы.**

1. Какие типы циклов существуют в Kotlin, и как они используются?

В Kotlin существуют следующие типы циклов:

* **Цикл for:** Используется для выполнения набора инструкций заданное количество раз или для итерации по элементам коллекции (например, массиву).
* **Цикл while:** Выполняет блок кода до тех пор, пока условие истинно. Если условие становится ложным, выполнение цикла прекращается.
* **Цикл do...while:** Отличается от while тем, что сначала выполняется тело цикла, а затем проверяется условие.

1. Какую роль выполняет цикл do...while, и в чем его отличие от while?

Цикл do...whileв отличие от while сначала выполняет тело цикла, а затем проверяет условие. Его роль – выполнение определенного блока кода до тех пор, пока заданное условие истинно.

1. Что произойдет, если использовать оператор !in с диапазоном? Приведите пример.

Оператор !in проверяет, что заданный элемент не входит в определенный диапазон.

Например, в данном участке кода проверяется, что переменная x со значением 6 не входит в диапазон от одного до пяти:

val x = 6

if (x !in 1..5) {

println("$x is NOT in the range")

}

1. Что такое массив фиксированного размера, и как его создать в Kotlin?

Если известно, что массив не будет изменяться по размеру, можно использовать массив фиксированного размера. Пример: val numbers = IntArray(5).

1. Как изменить значение элемента массива в Kotlin? Приведите пример.

Элементы массива могут быть изменены с использованием их индексов. Пример:

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

numbers[0] = 10

println(numbers[0])

1. Как перебрать элементы массива с использованием цикла for?

Элементы массива могут быть перебраны с использованием цикла for. Пример:

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

for (number in numbers) {

println(number)

}