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.

Карлюк П.А.

Минск 2024

Вариант 8.Напишите программу, которая находит пересечение и объединение двух массивов чисел, но с учетом заданных пользователем условий (например, числа должны быть четными и больше 10).

Листинг кода:

fun main() {  
  
 var length: Int  
  
 *print*("Введите длину первого массива: ")  
 length = *generateLength*()  
 var firstArray: Array<Float>  
 firstArray = *generateArray*(length)  
  
 *print*("Введите длину второго массива: ")  
 length = *generateLength*()  
 var secondArray: Array<Float>  
 secondArray = *generateArray*(length)  
  
 var operations: String  
  
 while(true){  
 var max : Float  
 var min : Float  
  
 *println*("Для выхода и отображения - 0")  
 *println*("Введите операцию (>, >=, <, <=, четность, нечетность): ")  
 operations = *readLine*().*toString*().*lowercase*()  
  
 when(operations){  
 ">" -> {  
  
 *println*("Введите число: ")  
 while (true){  
 max = *readLine*()?.*toFloatOrNull*()!!  
 if(!max.equals(null) ){  
 firstArray = firstArray.*filter***{ it** > max**}**.*toTypedArray*()  
 secondArray = secondArray.*filter***{ it** > max**}**.*toTypedArray*()  
 break  
 }  
 else{  
 *println*("Некорретный ввод. Попробуйте снова")  
 }  
 }  
 }  
 "<" -> {  
  
 *println*("Введите число: ")  
 while (true){  
 min = *readLine*()?.*toFloatOrNull*()!!  
 if(!min.equals(null) ){  
 firstArray = firstArray.*filter***{ it** < min**}**.*toTypedArray*()  
 secondArray = secondArray.*filter***{ it** < min**}**.*toTypedArray*()  
 break  
 }  
 else{  
 *println*("Некорретный ввод. Попробуйте снова")  
 }  
 }  
  
 }  
  
 ">=" -> {  
  
 *println*("Введите число: ")  
 while (true){  
 max = *readLine*()?.*toFloatOrNull*()!!  
 if(!max.equals(null) ){  
 firstArray = firstArray.*filter***{ it** >= max**}**.*toTypedArray*()  
 secondArray = secondArray.*filter***{ it** >= max**}**.*toTypedArray*()  
 break  
 }  
 else{  
 *println*("Некорретный ввод. Попробуйте снова")  
 }  
 }  
  
  
 }  
 "<=" -> {  
  
 *println*("Введите число: ")  
 while (true){  
 min = *readLine*()?.*toFloatOrNull*()!!  
 if(!min.equals(null) ){  
 firstArray = firstArray.*filter***{ it** <= min**}**.*toTypedArray*()  
 secondArray = secondArray.*filter***{ it** <= min**}**.*toTypedArray*()  
 break  
 }  
 else{  
 *println*("Некорретный ввод. Попробуйте снова")  
 }  
 }  
 }  
  
 "четность" ->{  
  
 firstArray = firstArray.*filter***{ it** % 2 == 0f **}**.*toTypedArray*()  
 secondArray = secondArray.*filter***{ it** % 2 == 0f **}**.*toTypedArray*()  
  
 }  
 "нечетность" ->{  
  
 firstArray = firstArray.*filter***{ it** % 2 != 0f **}**.*toTypedArray*()  
 secondArray = secondArray.*filter***{ it** % 2 != 0f **}**.*toTypedArray*()  
  
 }  
 "0" ->{  
  
 val intersection = firstArray.*intersect*(secondArray.*toSet*())  
  
 val union = firstArray.*union*(secondArray.*toSet*())  
  
 *println*("Пересечение: $intersection")  
 *println*("Объединение: $union")  
 break  
  
 }  
 else -> {  
 *println*("Некорректный ввод. Попробуйте снова")  
 }  
 }  
 }  
}  
  
fun generateLength(): Int{  
  
 while(true) {  
 val input = *readLine*()?.*toIntOrNull*()  
 if(input != null && input > 0) {  
 return input.toInt()  
 }  
 else{  
 *println*("Неккоретный ввод. Попробуйте снова")  
 }  
 }  
  
}  
  
fun generateArray(length: Int) : Array<Float> {  
  
 *println*("Введите $length элементов массива: ")  
  
 var counter = 0  
 val array = FloatArray(length)  
 while(counter < length) {  
  
 val input = *readLine*()?.*toFloatOrNull*()  
  
 if(input != null) {  
 array[counter] = input.toFloat()  
 counter++  
 }  
 else{  
 *println*("Неккоретный ввод. Попробуйте снова")  
 }  
 }  
  
 return array.*toTypedArray*()  
  
}

Контрольные вопросы:

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

В *Kotlin* существуют следующие типы циклов:

* *for*: используется для итерации по коллекциям, массивам и диапазонам.
* *while*: выполняет блок кода, пока условие истинно.
* *do*...*while*: выполняет блок кода хотя бы один раз, затем проверяет условие.

1. Что такое шаг диапазона, и как его задать? Приведите пример.

Шаг диапазона – это количество, на которое увеличивается (или уменьшается) значение при каждой итерации цикла. Он задается с помощью ключевого слова *step*.

for (i in 1..10 step 2) {

println(i) // Выведет: 1, 3, 5, 7, 9

}

1. Как подсчитать количество элементов в массиве, соответствующих заданному условию?

Для подсчета количества элементов в массиве, соответствующих заданному условию, можно использовать метод *count*:

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

val count = numbers.count { it > 3 } // 2