Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Отчет

по практической работе №2

на тему:

**ЦИКЛЫ, ДИАПАЗОНЫ И МАССИВЫ**

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

(подпись)

Выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Папко К.А.

(подпись)  гр. 214301

Минск, 2024

**Задание:** Напишите программу, которая создает магический квадрат заданного размера (матрица, где суммы чисел в каждой строке, столбце и диагонали равны). Программа должна проверять, что введенное число является допустимым для создания магического квадрата.

1. ВЫПОЛНЕНИЕ ЗАДАНИЯ

import kotlin.system.exitProcess  
  
fun main() {  
 println("Введите размер магического квадрата (нечетное число или кратное 4):")  
  
 while(true){  
 val n = readln().toIntOrNull()  
  
 if (n == null || !isValidMagicSquareSize(n)) {  
 println("Ошибка: размер магического квадрата должен быть нечетным числом или кратным 4.")  
 exitProcess(1)  
 }  
  
 val magicSquare = generateMagicSquare(n)  
 printMagicSquare(magicSquare)  
 }  
  
}  
  
fun isValidMagicSquareSize(n: Int): Boolean {  
 return n % 2 != 0 || n % 4 == 0  
}  
  
fun generateMagicSquare(n: Int): Array<IntArray> {  
 return if (n % 2 != 0) {  
 generateOddMagicSquare(n)  
 } else {  
 generateDoublyEvenMagicSquare(n)  
 }  
}  
  
fun generateOddMagicSquare(n: Int): Array<IntArray> {  
 val magicSquare = Array(n) **{** IntArray(n) **}** var number = 1  
 var i = 0  
 var j = n / 2  
  
 while (number <= n \* n) {  
 magicSquare[i][j] = number  
 number++  
  
 val previousI = i  
 val previousJ = j  
  
 i -= 1  
 j += 1  
  
 if (i < 0) i = n - 1  
 if (j == n) j = 0  
  
 if (magicSquare[i][j] != 0) {  
 i = previousI + 1  
 j = previousJ  
 }  
 }  
  
 return magicSquare  
}  
  
fun generateDoublyEvenMagicSquare(n: Int): Array<IntArray> {  
 val magicSquare = Array(n) **{** IntArray(n) **}** var number = 1  
 var reverseNumber = n \* n  
  
 for (i in 0 until n) {  
 for (j in 0 until n) {  
 if (i % 4 == j % 4 || (i + j) % 4 == 3) {  
 magicSquare[i][j] = reverseNumber  
 } else {  
 magicSquare[i][j] = number  
 }  
 number++  
 reverseNumber--  
 }  
 }  
  
 return magicSquare  
}  
  
fun printMagicSquare(magicSquare: Array<IntArray>) {  
 val n = magicSquare.size  
 println("Магический квадрат размера $n:")  
 for (i in 0 until n) {  
 for (j in 0 until n) {  
 print("${magicSquare[i][j]}\t")  
 }  
 println()  
 }  
}

1. ОТВЕТЫ НА ВОПРОСЫ
2. Какие типы циклов существуют в Kotlin, и как они используются?  
   В Kotlin существуют три типа циклов: for, while, и do...while. Цикл for используется для итерации по коллекциям и диапазонам, цикл while — для выполнения кода, пока условие истинно, а do...while выполняет код хотя бы один раз перед проверкой условия.
3. Как работает цикл for для итерации по элементам массива?  
   Цикл for последовательно перебирает элементы массива, предоставляя доступ к каждому из них внутри тела цикла.
4. Чем отличается цикл while от цикла for?  
   Цикл while используется для выполнения блока кода, пока условие истинно, и чаще применяется, если заранее неизвестно количество итераций. Цикл for применяется для итерации по известной последовательности или диапазону.
5. Какую роль выполняет цикл do...while, и в чем его отличие от while?  
   Цикл do...while выполняет код хотя бы один раз, даже если условие изначально ложно, поскольку проверка условия происходит после выполнения тела цикла.
6. Что такое диапазон в Kotlin, и как его создать?  
   Диапазон в Kotlin — это последовательность значений, создаваемая с помощью оператора ... Его можно использовать, например, для итераций в цикле for.
7. ВЫВОД

В ходе выполнения практической работы были изучены циклы, диапазоны и массивы в языке программирования Kotlin. Рассмотрены основные виды циклов и их применение в различных сценариях.