Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем

Отчёт

по лабораторной работе №2

на тему:

**ЦИКЛЫ, ДИАПАЗОНЫ И МАССИВЫ**

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

(подпись)

Выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. Ю. Позняк

(подпись) гр. 214302

Минск, 2024

**ЗАДАНИЕ**

Генерация магического квадрата: 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.

**РЕШЕНИЕ**

import kotlin.random.Random

fun main() {

var n: Int

while (true) {

println("Введите размер магического квадрата :")

n = readLine()?.toIntOrNull() ?: 0

if (n < 1) {

println("Размер должен быть положительным числом.")

} else {

printMagicSquare(generateMagicSquare(n))

break

}

}

}

fun generateMagicSquare(n: Int): Array<IntArray> {

println("Загрузка...")

while (true) {

val magicSquare = Array(n) { IntArray(n) { Random.nextInt(0, 10) } }

if (isMagicSquare(magicSquare)) {

return magicSquare

}

}

}

fun isMagicSquare(square: Array<IntArray>): Boolean {

val n = square.size

val sums = mutableListOf<Int>()

// Считает сумму строк

for (i in 0 until n) {

sums.add(square[i].sum())

}

// Считает сумму столбцов

for (i in 0 until n) {

sums.add(square.sumOf { it[i] })

}

// Считает сумму диагоналей

sums.add(square.indices.sumOf { square[it][it] })

sums.add(square.indices.sumOf { square[it][n - it - 1] })

// Проверяет, все ли суммы равны

return sums.distinct().size == 1 // Количество уникальных значений == 1

}

fun printMagicSquare(magicSquare: Array<IntArray>) {

for (row in magicSquare) {

for (num in row) {

print("$num\t")

}

println()

}

}

**КОНТРОЛЬНЫЕ ВОПРОСЫ**

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

**Цикл *for*:** Используется для выполнения набора инструкций заданное количество раз или для итерации по элементам коллекции.

**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. Чем отличается цикл *while* от цикла *for*?

*While* выполняется, пока условие истинно, а *for* проходит по элементам коллекции или диапазона.

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

Цикл *do...while* в программировании используется для выполнения блока кода хотя бы один раз, а затем повторения этого блока, пока условие истинно. Основное отличие от цикла *while* заключается в том, что проверка условия в *do...while* происходит после выполнения тела цикла, а не до.

1. Что такое диапазон в *kotlin*, и как его создать?

Диапазон в *Kotlin* представляет собой последовательность значений, определенную начальным и конечным значениями. Диапазоны могут быть использованы для итерации в циклах и проверки принадлежности значений.

val range = 1..5

val reverseRange = 5 downTo 1

val stepRange = 1..10 step 2