Министерство науки и высшего образования Российской Федерации

**ФГБОУ ВО «Кубанский государственный технологический университет»**

**(ФГБОУ ВО «КубГТУ»)**

Институт компьютерных систем и информационной безопасности

Кафедра информационных систем и программирования

Направление 10.05.01 Компьютерная безопасность

**ОТЧЕТ по практическому занятию №5**

по дисциплине: «Языки программирования» на тему:

«Многомерные массивы.»

 Выполнил студент: Рудометкин Станислав Владимирович, 1 курс, группа 22-К-КБ1

Допущена к защите\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель работы,

 профессор                                                                                                      Г.С. Петриченко

Краснодар

2023

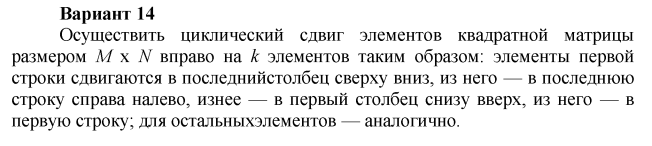
**ПРАКТИЧЕСКАЯ РАБОТА №5**

**Тема**: Многомерные массивы

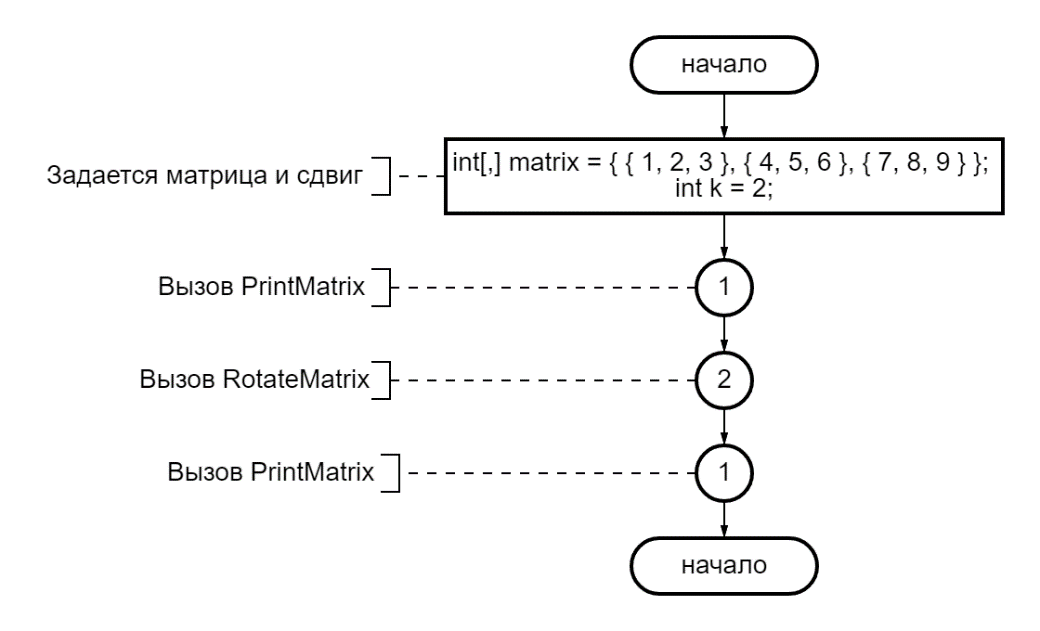
**Цель**: изучить методы алгоритмизации и программирования с

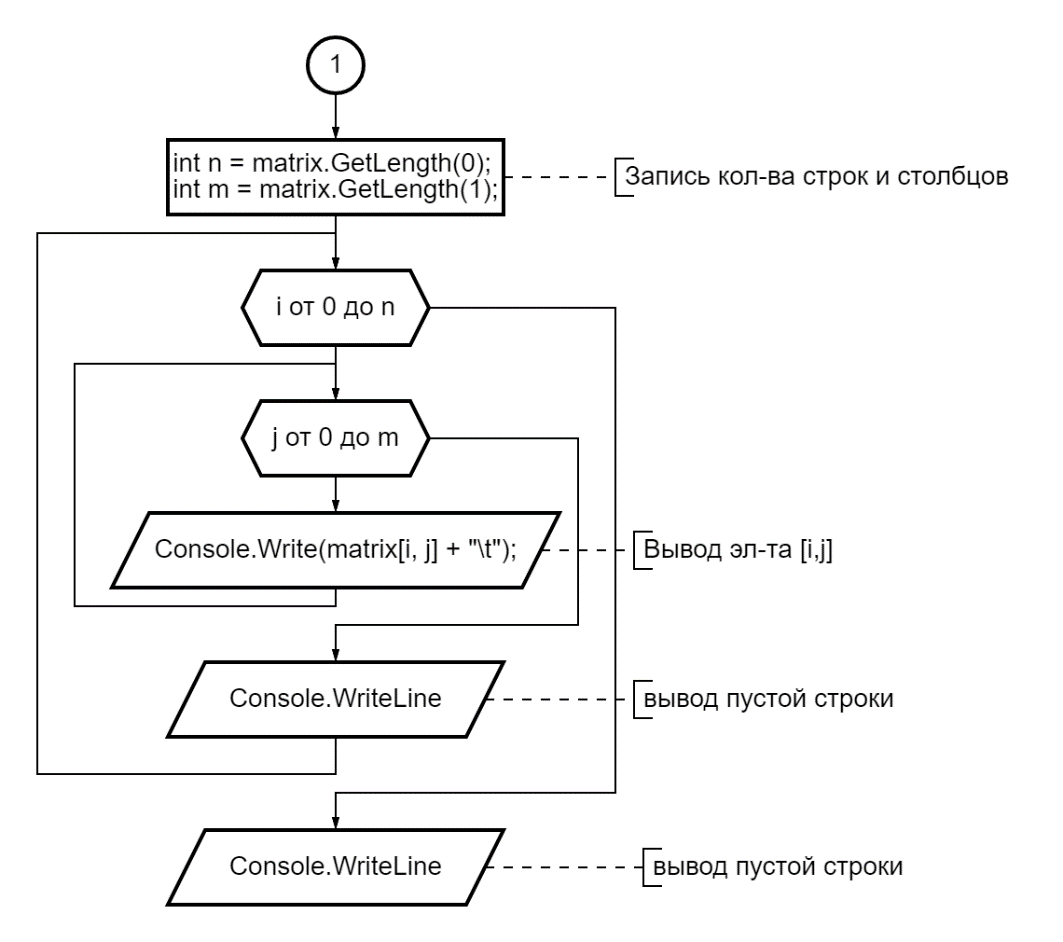
использованием многомерных массивов.

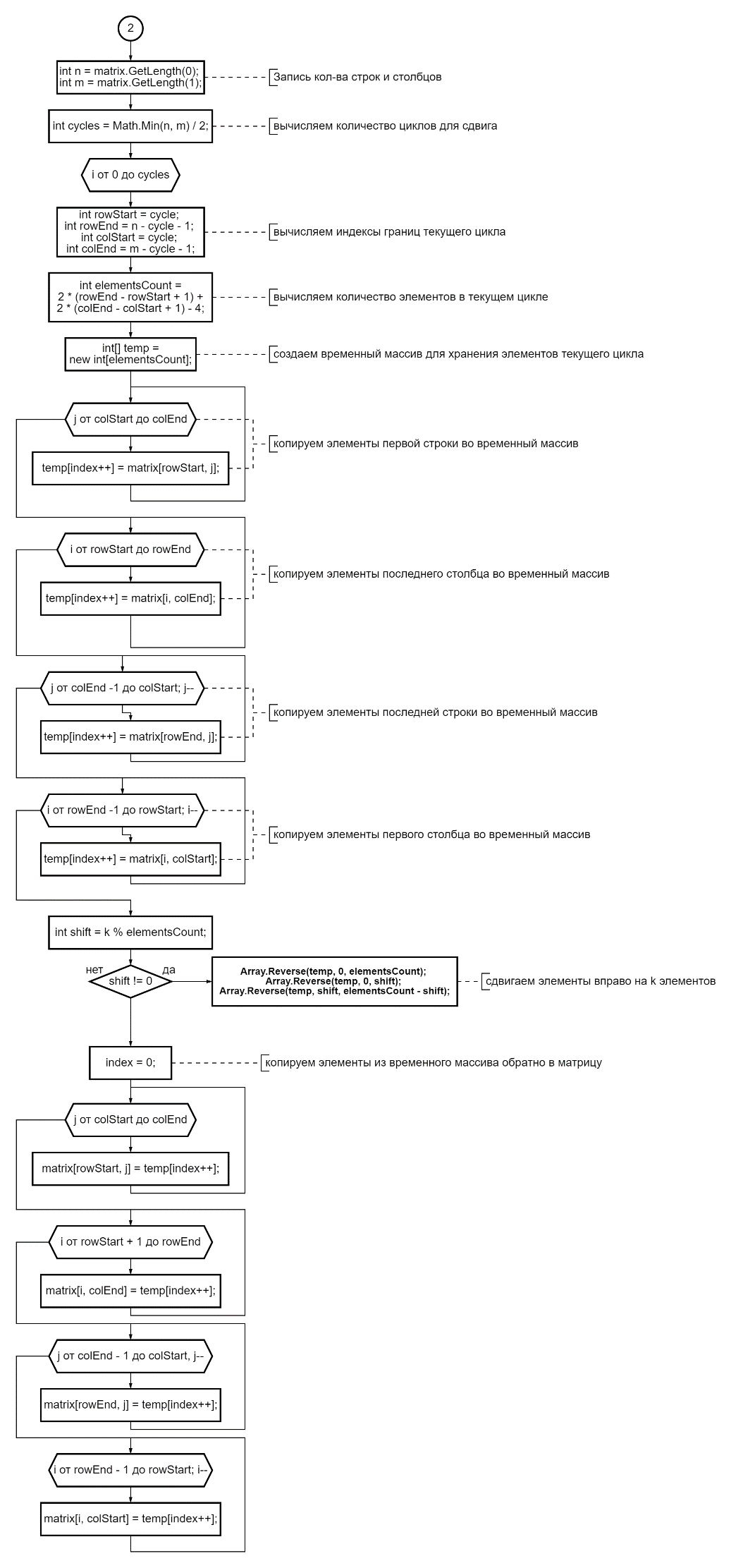
**Задание (вариант 14):**



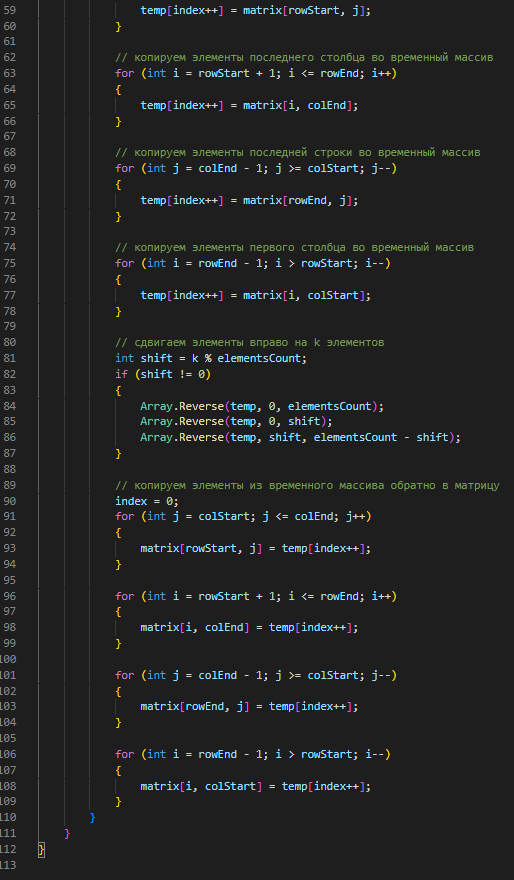
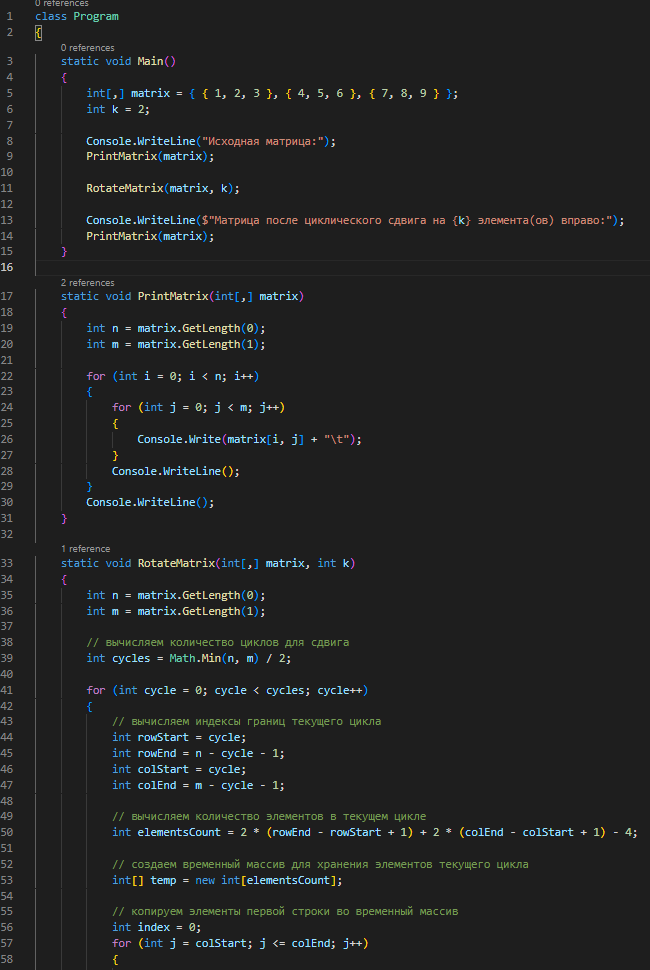
**Алгоритм схемы:**







**Код программы Скриншот**



**Исходный код программы**

class Program

{

static void Main()

{

int[,] matrix = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };

int k = 2;

Console.WriteLine("Исходная матрица:");

PrintMatrix(matrix);

RotateMatrix(matrix, k);

Console.WriteLine($"Матрица после циклического сдвига на {k} элемента(ов) вправо:");

PrintMatrix(matrix);

}

static void PrintMatrix(int[,] matrix)

{

int n = matrix.GetLength(0);

int m = matrix.GetLength(1);

for (int i = 0; i < n; i++)

{

for (int j = 0; j < m; j++)

{

Console.Write(matrix[i, j] + "\t");

}

Console.WriteLine();

}

Console.WriteLine();

}

static void RotateMatrix(int[,] matrix, int k)

{

int n = matrix.GetLength(0);

int m = matrix.GetLength(1);

// вычисляем количество циклов для сдвига

int cycles = Math.Min(n, m) / 2;

for (int cycle = 0; cycle < cycles; cycle++)

{

// вычисляем индексы границ текущего цикла

int rowStart = cycle;

int rowEnd = n - cycle - 1;

int colStart = cycle;

int colEnd = m - cycle - 1;

// вычисляем количество элементов в текущем цикле

int elementsCount = 2 \* (rowEnd - rowStart + 1) + 2 \* (colEnd - colStart + 1) - 4;

// создаем временный массив для хранения элементов текущего цикла

int[] temp = new int[elementsCount];

// копируем элементы первой строки во временный массив

int index = 0;

for (int j = colStart; j <= colEnd; j++)

{

temp[index++] = matrix[rowStart, j];

}

// копируем элементы последнего столбца во временный массив

for (int i = rowStart + 1; i <= rowEnd; i++)

{

temp[index++] = matrix[i, colEnd];

}

// копируем элементы последней строки во временный массив

for (int j = colEnd - 1; j >= colStart; j--)

{

temp[index++] = matrix[rowEnd, j];

}

// копируем элементы первого столбца во временный массив

for (int i = rowEnd - 1; i > rowStart; i--)

{

temp[index++] = matrix[i, colStart];

}

// сдвигаем элементы вправо на k элементов

int shift = k % elementsCount;

if (shift != 0)

{

Array.Reverse(temp, 0, elementsCount);

Array.Reverse(temp, 0, shift);

Array.Reverse(temp, shift, elementsCount - shift);

}

// копируем элементы из временного массива обратно в матрицу

index = 0;

for (int j = colStart; j <= colEnd; j++)

{

matrix[rowStart, j] = temp[index++];

}

for (int i = rowStart + 1; i <= rowEnd; i++)

{

matrix[i, colEnd] = temp[index++];

}

for (int j = colEnd - 1; j >= colStart; j--)

{

matrix[rowEnd, j] = temp[index++];

}

for (int i = rowEnd - 1; i > rowStart; i--)

{

matrix[i, colStart] = temp[index++];

}

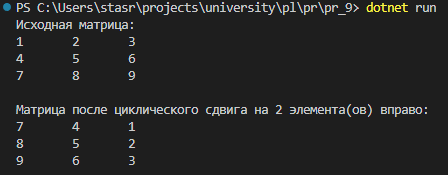
}

}

}

**Вывод результатов работы программы**

**Скриншот**



**Пояснение работы программы**

Данная программа реализует функцию для циклического сдвига элементов квадратной матрицы на заданное количество элементов вправо.

Сначала мы создаем двумерный массив matrix и заполняем его значениями. Затем вызываем функцию RotateMatrix с переданным параметром k=2, чтобы сдвинуть элементы на два элемента вправо.

Функция RotateMatrix сначала определяет размеры матрицы n и m, затем вычисляет количество элементов в матрице total и количество элементов, которые нужно циклически сдвинуть shiftCount. Далее происходит циклический сдвиг элементов матрицы с помощью временной переменной temp.

В конце мы выводим на экран исходную матрицу и матрицу после сдвига.