Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №7

# по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Обход графа в глубину»

Выполнили:

студенты группы 24ВВВ3

Агапов И.А.

Любченко В.К.

Приняли:

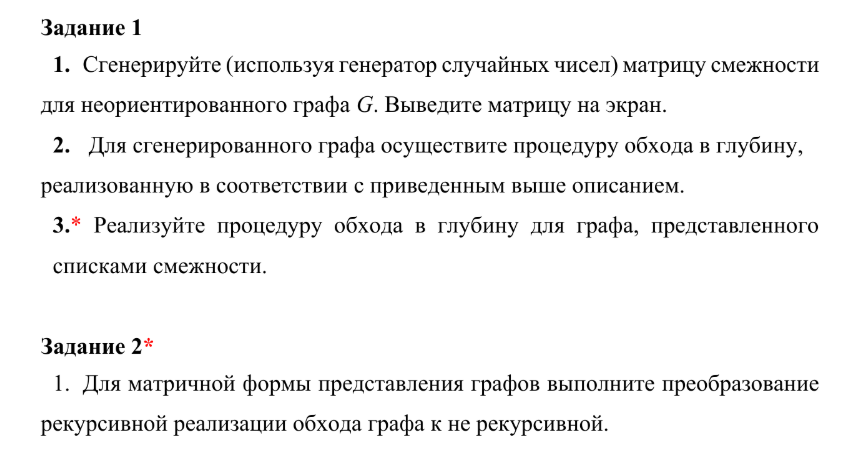
к.т.н., доцент Юрова О.В.

к.т.н., Деев М.В.

Пенза 2025

**Цель работы** –  Освоить алгоритм обхода графа в глубину (DFS) на практике, научившись генерировать неориентированный граф, представлять его в виде матрицы смежности и списков смежности, реализовывать рекурсивный и нерекурсивный обход в глубину для разных форм представления графа.

**Лабораторное задание:**



**Код программы**

**C#**

class proekt

{

static bool[] visitors;

public static int[,] CreateMatr()

{

Console.WriteLine("Введите размер матрицы: ");

int size = Convert.ToInt32(Console.ReadLine());

int[,] G = new int[size, size];

Random rnd = new Random();

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

{

for (int j = i; j < size; j++)

{

if (i == j)

{

G[i, j] = 0;

}

else

{

int value = rnd.Next(0, 2);

G[i, j] = value;

G[j, i] = value;

}

}

}

PrintMatr(G);

return G;

}

public static void PrintMatr(int[,] matr)

{

for (int i = 0; i < matr.GetLength(0); i++)

{

for (int j = 0; j < matr.GetLength(1); j++)

{

Console.Write(matr[i, j] + " ");

}

Console.WriteLine();

}

}

public static void Main()

{

//int[,] graph = {

// {0, 1, 1, 0, 0}, // 0 соединен с 1 и 2

// {1, 0, 0, 1, 0}, // 1 соединен с 0 и 3

// {1, 0, 0, 0, 1}, // 2 соединен с 0 и 4

// {0, 1, 0, 0, 0}, // 3 соединен с 1

// {0, 0, 1, 0, 0} // 4 соединен с 2

//};

int[,] graph = CreateMatr();

visitors = new bool[graph.GetLength(0)];

DFS(graph, 0);

DFSWithStack(graph, 0);

}

static void DFS(int[,] G, int currentVisit)

{

visitors[currentVisit] = true;

Console.WriteLine(currentVisit + " ");

for (int i = 0; i < G.GetLength(0); i++)

{

if (G[currentVisit, i] == 1 && visitors[i] == false)

{

DFS(G, i);

}

}

Console.WriteLine("вернулись в " + currentVisit);

}

static void DFSWithStack(int[,] graph, int startVertex)

{

int verticesCount = graph.GetLength(0);

bool[] visited = new bool[verticesCount];

Stack<int> stack = new Stack<int>();

stack.Push(startVertex);

while (stack.Count > 0)

{

int currentVertex = stack.Pop();

if (!visited[currentVertex])

{

visited[currentVertex] = true;

Console.WriteLine($"Посетили вершину: {currentVertex}");

for (int neighbor = verticesCount - 1; neighbor >= 0; neighbor--)

{

if (graph[currentVertex, neighbor] == 1 && !visited[neighbor])

{

Console.WriteLine($" Добавили в стек: {neighbor}");

stack.Push(neighbor);

}

}

}

}

}

}

**Результаты работы программы**

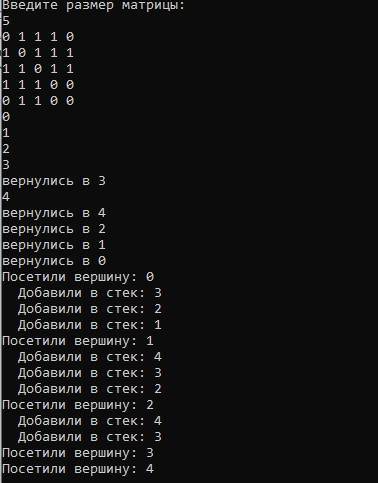


Рисунок 1 - Результат работы программы

**Вывод:** В ходе выполнения лабораторной работы был освоен алгоритм обхода графа в глубину (DFS) и приобретены практические навыки работы с различными представлениями графов.