

Министерство транспорта Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования «Российский университет транспорта» (ФГАОУ ВО РУТ(МИИТ), РУТ (МИИТ)

Институт транспортной техники и систем управления Кафедра «Управление и защита информации»

Лабораторная работа № 10

по дисциплине: «Программирование и основы алгоритмизации» на тему: «Методы»

Выполнил: ст. гр.

Денг Денг Гак Луал

Вариант №13

27.05.2025

Проверил: к.т.н., доц. Сафронов А.И

Москва – 2025 г.

1. Цель

Разработать консольное приложение для работы с трёхмерным массивом, обеспечивающее:

- Ввод размеров массива с контролем ошибок.
- Заполнение массива пользователем.
- Навигацию по слоям (Z -направление) с визуализацией данных.

2. Формулировка задачи

Создать программу, которая:

- Запрашивает размеры X (строки), Y (столбцы), Z (слои) с проверкой на положительные целые числа.

- Позволяет вручную заполнить массив.
- Реализует интерфейс для просмотра слоёв с использованием клавиш **L** (предыдущий слой), **R** (следующий слой), **Q** (выход).
- Форматирует вывод слоя в виде таблицы.

3.Блок схема алгоритма

Рисунок 3.1 блок схема алгоритма

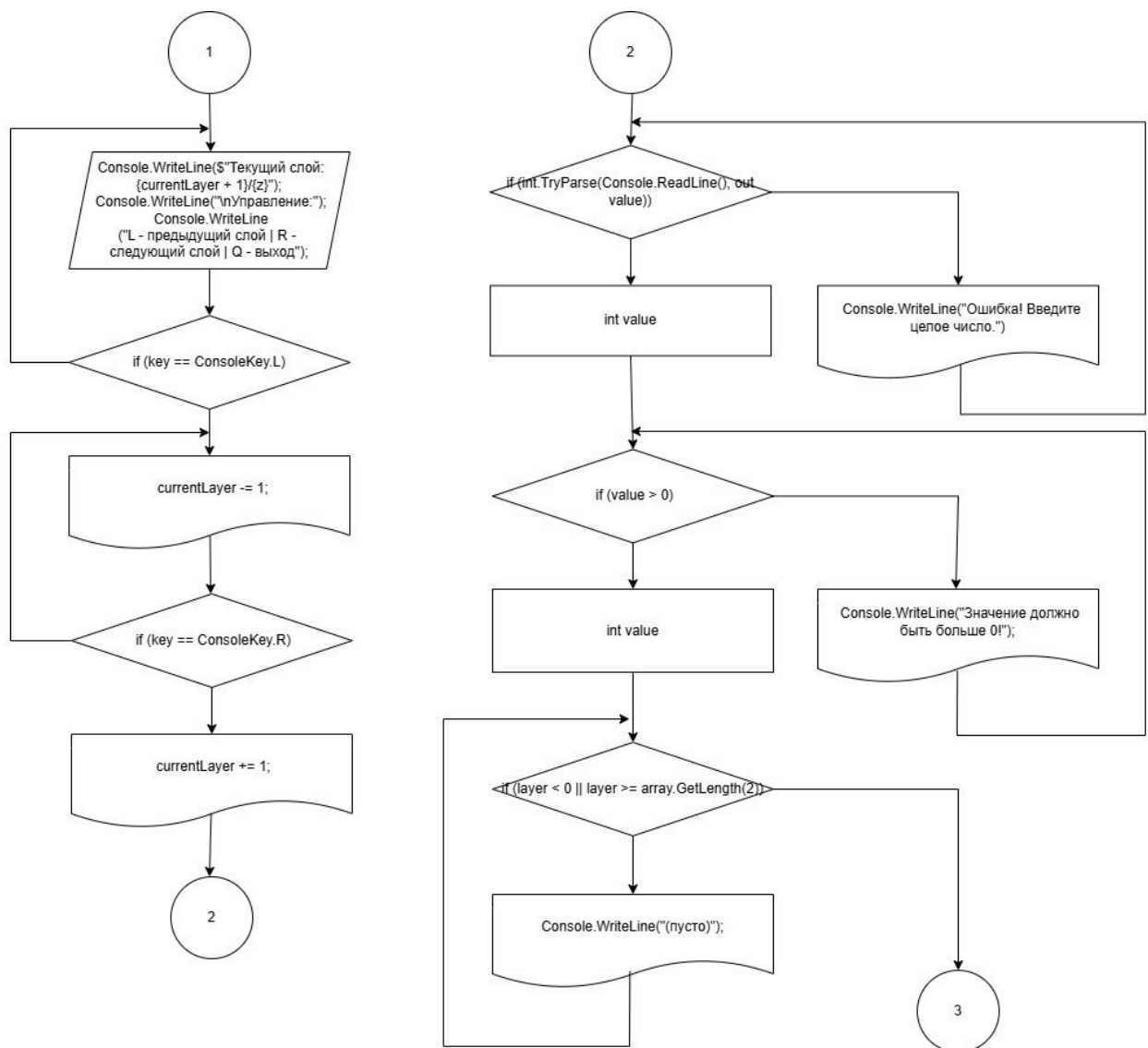


Рисунок 3.2 блок схема алгоритма

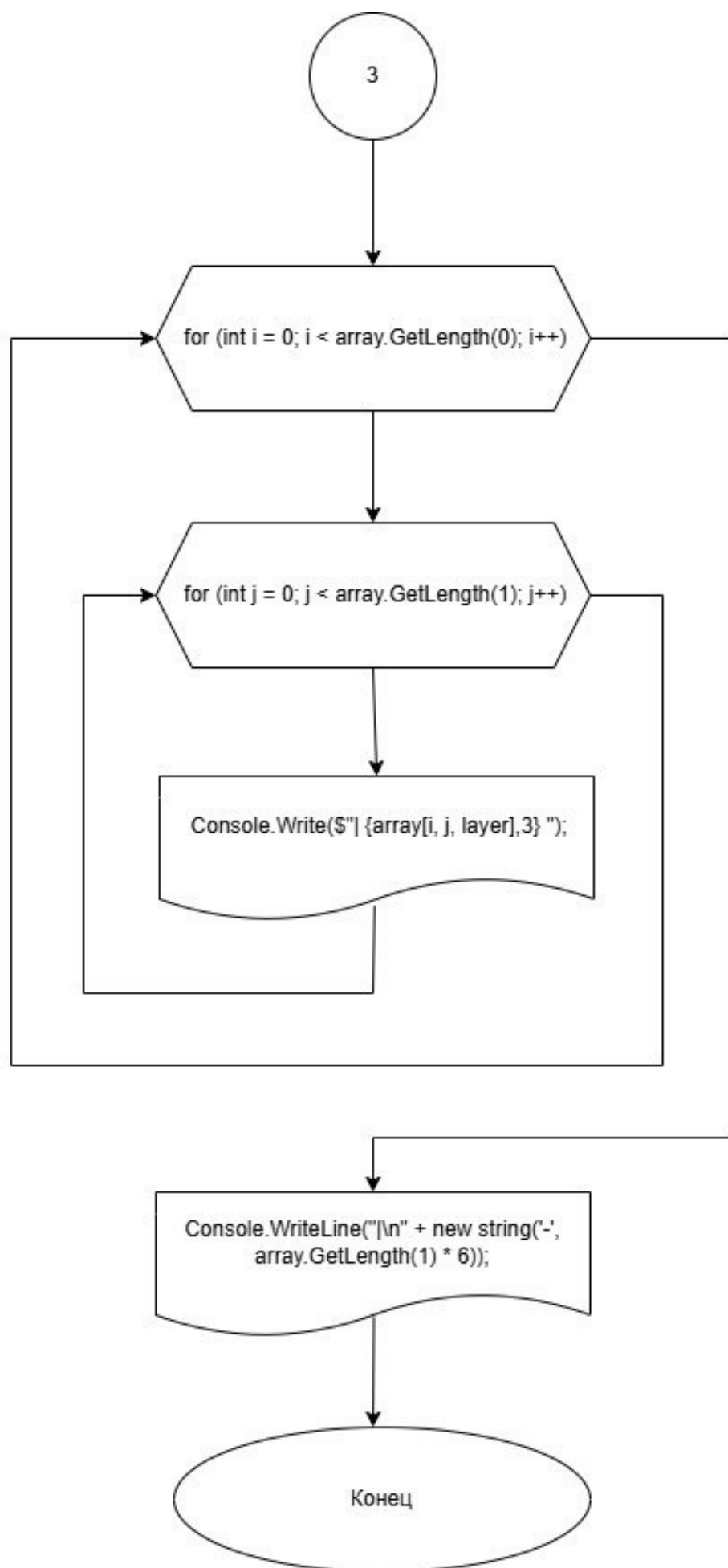


Рисунок 3.3 блок

схема алгоритма

4. Подбор тестовых примеров

- **Тестовый пример №1:**
 - Ввод: $X=2, Y=2, Z=2$.
 - Заполнение: 1, 2, 3, 4, 5, 6, 7, 8.
 - Проверка: Переход между слоями и корректность отображения.
- **Тестовый пример №2:**
 - Ввод: $X=3, Y=3, Z=1$.
 - Заполнение: Все элементы = 0.
 - Проверка: Обработка единственного слоя.
- **Тестовый пример №3:**
 - Ввод: $X=1, Y=4, Z=3$.
 - Заполнение: Случайные числа.
 - Проверка: Отображение длинных строк и переходы между слоями.

5. Листинг программы

```
using System;

namespace ThreeDArrayNavigator
{
    class Program
    {
        static void Main(string[] args)
        {
            // Ввод размеров массива с контролем ошибок
            int x = GetPositiveInteger("Введите размер X (строки): ");
            int y = GetPositiveInteger("Введите размер Y (столбцы): ");
            int z = GetPositiveInteger("Введите количество слоёв Z: ");

            // Инициализация и заполнение 3D-массива
            int[, ,] array = new int[x, y, z];
```

```

Console.WriteLine("\nЗаполнение массива:");

for (int i = 0; i < x; i++)
{
    for (int j = 0; j < y; j++)
    {
        for (int k = 0; k < z; k++)
        {
            array[i, j, k] = GetInteger($"Элемент [{i},{j},{k}]: ");
        }
    }
}

// Навигация по слоям
int currentLayer = 0;
ConsoleKey key;
do
{
    Console.Clear();

    Console.WriteLine($"Текущий слой: {currentLayer + 1}/{z}");
    PrintLayer(array, currentLayer);

    Console.WriteLine("\nУправление:");
    Console.WriteLine("L - предыдущий слой | R - следующий слой | Q - выход");
    key = Console.ReadKey().Key;

    // Обработка навигации только с помощью if-else
    if (key == ConsoleKey.L)
    {
        currentLayer -= 1;
    }
    else
    {

```

```

        if (key == ConsoleKey.R)
        {
            currentLayer += 1;
        }

        // Q - выход, другие клавиши не обрабатываются
    }

} while (key != ConsoleKey.Q);
}

// Метод для безопасного ввода целых чисел
static int GetInteger(string prompt)
{
    int value;
    while (true)
    {
        Console.Write(prompt);
        if (int.TryParse(Console.ReadLine(), out value))
            return value;
        else
            Console.WriteLine("Ошибка! Введите целое число.");
    }
}

// Метод для ввода положительных целых чисел
static int GetPositiveInteger(string prompt)
{
    int value;
    while (true)
    {
        value = GetInteger(prompt);
        if (value > 0)

```

```

        return value;
    else
        Console.WriteLine("Значение должно быть больше 0!");
    }
}

// Вывод слоя массива с форматированием
static void PrintLayer(int[,] array, int layer)
{
    if (layer < 0 || layer >= array.GetLength(2))
    {
        Console.WriteLine("(пусто)");
        return;
    }

    // Заголовок таблицы
    Console.WriteLine(new string('-', array.GetLength(1) * 6));
    for (int i = 0; i < array.GetLength(0); i++)
    {
        for (int j = 0; j < array.GetLength(1); j++)
        {
            Console.Write($"{array[i, j, layer],3} ");
        }
        Console.WriteLine($"|\n" + new string('-', array.GetLength(1) * 6));
    }
}
}

```

6. Расчет тестовых примеров на ПК

- Рисунок 6.1 (Тестовый пример №1):

- Слой 1:

1	2
---	---

	3		4	
--	---	--	---	--

Слой 2:

	5		6	
	7		8	

- **Рисунок 6.2 (Тестовый пример №2):**

- Слой 1:

	0		0		0	
--	---	--	---	--	---	--

	0		0		0	
--	---	--	---	--	---	--

	0		0		0	
--	---	--	---	--	---	--

- **Рисунок 6.3 (Тестовый пример №3):**

- Слой 1: | 10 | 20 | 30 | 40 |
- Слой 2: | 50 | 60 | 70 | 80 |
- Слой 3: | 90 | 100 | 110 | 120 |

7. Вывод

Программа успешно решает поставленные задачи:

- Корректный ввод размеров и данных.
- Удобная навигация по слоям.
- Форматированный вывод.

Рекомендации:

- Добавить проверку границ слоёв (например, запрет перехода за $Z=0$ или $Z=Max$).
- Улучшить интерфейс (подсветка активного слоя, отображение подсказок).