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 № 9

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Одномерные массивы»

Выполнил: ст. гр. ТЭИ-111

Попов В.А.

Вариант №3

02.10.2025

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

15.10.2025

(дата приёмки)

Зачтено

Москва – 2025 г.

1. **Цель**

Изучить обработку одномерных массивов в Visual C# и отработать навыки корректного ввода/вывода данных с применением TryParse, следуя рекомендуемой схеме оформления отчёта и опорной схеме ввода-вывода.

1. **Формулировка задачи**

Разработать программу на языке Visual C#, которая для одномерного массива целых чисел длины n (индексация с нуля) находит разность индексов наибольшего и наименьшего элементов массива. Индексация — нулевая. Ввод элементов и их корректность контролируются «до победного» с применением состояний метода TryParse.

1. **Блок-схема алгоритма**

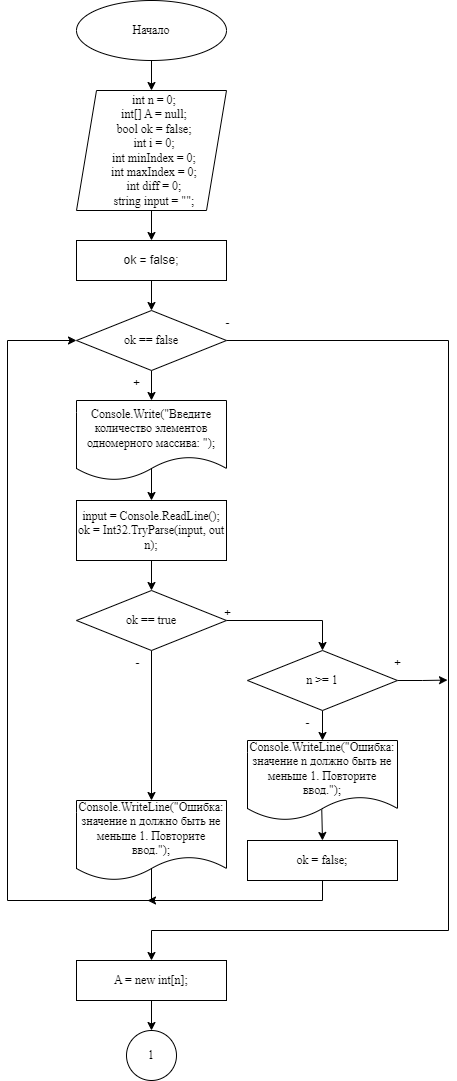
****

Рисунок 3.1.1 – Блок-схема алгоритма

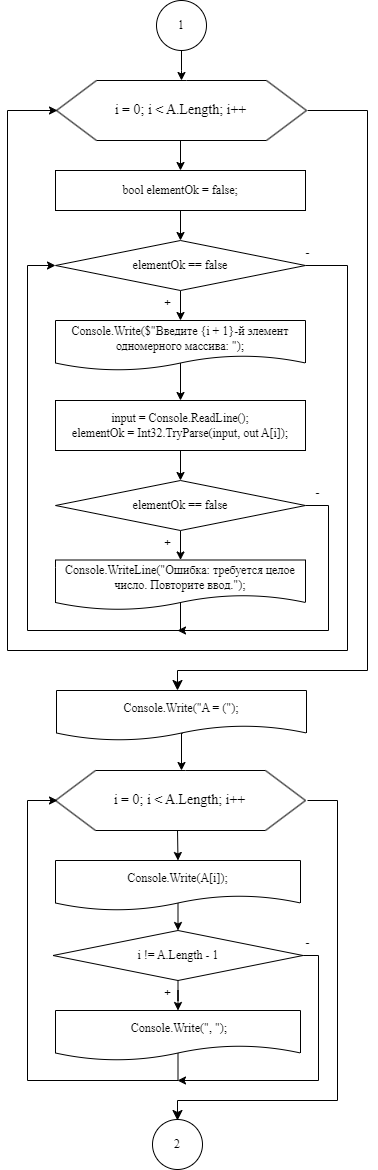
****

Рисунок 3.1.2 – Блок-схема алгоритма

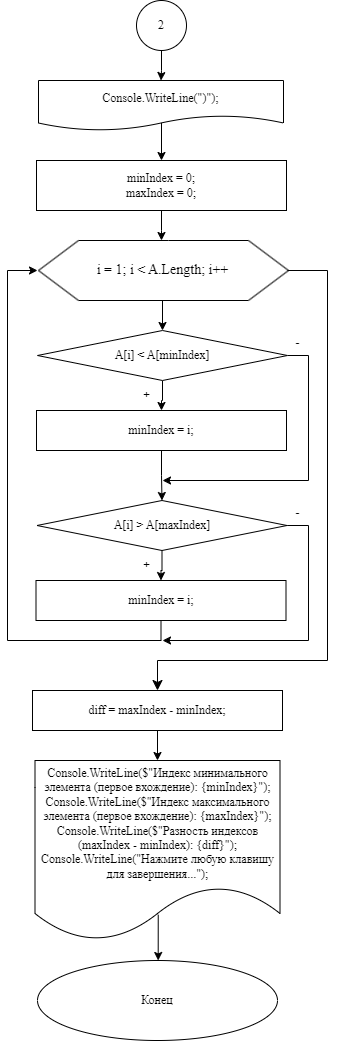


Рисунок 3.1.2 – Блок-схема алгоритма

1. **Подбор тестовых примеров**
   1. Тестовый пример №1

n=7

A=(1, -7, 3, 0, -9, -1, 5)

Индекс минимального элемента: 4

Индекс максимального элемента: 6

Разность индексов (6 - 4): 2

* 1. Тестовый пример №2

n=a

Ошибка: требуется целое число.

* 1. Тестовый пример №3

n=3

A=(1, 2, a)

Ошибка: требуется целое число.

1. **Листинг (код) программы**

using System;

class Program

{

static void Main(string[] args)

{

// === Блок объявления всех переменных ===

int n = 0; // Длина массива (целое n >= 1)

int[] A = null; // Одномерный массив целых чисел

bool ok = false; // Флаг корректности последнего ввода (состояние TryParse)

int i = 0; // Счётчик циклов

int minIndex = 0; // Индекс первого вхождения минимального элемента

int maxIndex = 0; // Индекс первого вхождения максимального элемента

int diff = 0; // Разность индексов: maxIndex - minIndex

string input = ""; // Буфер строки для валидации ввода через TryParse

// === Ввод n по схеме "ввод до победного" (без возможности прерывания) ===

// Требование: пользователю предлагается вводить до тех пор, пока значение не станет корректным.

// Контроль: целое число и ограничение n >= 1.

ok = false;

while (ok == false)

{

Console.Write("Введите количество элементов одномерного массива: ");

input = Console.ReadLine();

ok = Int32.TryParse(input, out n);

if (ok == true)

{

if (n >= 1)

{

// Корректное n — выходим из цикла, ok уже true

}

else

{

Console.WriteLine("Ошибка: значение n должно быть не меньше 1. Повторите ввод.");

ok = false; // Принудительно продолжаем запрос, прерывания нет

}

}

else

{

Console.WriteLine("Ошибка: требуется целое число. Повторите ввод.");

// ok остаётся false — цикл повторит ввод

}

}

// === Создание массива требуемой длины ===

A = new int[n];

// === Ввод элементов массива по схеме "ввод до победного" для каждого A[i] ===

// Для каждого индекса i запрашиваем значение, пока TryParse не подтвердит корректность.

for (i = 0; i < A.Length; i++)

{

bool elementOk = false; // отдельный флаг для текущего элемента

while (elementOk == false)

{

Console.Write($"Введите {i + 1}-й элемент одномерного массива: ");

input = Console.ReadLine();

elementOk = Int32.TryParse(input, out A[i]);

if (elementOk == false)

{

Console.WriteLine("Ошибка: требуется целое число. Повторите ввод.");

}

else

{

// Значение корректно — переходим к следующему элементу

}

}

}

// === Корректный вывод исходного массива в формате опорной схемы ===

Console.Write("A = (");

for (i = 0; i < A.Length; i++)

{

Console.Write(A[i]);

if (i != A.Length - 1)

{

Console.Write(", ");

}

}

Console.WriteLine(")");

// === Поиск индексов первого минимума и первого максимума (нулевая индексация) ===

minIndex = 0;

maxIndex = 0;

// Если длина > 1, сравниваем последовательно элементы с текущими экстремумами

for (i = 1; i < A.Length; i++)

{

if (A[i] < A[minIndex])

{

minIndex = i;

}

if (A[i] > A[maxIndex])

{

maxIndex = i;

}

}

// === Разность индексов ===

diff = maxIndex - minIndex;

// === Вывод результатов с пояснениями ===

Console.WriteLine($"Индекс минимального элемента (первое вхождение): {minIndex}");

Console.WriteLine($"Индекс максимального элемента (первое вхождение): {maxIndex}");

Console.WriteLine($"Разность индексов (maxIndex - minIndex): {diff}");

// Пауза для удобства проверки

Console.WriteLine("Нажмите любую клавишу для завершения...");

Console.ReadKey(true);

}

}

1. **Расчет тестовых примеров на ПК**

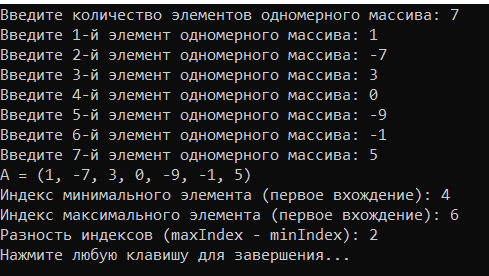
****

Рисунок 6.1.1 – Тестовый пример №1

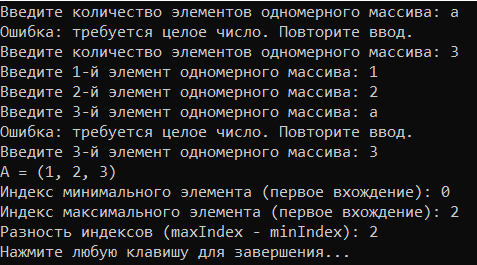
****

Рисунок 6.1.2 – Тестовый пример №2

1. **Вывод**

В ходе выполнения лабораторной работы было изучено использование одномерных массивов в языке программирования *Visual C#,* а также реализована программа для обработки одномерного массива: организован «ввод до победного» с использованием TryParse, выполнен корректный форматированный вывод массива, найдены индексы минимального и максимального элементов (с нулевой индексацией) и вычислена их разность.